

Министерство образования и науки Российской Федерации
САНКТ-ПЕТЕРБУРГСКИЙ
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО

О.Р. Рыкин

**ЛИНЕЙНОЕ
ПРОГРАММИРОВАНИЕ
В МАТЛАБЕ**
Универсальные линейные оптимизаторы
Задания и Задачи



Издательство Политехнического Университета
Санкт-Петербург
2016

Министерство образования и науки Российской Федерации

САНКТ-ПЕТЕРБУРГСКИЙ
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО

О. Р. Рыкин

ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ
В МАТЛАБЕ
УНИВЕРСАЛЬНЫЕ ЛИНПРОГОПТИМИЗАТОРЫ:
ПРОИЗВОДИТЕЛЬНОСТЬ
И ТАБЛИЧНЫЙ ФОРМАТ РЕЗУЛЬТАТА
ЗАДАНИЯ И ЗАДАЧИ



Санкт-Петербург
2016

УДК 004.438

ББК 32.81

Р94

Рецензенты:

Доктор технических наук, профессор СПбГУТ *М. В. Савищенко*

Кандидат технических наук, профессор СПбПУ *Л. В. Бабко*

**Рыкин О. Р. Линейное программирование в Матлабе. Универсальные линпро-
гоптимизаторы: производительность и табличный формат результата. Задания и
задачи / О. Р. Рыкин. — СПб. : Изд-во Политехн. ун-та, 2016. — 208 с.**

В монографии предложены усовершенствованные и развитые программные инструменты (13 объектов) решения непараметрических и однопараметрических задач линпрогоптимизации. Они обеспечивают многократное повышение операторской производительности, создание времянезависимой связи результата и условий оптимизации при решении множества задач, выдачу результата в таблице офисного формата, автоматическое создание базы результатов линпрогоптимизации, возможность простой коррекции условий оптимизации в случае начального неуспеха. Показаны примеры применения каждого инструмента, в том числе для некоторых — решение задач с размерностью оптимизационного вектора в 5000 и 7000 элементов. Приведены правила настройки и указания по применению. Инструменты созданы на основе матлабовской функции `linprog(...)` с использованием команд Матлаба, описание которых приведено в приложении.

В целях быстрого и надёжного усвоения материала даны детальные методики решения примеров и тематических задач по линпрогоптимизации. Часть примеров непараметрической и однопараметрической оптимизации имеет размерность 5...10, а остальная часть — непараметрическая оптимизация с размерностью 1000...8000. Число вариантов примеров — 90. Число тематических задач — 80.

Монография предназначена для использования при подготовке бакалавров по направлению 27.03.05 «Инноватика» по дисциплинам «Системный анализ и принятие решений» и «Теория и системы управления» для углублённого освоения материала дисциплин и при проведении курсовых и лабораторных работ.

Табл. 95. Ил. 52. Библиогр.: 18 назв.

Печатается по решению

Совета по издательской деятельности Ученого совета

Санкт-Петербургского политехнического университета Петра Великого.

© Рыкин О. Р., 2016

© Санкт-Петербургский политехнический
университет Петра Великого, 2016

ISBN 978-5-7422-5472-0

ОГЛАВЛЕНИЕ

Введение	8
1. Инструменты линейного программирования Матлаба linprog и bintprog.....	11
1.1. Рабочее пространство Матлаба.....	11
Ключевые термины раздела 1.1.....	11
Описание пространства.....	13
1.2. Схема простого вычислительного процесса в Матлабе.....	15
1.2.1. Подготовка среды.....	15
1.2.2. Ввод исходных данных и команд.....	16
1.2.3. Вычисления и сохранение результатов.....	18
1.3. Функции линейного программирования.....	19
2. Функция линейного программирования linprog(...) и линпрогоптимизаторы на её основе.....	20
2.1. Ключевые термины раздела 2.....	20
2.2. Описание функции линейного программирования linprog(...).....	22
2.2.1. Алгоритмы функции linprog.....	23
2.2.2. Справочные команды и программа функции.....	23
2.2.3. Непараметрические и параметрические линпрогоптимизаторы на основе функции linprog.....	23
2.3. Вычислительные возможности linprog. Поиск 5000-компонентного минимизационного вектора.....	26
2.4. Линпрогуказания и линпрогоптимизаторы с использованием функции linprog.....	27
2.4.1. Шаблон линпрогуказания с ручным вводом номеров объектов и выходных данных.....	28
2.4.2. Универсальный линпрогоптимизатор с переключательным вводом типа оптимизации, номеров объектов и выходных данных.....	29
2.4.3. Применение универсального линпрогоптимизатора для решения задач с размерностью 7000.....	32
2.4.3.1. Линпрогминимум для Ao 7000x7000 с числом ненулевых значений 100x50.....	32
2.4.3.2. Линпрогмаксимум, Ao из 1-й задачи, Aeqo с ненулевыми значениями 300x500.....	34
2.4.4. Универсальный линпрогоптимизатор с выдачей единой итог-таблицы запуска оптимизации.....	35
2.4.4.1. Описание универсальных итогтабличных линпрогуказания LinProgUnivUkz_Tab и линпрогоптимизатора LinProgUniversAvtomatMU_Tab.....	35
2.4.4.2. Формирование шаблонов итог-таблиц ItogTabLPrmi.rpt/ ItogTabLPrma.rpt/.....	42
2.4.4.3. Настройки при применении универсального итогтабличного линпрогоптимизатора LinProgUniversAvtomatMU_Tab.....	44
2.5. Коррекция задач линпрогоптимизации с использованием выходных объектов pr и gv.....	44
2.6. Решение серии линпрогзадач в одной сессии Матлаба.....	46
2.7. Правила и этапы решения непараметрических задач линейного программирования посредством линпрогоптимизаторов с функцией linprog.....	49
2.7.1. Правила и этапы.....	49
2.7.2. Уставки оптимизатора linprog и значения переменной exitflag (ef).....	51
3. Зависимость линпрогоптимума от параметра p1. Однопараметрические линпрогоптимизаторы: с ручной настройкой и универсальный.	53
3.1. Ручной однопараметрический оптимизатор pLinProgOptRuch.	53
Содержание и смысл команд ряда строк линпрогоптимизатора pLinProgMinRuch.....	55
3.2. Необходимость вставки строки «Очистка предыдущего результата» в однопараметрические линпрогоптимизаторы	56
3.3. Построение зависимости Fcmin(P1)	58
3.4. Коррекция задачи однопараметрической линейного программирования с использованием выходных объектов prgm/ prgma в отладочной таблице. Программа pTabLinProgRuchOtladHtm построения таблицы.....	59
3.4.1. Пример коррекции условий задачи однопараметрической линпрогмаксимизации.....	59
3.4.1.1. Правила коррекции условий однопараметрической линпрогзадачи.....	59
3.4.1.2. Процедура коррекции условий задачи линпрогмаксимизации оптимизатора рис. 3.2.	61

3.4.1.3. Программа pTabLinProgRuchOtladHtm построения отладтаблицы.	62
3.5. Построение итогтаблиц однопараметрической линпрогоптимизации для оптимизаторов с ручной настройкой pLinProgMinRuch.m и pLinProgMaxRuch.m	66
3.5.1. Описание итогтаблиц	66
3.5.2. Формирователь итогтаблиц p-линпрогоптимизации - pLinProgRuchP1OptTabFormer	67
3.5.3. Настройка html-шаблона итогтаблиц pLinProgItogRuchTab.rpt для формирователя pLinProgRuchP1OptTabFormer	68
3.6. Универсальный p-линпрогоптимизатор pLinProgTabGenOptimizer	70
3.6.1. Описание	70
3.6.2. Построитель итогтаблиц pLinProgUnverP1TabStroyker	72
3.6.3. Формирователь отладтаблиц pLinProgUnverOtladTab	73
3.6.4. Пример использования формирователя pLinProgUnverOtladTab для коррекции условий p-линпрогоптимизации	76
3.6.5. Процедура применения универсального p-линпрогоптимизатора	78
4. Задание 4. Решение примеров и задачи линейного программирования посредством универсальных линпрогоптимизаторов.....	80
4.1. Применение универсальных линпрогоуказания LinProgUnivUkz_Tab и линпрогоптимизатора LinProgUnivesAvtomatMU_Tab для решения примеров линпрогоптимизации с размерностью 5...10	80
4.2. Применение универсальных линпрогоуказания LinProgUnivUkz и линпрогоптимизатора LinProgUnivesAvtomatMU для решения примеров линпрогоптимизации с размерностью 1000...8000	82
4.3. Вычисление зависимости линпрогоптимумов от одного параметра p с использованием универсального p-линпрогоптимизатора pLinProgTabGenOptimizer и построителя итогтаблиц pLinProgUnverP1TabStroyker.....	83
5. Варианты данных к Заданию 4	85
5.1. Структура линпрогпримеров (ЛПП) для данных табл. дз5.1	85
Таблица дз5.1	85
Коэффициенты линпрогпримеров	85
Таблица дз5.2	100
Данные для линпрогоптимизации с размерностью 1000...8000	100
Таблица дз5.3	103
Данные для ЦФ $y_{42}(x, p_1)$ и $y_{43}(x, p_1, p_2)$	103
6. Самостоятельное решение тематической линпрогозадачи из разд.7	107
7 . З А Д А Ч И	109
1. Машиностроительное предприятие.	109
2. Предприятие выпускает радиоприемники трех различных моделей.	109
3. Автозавод выпускает две модели автомобилей.	109
4. Цех выпускает 5 деталей.	110
5. Для производства комплексной продукции требуется изготовить 4 вида изделий.....	110
6. На мебельной фабрике требуется раскроить.	110
7. Фирма – булочно-кондитерский комбинат (БКК).....	110
8. Для строительства домов на 100 строительных площадках.	111
9. Участник экспедиции укладывает рюкзак.....	111
10. Пряжильная фабрика для производства двух видов пряжи.	111
11. Нефтеперерабатывающий завод может использовать.	112
12. Рацион кормления коров на молочной ферме.	113
13. В области имеются 3 цементных завода.....	113
14. Колхоз «Куликово поле» может засеять свои поля пшеницей пяти сортов.	113
15. Имеются 2 склада и 6 цехов обработки деталей.	114
16. Фирма производит комплект деталей.	114
17. Фирма продаёт консервы.	114
18. Определение плана компании по закупкам и продажам бананов.	115
19. В механическом цехе имеется три металлорежущих станка.....	115
20. В амбаре 4 мышиных норы.	116
21. На трёх морских линиях используются суда пяти типов.....	116

22. Порт обеспечивается вагонами 4-х типов для погрузки 3-х видов различных грузов.....	116
23. Хозяйство имеет 850 гектаров земли.	117
24. На морском бассейне имеются 3 судоремонтных завода.	117
25. Хозяйство «Курская дуга» специализируется в полеводстве.....	117
26. Для перегрузки 5 типов грузов выделено 4 козловых крана.	118
27. На фабрике эксплуатируются 4 типа ткацких станков С1... С3.....	118
28. Для перевозок пассажиров на 3 линиях используются суда 3 типов.....	118
29. Три грузопотока обрабатываются 5 типами перегрузочного оборудования.	119
30. Семейная фирма производит три широко популярных напитка.....	119
31. Завод-производитель высокоточных элементов для автомобилей.....	119
32. Завод по производству компьютеров и системы подготовки текстов.....	120
33. Менеджер по ценным бумагам намерен разместить 100 000 ф. ст. капитала.	120
34. Оливер А. Петерс скоро выйдет на пенсию.....	120
35. Фабрика «GRM plc» выпускает три вида каш для завтрака.	121
36. Порт обеспечивается вагонами 4 типов для погрузки 3 видов грузов.	121
37. Для перевозок пассажиров на 3 линиях используются суда 4 типов.....	121
38. Нефтяная компания "РТ" улучшает выпускаемое топливо.	122
39. Завод производит порошок для изготовления напитков трех видов.....	122
40. "Princetown Paints Ltd" выпускает три основных типа румян.....	123
41. Администрация компании "Nemesis Company" приняла решение.	123
42. Менеджер международной банковской организации по инвестициям.	124
43. Компания "Bermuda Paint" – частная промышленная фирма.....	124
44. Компания имеет четыре сбытовые базы и четыре заказа.....	125
45. В распоряжении некоторой компании имеется 6 торговых точек и 6 продавцов.....	125
46. Три завода поставляют разновидность стали на пять торговых складов.....	125
47. Компании "Zeit pie" принадлежат три фермы, где выращиваются овощи.....	126
48. Дирекция деревообрабатывающего предприятия "Vibra" приняла на работу.	126
49. В Kingdom of the Republik of Jdion имеется пять угольных шахт.	127
50. Компания "Braintree Electronics Company" выпускает 100-метровые ленты.....	127
51. Компания "Royal Wedgetoun Pottery" получила заказы на 3 вида изделий.	128
52. В комбинате "Сосенка" приняты на работу 6 человек.....	128
53. Компания "Orange Computer" производит матричные печатающие устройства.	129
54. Членов Ассоциации ученых Мидленда недавно уведомили.....	129
55. Комбинат выпускает книжные полки А из натурального дерева со стеклом.....	130
56. На приобретение оборудования для нового участка выделено 108 тыс. руб.	130
57. Рацион кормления коров на молочной ферме может состоять из 6 продуктов.....	131
58. Отдел кадров предприятия устроил конкурсный набор.....	131
59. На складах хранится мука, которую необходимо завезти в хлебопекарни.	132
60. По заказу пяти потребителей А, Б, В, Г, Д.	132
61. По заказу пяти потребителей А, Б, В, Г, Д на 4 предприятиях-изготовителях.....	133
62. По заказу пяти потребителей А, Б, В, Г, Д на 4 предприятиях-изготовителях.....	134
63. На АО «Светлана» подготовлены к серийному производству 5 новых изделий.....	134
64. На АО «Позитрон» подготовлены к серийному производству 5 новых изделий.	135
65. Промышленное предприятие «Белподшип» может выпускать подшипники.	136
66. Продукция городского молочного завода.	136
67. Для поддержания нормальной жизнедеятельности среднему человеку... ..	137
68. Диспетчерская служба имеет следующие минимальные потребности.	137
69. Завод должен изготовить 3 типа стальных сплавов.	137
70. Компания по производству гусеничных механизмов.	138
71. В отделе технического контроля фирмы работают контролеры.....	138
72. Для обогрева помещений используют четыре агрегата.....	138
73. При производстве четырех типов кабеля выполняется пять групп операций.	139
74. Цех выпускает комплекты из 3 деталей.....	139
75. Ресторан работает 7 дней в неделю.	140
76. Авиакомпания «Альфа» составляет расписание вылетов.	140

77. В лесничестве.....	140
78. Импресарио готовит выставку старинных автомобилей.	141
79. Электроэнергетическая сеть.	141
80. Винодел смешивает четыре сорта винограда.....	141
Приложение 1	143
Пример оформления отчета по решению линпрогзадачи из разд. 7.....	143
Приложение 2	145
Программа linprog	145
Приложение 3. Команды Матлаба R2009b, используемые в программах Монографии.....	153
1. inputdlg	153
2. keyboard.....	155
3. menu	155
4. assignin	156
Syntax	156
Description	156
5. num2str	156
6. now.....	157
7. disp.....	157
8. num2str	158
9. str2num	159
10. clock	160
11. datenum.....	161
12. datestr	163
13. Operators	166
On this page.....	166
Arithmetic Operators	166
Relational Operators	166
Logical Operators	166
Operator Precedence	166
Arithmetic Operators	166
Arithmetic Operators and Arrays.....	167
Relational Operators	167
Relational Operators and Arrays	167
Relational Operators and Empty Arrays	168
Logical Operators	168
Element-Wise Operators and Functions.....	168
Other Array Functions.....	169
Short-Circuiting in Elementwise Operators.....	169
Logical Expressions Using the find Function.	169
Bit-Wise Functions	170
Short-Circuit Operators.....	170
Приоритет операторов.....	171
Precedence of AND and OR Operators	171
Overriding Default Precedence	172
14. isempty	172
15. char.....	172
Syntax	172
16. Other Data Structures	172
Multidimensional Arrays	173
Cell Arrays	174
17. cell.....	177
18. cat	178
19. if.....	178
20. report.....	180
21. blanks	180

22. cellstr	181
23. fprintf	181
24. num2cell	185
25. Cell Arrays	186
Operators That Concatenate Cells and Cell Content	196
Operators Used for Cell Array Indexing.....	196
Syntax	196
Description	196
Functions Related to Constructing the Array	197
Functions Related to the Type of the Array	197
Function.....	197
Description	197
cell2struct	197
Convert cell array to structure array.	197
class	197
Create object or return class of object.....	197
iscell	197
Determine whether input is cell array.	197
struct2cell	197
Convert structure to cell array.....	197
whos	197
List variables in workspace.	197
Functions Related to Obtaining Cell Array Contents	197
Function.....	197
Description	197
celldisp	197
Display cell array contents.....	197
cellplot	197
Display a graphical depiction of a cell array.....	197
deal.....	197
Copy input to separate outputs.	197
Functions Related to Applying Functions to a Cell Array	197
Functions Used with Cell Array Conversion	198
Function.....	198
Description	198
cell2struct	198
Convert cell array into struct array.....	198
struct2cell	198
Convert struct array into cell array.....	198
mat2cell.....	198
Divide matrix into cell array of matrices.....	198
cell2mat.....	198
Convert cell array of matrices to single matrix.....	198
num2cell.....	198
Convert numeric array to cell array	198
26. copyfile	198
27. clear	200
БИБЛИОГРАФИЯ	204

Введение

Для решения задач линейного программирования (линпрогзадач) в Матлабе имеется функция `linprog(...)`, обеспечивающая вычисление глобального минимума линейной целевой функции (ц-функции) при заданных линейных ограничениях-неравенствах и ограничениях-равенствах в области определения ц-функции, задаваемой нижними и верхними границами изменения оптимизационного вектора x_0 – см. разд. 2.2. Совокупность ц-функции, ограничений (неравенств и равенств), нижней и верхней границ для вектора x_0 , команды изменения уставок (при необходимости) функции `linprog(...)` относительно номинальных и самой функции – в форме векторных и матричных объектов, записанных в текстовый М-файл, является простейшим линпрогминимизатором, рекомендуемым в справке Матлаба. Если вектор коэффициентов ц-функции умножить на «-1» и ввести дополнительную команду умножения на «-1» оптимизационное значение ц-функции результата, то получится линпрогмаксимизатор. Результаты вычислений с такими линпрогоптимизаторами представлены на рис. 2.2.

Достоинства таких оптимизаторов – простота. Ничего лишнего, только то, что входит в список аргументов функции `linprog(...)`.

Недостатки. Их несколько, и они серьёзные.

- 1) Нестандартный формат результата (выходные величины – в форме отдельных объектов, а не в одной таблице), не обеспечивающий прямую вставку результата в текстовый документ – см. рис. 2.2.
- 2) Необходимость ручной нумерации имён условий, выходных переменных и объектов при выполнении нескольких задач оптимизации одного типа в одной Матлаб-сессии с сохранением условий всех задач.
- 3) Необходимость ручного наименования объектов по типу оптимизации для их различения в Вычислительном пространстве (Вычпространстве, Workspace – по Матлабу).
- 4) Сохранение величин результата оптимизации (x_0 , y_0 , e_0 – флага оптимизации) только в виде отдельных объектов в Вычпространстве, затрудняющих их совместный вызов в Командное окно (Комокно) и делающим практически невозможным сравнительный анализ результатов нескольких линпрогоптимизаций. Эта же причина препятствует проведению сравнительного анализа вне сессии Матлаба.
- 5) Отсутствует возможность автоматического создания базы данных линпрогоптимизаций (результаты и породившие их условия) в рамках Матлаба.
- 6) В функции `linprog(...)` не предусмотрена возможность линпрогоптимизации с параметрами.
- 7) Отсутствуют средства коррекции условий в случае неуспеха оптимизации.

В целях устранения указанных недостатков предложены ряд линпрогоптимизаторов, различающихся по количеству устраняемых недостатков и соответственно – по сложности реализуемых их программ. Программы написаны с использованием команд Матлаба. Список этих команд с описанием их свойств (копии из Справки Матлаба) приведен в Приложении 3.

Первый такой оптимизатор-линпрогуказание в виде 2 шаблонов (для минимума/ максимума) описан в разд. 2.4.1, рис. 2.3. В нём устранены недостатки 3 и 7. Устранение последнего связано с введением в программы указаний выходных объектов `ngm/ ngma` и `gvm/ gvma` (для мин/ макс). Каждый объект является 2-столбцовой матрицей, первый столбец которой равен произведению матрицы соответствующего ограничения на оптимизационный вектор, а 2-й – вектор-столбец правой части ограничения. Использование этих объектов для коррекции условий линпрогоптимизации показано в разд. 2.5. Поскольку вывод объектов возможен только в Комокно или в текстфайл, процедура коррекции представляется излишне трудоёмкой, особенно для таблиц, сравнимых с высотой экрана монитора из-за невозможности одновременного и постоянного выделения неудовлетворяющих ограничению строк.

Следующий вид оптимизатора – Универсальный линпрогоптимизатор с переключательным вводом типа оптимизации и автоматической уникальной нумерацией (по типу оптимизации) объектов условий и выходных данных – разд. 2.4.2. В нём устранены следующие недостатки исходного оптимизатора: 2-й, 3-й и 7-й. Этот оптимизатор состоит из 2 объектов: универсального линпрогуказания `LinProgUnivUkz` (рис. 2.4) и универсального линпрогоптимизатора `LinProgUniversAvtomatMU` (рис. 2.5), вызываемого для оптимизации в предпоследней строке линпрогуказания. В линпрогуказании сосредоточены все настройки оптимизатора, включая ввод условий оптимизации и изменение уставок функции `linprog(...)` относительно номинальных. Собственно же линпрогоптимизатор `LinProgUniversAvtomatMU` не настраивается при вычислениях, что обеспечивает защиту кода

программы (60 строк) от неумышленных повреждений, могущих возникнуть при вводе новых (особенно больших) условий оптимизации.

Третий (и последний вид) оптимизатора, предлагаемый для непараметрической линпрогооптимизации – это связка универсального итогтабличного линпрогуказания `LinProgUnivUkz_Tab` (рис. 2.12) и универсального итогтабличного линпрогооптимизатора `LinProgUniversAvtomatMU_Tab` (рис. 2.13) – разд. 2.4.4. В нём устранены следующие недостатки простого оптимизатора: 1-й, 2-й, 3-й, 4-й, 5-й наполовину и 7-й. Шестой недостаток устраняется в другой категории линпрогооптимизаторов – параметрических.

Половинчатость устранения 5-го недостатка состоит в следующем. Результат оптимизации сохраняется в итогтаблице в `html`-формате в действующей папке, а условия оптимизации – в виде отдельных объектов в Вычпространстве, т.е. совершенно в другом хранилище, что делает практически невозможным сравнительный анализ результатов-условий ряда задач оптимизации. В этом составном оптимизаторе все настройки производятся в указании `LinProgUnivUkz_Tab`, за исключением 2 одноразовых (при вставке в действующую папку обоих объектов оптимизатора), заключающихся в формировании 2 шаблонов итогтаблиц (для мин/ макс), выполняемых в линпрогооптимизаторе `LinProgUniversAvtomatMU_Tab` (см. разд. 2.4.4.2).

При разработке однопараметрических оптимизаторов был изменён подход к сохранению условий оптимизации каждой задачи. Если в непараметрических оптимизаторах величины условий из Вычпространства однозначно отражают условия оптимизации, то в случае параметрической оптимизации это не так: в них нет данных о местах размещения параметра в величинах условий и его структуре. Поэтому условия сохраняются в автоматически создаваемой копии линпрогуказания одновременно с генерацией итогтаблицы оптимизации в действующей папке. Имя копии совпадает с именем файла итогтаблицы.

В монографии предлагаются 2 вида *p*-линпрогооптимизаторов (однопараметрических оптимизаторов): с ручной нумерацией объектов (ручной) и универсальный. Ручной оптимизатор может быть двух типов: *p*-линпрогминимизатор и *p*-линпрогмаксимизатор. Имена переменных, сохраняемых в Вычпространстве, в каждом из двух *p*-оптимизаторов свои для предотвращения замещения одних другими при запусках в одной Матлаб-сессии разных *p*-оптимизаторов.

Ручной *p*-линпрогминимизатор `pLinProgMinRuch.m` (разд. 3.1, рис. 3.1) включает группу строк, свойстванных линпрогуказанию (указстроки), и в двух предпоследних строках – вызовы текстфайла `pTabLinProgRuchOtladHtm` построителя отладочной `html`-таблицы с объектами `prnmi1`, `prvmi1` (разд. 3.4.1.3, рис. 3.6) и текстфайла `pLinProgRuchP1OptTabFormer` универсального формирователя `html`-итогтаблицы *p*-линпрогооптимизации и копии *p*-линпрогминимизатора с именем файла итогтаблицы (разд. 3.5.2, рис. 3.10). В нём отсутствуют все недостатки простого линпрогооптимизатора, за исключением 2-го. Если предполагается решать несколько задач *p*-линпрогооптимизации с разными условиями в одной Матлаб-сессии, то для предотвращения путаницы с условиями их следует нумеровать номером «своей» задачи. В указстроках сосредоточены все настройки *p*-минимизатора, в том числе и функция `linprog(...)`, за исключением одноразовых настроек `html`-шаблонов отладтаблицы и итогтаблицы. Настройки шаблонов обеих таблиц производятся один раз при вставке программ построителя `pTabLinProgRuchOtladHtm` и формирователя `pLinProgRuchP1OptTabFormer` в действующую папку согласно разделам соответственно (3.4.2) и (3.5).

Ручной *p*-линпрогмаксимизатор `pLinProgMaxRuch.m` (разд. 3.1, рис. 3.2) по структуре совпадает с предыдущим *p*-оптимизатором, отличаясь только значением идентификатора типа оптимизации, дополнительной командой изменения знака оптимального значения *c*-функции и именами величин, отражающих максимизацию.

Универсальный *p*-линпрогооптимизатор `pLinProgTabGenOptimizer` (разд. 3.6.1, рис. 3.12) по структуре совпадает с любым из двух ручных *p*-оптимизаторов. В двух предпоследних его строках находятся имена текстфайлов: формирователя *p*-отладтаблицы `pLinProgUnverOtlad` (разд. 3.6.3, рис. 3.17) и построителя *p*-итогтаблиц `pLinProgUnverP1TabStroyker` (разд. 3.6.2, рис. 3.15), обеспечивающего копирование файла этого оптимизатора с именем файла *p*-итогтаблицы в действующую папку.

В универсальном *p*-оптимизаторе имена переменных носят шаблонный вид, не отражающий тип оптимизации. В Вычпространство они копируются оптимизатором напрямую, а выходные переменные функции `linprog(...)`, параметр и отладочные (выходные) объекты `rng/ rgv` копируются вторично с именами, отражающим тип оптимизации. В этом оптимизаторе одни и те же условия можно использовать для вычисления *p*-минимума и *p*-максимума путём ручного переключения типа оптимизации. При производстве ряда (или массовых) *p*-

линпрогоптимизаций преимущество универсального перед ручным в отношении трудоёмкости многократно возрастает. Настройка html-шаблонов отладочных и итоговых таблиц для этого р-оптимизатора приведена в разд. 3.6.3 и 3.6.2.

В универсальном р-линпрогоптимизаторе `pLinProgTabGenOptimizer` полностью устранены все 7 недостатков упомянутого выше простого оптимизатора, включая и второй. В нём нет нумерации объектов оптимизации, и шаблонные переменные получают значения, зависящие от решаемой задачи. Но чтобы переменные разных задач не мешали друг другу, в первых двух строках оптимизатора помещены команды очистки всех переменных предыдущего запуска, т.е. каждая задача работает только со своими данными – чужие отсутствуют.

Для того чтобы читатель почувствовал вычислительные возможности линпрогоптимизаторов, обеспечиваемые вычислительной мощностью функции `linprog(...)`, в разд. 2.3 и 2.4.3 показаны результаты поиска линпрогоптимума для оптимизационных векторов с количеством компонент 5000 и 7000. Например, для задачи поиска максимума, в которой матрица A (ограничений-неравенств) размером 7000×7000 имеет в верхнем левом углу содержит 100×50 ненулевых элементов (остальные – нули), а матрица A_{eq} (ограничений-равенств) того же размера в левом верхнем углу содержит 300×500 ненулевых элементов, время решения на обычном ноутбуке составило 1,97 с – разд. 2.4.3.2, рис. 2.11.

Для самостоятельного изучения принципов работы линпрогоптимизаторов, их настройки и приобретения навыков решения линпрогозадач, включая коррекцию оптимизационных условий, в монографии публикуется детализированная методика решения примеров (разд. 4), две категории из трёх которых имеют размерность $5 \dots 10$, а в оставшейся – $1000 \dots 8000$. Исходные данные для примеров имеют 90 вариантов, и помещены в разд. 5. Чтобы знание инструментов линпрогоптимизации не носило абстрактный характер, в разделе 7 помещены 80 тематических задач из различных сфер хозяйственной деятельности. Прежде чем решать задачу, следует составить её математическое описание, которое переводится в линпрогозадачу, решаемую в заранее подготовленной Матлаб-среде (включая создание и настройку шаблона итоговой таблицы решения) посредством непараметрических универсальных итоговых линпрогоуказаний `LinProgUnivUkz_Tab` (разд. 2.4.4.1, рис. 2.12) и линпрогоптимизатора `LinProgUniversAvtomatMU_Tab` (разд. 2.4.4.1, рис. 2.13, А). Методика решения приведена в разд. 6, образец решения – в Приложении 1.

Для тех, кто не знаком или мало знаком с Матлабом, в разд. 1.1 даётся детализированное описание рабочего пространства Матлаба, а в разд. 1.2 – поэтапная последовательность организации вычислительной среды, ввода данных, выполнения вычислений и сохранения их результатов.

1. Инструменты линейного программирования Матлаба linprog и bintprog

1.1. Рабочее пространство Матлаба

Ключевые термины раздела 1.1

ВЫЧИСЛИТЕЛЬНАЯ СЕССИЯ	– Набор вычислительных процедур и результатов, введенных и полученных Пользователем за один сеанс работы с Матлабом.
ВЫЧПРОСТРАНСТВО (WORKSPACE)	– Окно, в которое помещаются Операционные объекты вычисления, обрабатываемых в файлах Действующей папки (Current Directory): имена переменных, структур, таблиц, массивов с их значениями. Сохраняется Пользователем в виде файла с расширением «.mat» (в мат-файле).
ДЕЙСТВУЮЩАЯ ПАПКА	– Папка, находящаяся в окне Current Directory, в которую Матлаб записывает все файлы, сохраняемые Пользователем. Выбирается Пользователем через поле поиска окна Действующей папки или через кнопку «Browse for folder»  на панели инструментов Рабочего стола Матлаба.
ИНЖЕНЕРНЫЙ ФОРМАТ ТАБЛИЦЫ ИТВ (ИНЖ-ФОРМАТ)	– Формат ИТВ, в котором устранены знаки и недостатки ячейкового типа данных и форматирования по умолчанию, мешающие восприятию данных таблицы. Таблица в этом формате создается Пользователем по ИТВ в ячк-формате с использованием матлабовского Генератора отчетов (см. п. 33 Правил из разд. 1.4 и разд. 1.7). В Матлабе реализуется в 2 вариантах: RTF-формате и html-формате.
ИТОГОВАЯ ТАБЛИЦА ВЫЧИСЛЕНИЙ (ИТВ)	– Таблица вычислений выражения (или функции) с параметрами, в которой каждое значение выражения напрямую визуально связано с теми значениями аргументов и параметров, которые участвовали в его вычислении. Создается программно Пользователем в ячейковом формате (ячк-формате). См. п. 33 Правил из разд. 1.4.
КОМАНДНОЕ ОКНО (COMMAND WINDOW)	– Центральное окно Рабочего стола Матлаба, в котором отражаются: ввод данных, вычислительные процедуры, отладка М-файлов, настройка встроенных функций и выдача справок с их описанием, производимые в нём и в других окнах Рабочего стола. Оно может также служить в качестве простого и программируемого калькулятора.
КОНТЕКСТ-КОМАНДА	– Команда Контекстного меню, т.е. меню, вызываемого щелчком правой клавишей мышки на рассматриваемом объекте (контексте).
ЛКМ, ПКМ	– Соответственно левая и правая кнопки мышки
МАТ-ФАЙЛ	– Файл Матлаба с расширением «.mat», в котором Пользователем сохраняются Операционные объекты вычислений с файлами из одной или нескольких папок тематических вычислений. Не регистрируется в Матлабе.
М-ФАЙЛ	– Файл Матлаба с расширением «.m»
М-ФАЙЛ-ФУНКЦИЯ	– М-файл, у которого 1-я строка является ключевой, т.е. состоит из ключевого слова function и определения функции; для функции с 2 выходными переменными pr1 и pr2 1-я строка имеет вид: <code>function [pr1,pr2] = f_name(arg1,arg2, ... ,argN)</code> . При этом имя файла должно совпадать с именем функции, т.е. быть <code>f_name.m</code> . Файл для работы должен быть зарегистрирован в Матлабе.
ОПЕРАЦИОННЫЕ ОБЪЕКТЫ (ОПЕРОБЪЕКТЫ)	– Переменные и операнды вычислений в Матлабе. Сохраняются Пользователем в составе мат-файла (Вычпространстве).
ОПРЕДЕЛЕНИЕ ФУНКЦИИ	– Выражение, состоящее из левой (список выходных переменных в квадратных скобках; в случае одной переменной скобки отсутствуют) и правой (имя функции со списком аргументов в круглых скобках) частей, соединенных знаком равно. Например, <code>outvar = f_name(arg1,arg2, ... ,argN)</code> .
ПРИСВАИВАНИЕ	– Выражение, состоящее из левой (имя переменной) и правой (массив значений или математическое выражение) частей, соединенных знаком равно («=»).
СЕССФАЙЛ (SCRIPT FILE)	– М-файл с протоколами вычислительных сессий, у которого отсутствует Ключевая строка. Содержит наборы данных, команд, функций, командных указаний и результатов вычисления, структурированных Пользователем по вычислительным сессиям. Не регистрируется в Матлабе.
ЯЧЕЙКОВЫЙ ТИП ДАННЫХ (ЯЧК-ФОРМАТ, ФОРМАТ) CELL-ФУНКЦИОНАЛЬНАЯ КОМАНДА	– Структурной единицей этого типа является ячейка (cell), в которую можно вставлять любые другие типы данных, включая и сами ячейки. Дает возможность формировать, преобразовывать и производить вычисления с массивами разнотипных данных. Массив из N пустых ячеек вводится командой <code>cell(N)</code> . – Объект Матлаба, порождённый определённой функцией и обеспечивающий при запуске вычисление согласно назначению этой функции и выдачу результата

та в соответствии с набором выходных переменных объекта. Для функции из определения М-файл-функции можно указать следующие команды, реализуемые в Матлабе: `[pr1,pr2] = f_name(arg1,arg2, ... ,argN)`, `pr1= f_name(arg1,arg2, ... ,argN)`, и `f_name(arg1,arg2, ... ,argN)`.

Описание пространства

Рабочее пространство Матлаба включает пять окон Рабочего стола и три других окна (см. табл. II.1.1).

Таблица II.1.1

Окно	Назначение окна
1. Командное (рис.1.1.1, В), (Command Window).	Управление процессами в Матлабе. Информационное оповещение о всех процессах и результатах.
2. Действующей папки (Current Directory; в примере – SanRazrab_MU-ZadOtc171007), (рис.1.1.1, Б).	Сохранение файлов, созданных в текущей сессии Матлаба, открытие файлов Матлаба и других приложений (Уорд, Эксель и т.п.), редактирование их имен.
3. Операционных объектов вычислений (Workspace – Вычпространство), (рис.1.1.1, Д).	Сохранение опереобъектов текущей сессии, просмотр и изменение их значений напрямую (при 1...3 значениях) или посредством редактора массивов при множестве значений. С закрытием Матлаба объекты Вычпространства автоматически не сохраняются. Для сохранения объектов Вычпространство следует сохранить кнопкой «Save» в ель иальный файл – мат-файл, создаваемый пользователем.
4. Протокола команд (Command History), (рис.1.1.1, Г).	Автоматическая запись исполняемых команд, функций и объектов из командного окна в рубрики с датой и временем и сохранение их для последующих сеансов, пока пользователь не очистит весь протокол или его часть. Обеспечение возможности Пользователю копирования команд и функций из записи предыдущих сессий в командное окно для работы.
5. Редактора массивов (Array Editor).	Просмотр содержания массивов в табличном формате и редактирование величин.
6. Редактора – отладчика (рис. 1.1.2) (Editor/Debugger).	Создание, редактирование и отладка М-файлов (файлов, содержащих программы вычислений или протоколы вычислительных сессий), а также сохранение, при необходимости, результатов вычислений.
7. Справки (Help Browser).	Просмотр и поиск технической документации для всего семейства средств Матлаба.

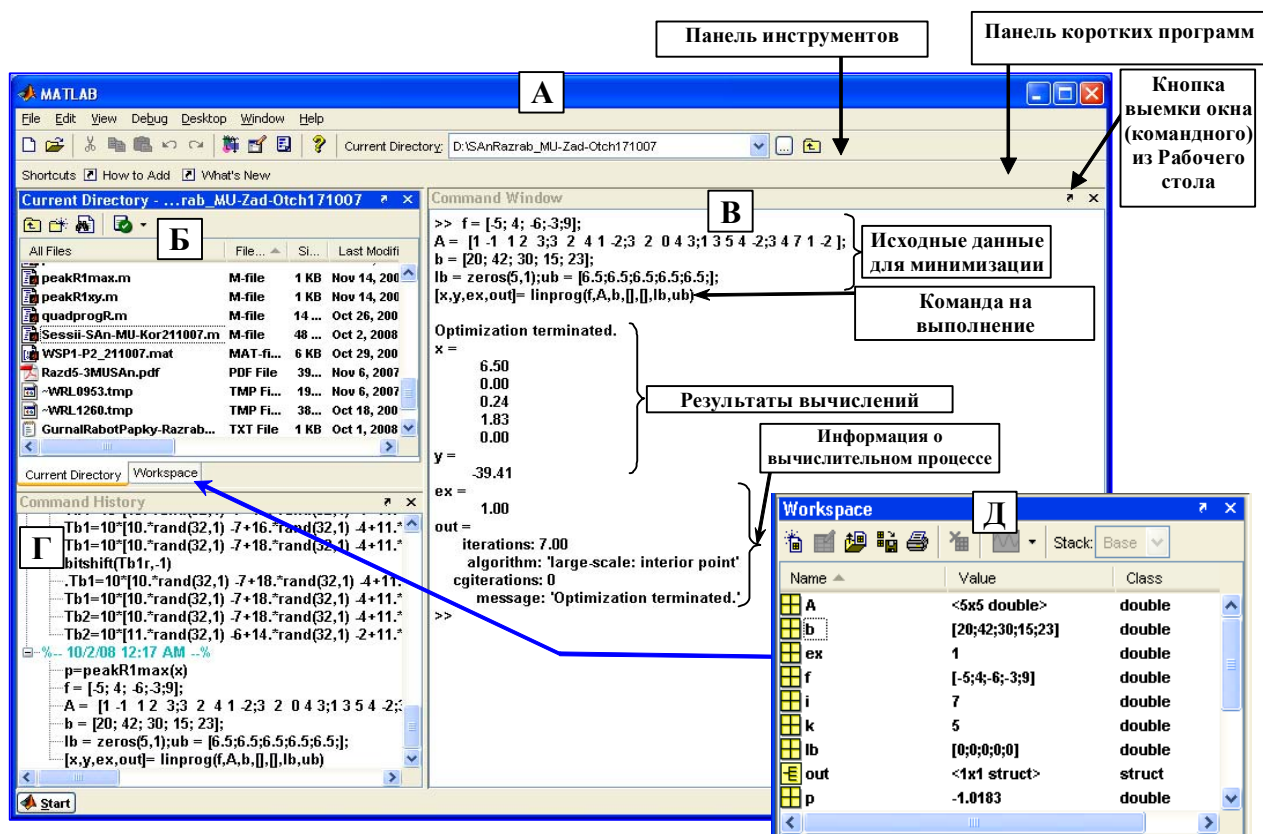


Рис. 1.1.1. Конфигурация Рабочего стола «по умолчанию» (Матлаб 7.0) с элементами пользовательской настройки и примером вычисления минимума линейной функции минимизатором `linprog(...)`. Вызывается командами: «Desktop_Desktop Layout_Default». А – вид Рабочего стола при активном окне Действующей папки (Б). Окна В, Г и Д: Командное, Протокола команд и Вычпространства. Назначение окон – в табл. 1.1.1.

Рабочий стол имеет 5 конфигураций, выбираемых командами: «Desktop_Desktop Layout_Конфигурация», и ряд других, создаваемых вручную и сохраняемых Матлабом меню-командами: «Desktop_Save Layout». Конфигурация сохраняется с определённым именем, например, «Familija_Layout» и помещается Матлабом в список

Desktop Layout после 5 стандартных.

В конфигурации Рабочего стола «по умолчанию» возможно переключение между парами окон (рис.1.1.1): Current Directory (Б) и Workspace (Д). Каждое из окон Рабочего стола может быть освобождено путём нажатия кнопки «Освободить»  в правом верхнем углу окна: окно приобретёт самостоятельность, а кнопка «Освободить» заменится на кнопку «Загрузить» (вернуть окно на место).

Для типовых вычислений следует убрать с Рабочего стола окно Протокола команд (Command History) нажатием на кнопку «х» в правом верхнем углу окна (с целью увеличения места для окна Действующей папки – Current Directory), а папку «Вычпространство» - освободить для независимого от окна Действующей папки елтупа и расширения её поля для увеличения количества столбцов информации об вычислительных объектах. В частности, следует открыть 5 (из 12) следующих столбцов: «Value», «Syze», «Class», «Min», «Max». Открытие столбцов: щелчком ПКМ на строке «Name» в окне Вычпространства вызывается список имен столбцов, и рядом с нужным именем нажатием ЛКМ проставляется галочка.

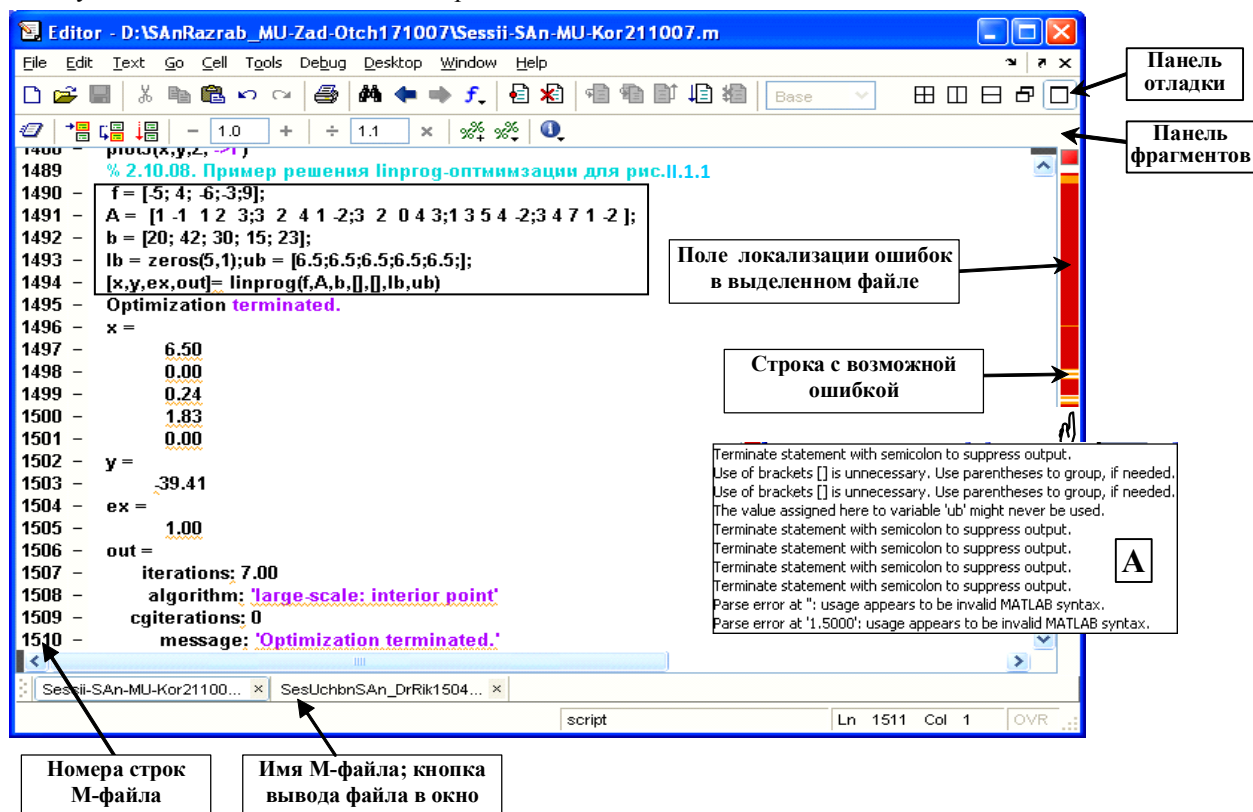


Рис. 1.1.2. Окно Редактора-отладчика с примером линпрогоптимизации (в рамке), решённым в Командном окне рис. 1.1.1, и с полем локализации возможных ошибок. Толщина оранжевой линии в поле соответствует определенному количеству подряд идущих строк с ошибками. Щелчок ЛКМ на линии подводит строку М-файла к указателю. Подвод мышки к линии вызывает появление сообщения типа А с предложениями по ликвидации «ошибок».

Редактор-отладчик предназначен для создания, отладки и ведения М-файлов-функций, сцен-файлов вычисляемых выражений и протокольных. Для оперативного управления отладкой помимо команд меню в Редакторе имеются 2 панели с наборами инструментов: Панель отладки для отладки М-файлов-функций и Панель фрагментов для отладки сцен-файлов и протокольных.

Панель отладки. Кнопки отладки (слева направо)

- Бинокль – поиск фрагмента текста в М-файле и автоматическая замена одного фрагмента на другой.
- Стрелки влево-вправо – перемещение по ключевым словам всех открытых в Редакторе файлов.
-  – приводит список имен функций, сочетающихся с ключевым словом function, и при нажатии ЛКМ на любое имя из списка показать строку с этим именем и вставленным в нее редакционного курсора.
- Установить/удалить точку прерывания – (Установить/Снять контрольную точку).
- Удалить все точки прерывания во всех М-файлах Редактора – (Снять все контрольные точки).
- Пошаговая трассировка М-файла.

- Пошаговая трассировка М-файла с заходом в вызываемые М-файлы (Заходить в блоки).
- Пошаговая трассировка М-файла без захода в вызываемые М-файлы (Пошаговое выполнение).
- Продолжить выполнение программы без трассировки (Продолжить),
- Завершить отладку (Окончить отладку).

Панель фрагментов

- Вычислить фрагмент.
- Вычислить фрагмент и перейти к следующему.
- Вычислить весь файл.
- Кнопки «-», «+», «:», «х» не задействованы.
- %%+ - вставка линейных выделителей фрагментов.
- %%□ - показать список фрагментов файла.
- Показать справочные ссылки о фрагментах.

При написании М-файла в окне осуществляется синтаксический контроль его программы с цветовым выделением синтаксических объектов:

- ключевые слова языка программирования (function, for, end и т.п.) — **синий цвет**;
- операторы, константы и переменные — **черный цвет**;
- комментарии после знака «%» — **зеленый цвет**;
- символьные переменные (в апострофах) — **зеленый цвет**;
- синтаксические ошибки, текст в кириллице — **красный цвет**.

1.2. Схема простого вычислительного процесса в Матлабе

Этой схеме следуют вычисления по линейному программированию, описываемые в нижеследующих разделах. Процесс включает подготовку среды, ввод исходных данных и формирование командования, вычисления, обработку и сохранение результатов.

Этапы процесса.

1.2.1. Подготовка среды

А) Имена папок, подпапок, файлов, переменных должны выполняться английским шрифтом. Наличие русских букв в директории файла, как следует из приведенного выше раздела, является синтаксической ошибкой и приводит к неузнаванию его Матлабом и к отказу открытия файла. Русские буквы в именах переменных, М-файл-функциях и пользовательских командах выделяются Матлабом красным цветом, и Матлаб не вводит их в Вычпространство и не исполняет такие команды, выдавая сообщение в Комокно об ошибке. Русские буквы допускаются в комментариях, начинающихся на каждой строке со знака проценты (%), в названиях строк и столбцов матриц в строковом и ячейковом форматах, а также – в виде значений компонентов и элементов векторов и матриц в тех же форматах.

Б) При открытом Рабочем столе, если подвести указатель к любому неименованому его элементу, то Матлаб выдаст его название или назначение.

В) Должна быть создана папка на диске (в разделе) без операционной системы с именем, например «Familija», в которую будут помещаться файлы, создаваемые в процессе вычислительной работы.

Г) Открывается Матлаб. Если он открывается впервые, то конфигурация Рабочего стола будет «по умолчанию» (см. рис. 1.1.1 для Матлаба 7.0, для Матлаба 7.9 окна Вычпространства и Протокола команд будут находиться справа от Комокна, а окно Действующей папки – слева). В качестве действующей будет показана папка Матлаба с директорией по-умолчанию, например, такой: C:\Users\Polzovatel\Documents\MATLAB. Если Матлаб открывался ранее, то конфигурация Рабочего стола может быть иной, созданной предыдущим Пользователем.

Необходимо создать конфигурацию Рабочего стола по-умолчанию, и Действующей папкой сделать созданную папку «Familija».

Д) Требуемая конфигурация создаётся цепочкой команд из строки меню Рабочего стола (1-й строки, см. рис. 1.1.1): «Desktop_Desktop Layout_Default». Для помещения папки «Familija» в окно Действующей необходимо нажатием кнопки «Browse for Folder»  справа от поля директории вызвать панель поиска папки, через которую и указать папку «Familija». Директория папки появится в поле директории, а имя папки – в заголовке окна

«Current Folder». Нажатием на кнопку «x» удаляется с Рабочего стола окно Протокола команд. Нажатие кнопки «Освободить»  в правом верхнем углу окна «Workspase» извлекает его из Рабочего стола и делает окно самостоятельным.

Е) Создание сессионного файла вычислений (сессфайла), в котором записываются исходные данные, команды, командования, небольшие программы, комментарии, рабочие результаты вычислений. По классификации Матлаба он относится к категории скрипт-М-файлов (текст-М-файлов). Нажатием кнопки «New M-file» (1-я во второй строке) открывается окно Редактора (Editor) с файлом Untitled и с нумерованной 1-й строкой. Клавишей Ввод следует ввести ещё несколько нумерованных строк и сохранить файл командами «File_Save As» с именем, например, «sessFamilija». После нажатия «OK» имя файла sessFamilija.m появится в окне Действующей папки.

Ж) Создание файла Вычпространства (Workspace). Этот файл не само пространство, а инструмент Пользователя для сохранения объектов пространства текущей вычислительной сессии с целью использования их в последующих сессиях. После закрытия Матлаба объекты Вычпространства не сохраняются, и в новой сессии Матлаб устанавливает пустое пространство, которое и следует заполнить из сохранённого файла (двойным щелчком ЛКМ на его имени).

Для создания файла необходимо войти в окно Вычпространства и командами меню «File_Save As» сохранить его с именем, например, «WS_Familija». После нажатия кнопки «OK» на панели сохранения в окне Действующей папки появится имя созданного файла «WS_Familija.mat».

1.2.2. Ввод исходных данных и команд

3) Ввод исходных данных и команд в Матлаб. Это можно организовать в двух вариантах: в Командное окно или в сессфайл. В 1-м варианте после ввода данных и команд (текста) для их обработки нажимается клавиша «Enter», и Матлаб выдаёт результат ниже набранного текста. Если в тексте была ошибка, то вместо результата выдаётся сообщение об ошибке. Так как содержимое Комокна не сохраняется ни по умолчанию, ни в файле, то такой способ ввода и работы подходит только к простейшим вычислениям. Кроме того, вычисления в Комокне функционально ограничены – см. п. И).

Основной вариант ввода – второй, в сессфайл, названный в п. Е) как sessFamilija.m. Ввод ряда объектов командования линпрог-минимизации линейной функции $fc(x) = -5x_1 + 4x_2 - 6x_3 - 3x_4 + 9x_5$ показан на рис. 1.1.2 в сессионный файл «Sess1-San-MU-Kor211007.m» в строки, ограниченные прямоугольной рамкой.

В этом указании введены: 5-компонентные вектор-столбцы (f, b, lb и ub), 5-строчная 5-столбцовая матрица A и функция линпрог-минимизации с упомянутыми объектами в качестве аргументов и с выходными переменными, которые нужно вычислить, – минимизационный вектором x, минимальным значением функции $y = fc(x)$, флагом окончания процесса вычисления ex (exit flag) и текстовым сообщением о параметрах выполненной линпрог-минимизации out.

Если в векторах f, b и ub точку с запятой между компонентами заменить пробелом, то вектор-столбец станет вектор-строкой. В выражении для lb функция zeros(5,1) при вводе генерирует 5-столбчатый вектор, состоящий из нулей. А функция zeros(1,5) генерирует 5-компонентный строка-вектор, состоящий из нулей.

В первых 4 строках команд-указания каждое присваивание заканчивается точкой с запятой, предотвращающей при вводе в Вычпространство появление значений переменной в Комокне. В то же время, в 5-й строке отсутствует точка с запятой, и поэтому линпрог-минимизатор выдаёт выходные данные не только в Вычпространство (см. рис. 1.1.1, Д – показаны ex и out), но и в Комокно.

Важным достоинством ввода данных в сессфайл является одновременный контроль ввода со стороны Матлаба с выдачей меток об ошибках в полосу контроля с информацией о номерах строк с ошибками (выдаётся при наведении курсора на метку). Кроме того, некоторые элементы в строках с «ошибками» (например, знак «=») Матлаб выделяет жёлтым цветом. Замечания Матлаба нужно исправлять и одновременно соотносить их уместность с вводимым контекстом. Например, в 5-й строке команд-указания в рамке на рис. 1.1.2 не поставлена точка с запятой, чтобы были выведены выходные данные в Комокно, но Матлаб в этой строке будет числить ошибку и советовать поставить пропущенный элемент – такой совет надо игнорировать.

Ряд правил набора вычислительных объектов с клавиатуры в Матлабе.

1) Знаком присваивания является привычный математический знак равенства =, а не комбинированный знак «:=», как во многих других математических системах.

2) Для отделения целой части числа от дробной используется точка, а не запятая как в русской математической традиции. Пример: 8.2 – в Матлабе, 8,2 – то же число в русской записи.

3) Чтобы ввести в Вычпространство значения какой-либо переменной необходимо составить выражение присваивания, в левой части которого стоит имя переменной, а в правой – значения (скаляр (одно значение), вектор или матрица). Далее нажать клавишу «Ввод» для набора присваивания в Комокне или выделить присваивание и нажать клавишу f9 (fn+f9 – для ноутбука) при наборе в Редакторе. Выделенное выражение в Редакторе можно также отправить на ввод контекст-командой «Evaluate Selection».

4) Нумерация (индексация) элементов вектора и матрицы в Матлабе начинается с единицы. Индексы элементов указываются в круглых скобках. Например, 3-й элемент вектора f из рис. 1.1.2 равен -6, Отражение этого факта в записи: f(3)= -6. Элемент матрицы индексируется 2 числами: номерами строки (на 1-м месте) и столбца (через запятую на 2-м), на пересечении которых он стоит. Например, 3-й элемент во второй строке (у него - 3-й столбец) матрицы A на том же рисунке равен 4, т.е. A(2, 3)=4.

5) Вектор-строка через свои компоненты записывается в виде строчного выражения в квадратных скобках, в которых компоненты отделяются друг от друга пробелом. Например, 4-компонентный вектор c1 (с компонентами 2; 3,5; 5 и 123) имеет вид: c1=[2 3.4 5 123]. В вектор-столбце копоненты отделяются друг от друга точкой с запятой – см. вектор f на рис. 1.1.2.

6) Двумерная матрица создается в виде выражения в квадратных скобках, в которые последовательно строчно вводятся элементы матрицы. Причем, соседние элементы каждой строки отделены друг от друга пробелом, а строки отделяются точкой с запятой. Например, если в окно Редактора ввести выражение A1 = [0.1 1 -3 4.2; 3 0 1 0.2; 0 2 0 5; 0 0 4 0], выделить и нажать клавишу <F9>, то Матлаб выдаст в Комокно матрицу следующего вида:

```
A1 =
    0.10    1.00   -3.00    4.20
    3.00     0    1.00    0.20
     0     2.00     0    5.00
     0     0     4.00     0
```

7) Комплексное или мнимое число определяется по наличию в его составе констант i или j, которым присваивается значение $\sqrt{-1}$. Комплексное число можно вводить одним из следующих способов: 5+3i; 5+3*I; 5+3j; 5+3*j или 5+3*sqrt(-1) – разные виды одного и того же числа.

8) Арифметические поэлементные операции для массивов отличаются от матричных операций наличием точки перед знаком операции: A.*B (умножение массивов – массивы должны иметь одинаковое количество чисел или один из них должен быть числом), A./B (деление), A.\B (левое деление), A.^B (возведение в степень).

Точка перед «+» и «-» не ставится!

9) Приоритеты выполнения операций (операции высшего приоритета выполняются первыми, низшего – вторыми).

Логические операции выше арифметических. Возведение в степень выше умножения и деления. Умножение и деление выше сложения и вычитания.

Для изменения приоритета операций используются круглые скобки. Степень вложения скобок не ограничена.

10) Встроенные элементарные функции (например, sin) записываются строчными буквами и их аргументы указываются в круглых скобках после имени без пробела. Если аргументов несколько, то они отделяются запятыми. Раздел элементарных функций Матлаба включает 79 типов, разделённых на 5 категорий: тригонометрические и гиперболические – 38 [19], экспоненциальные – 13, обработка комплексных величин – 12, округление и вычисление остатка – 7, дискретная математика – 9. Тригонометрические функции существуют в двух видах: без буквы «d» в конце имени и с этой буквой (d – degrees, градусы). У функций без «d» - аргумент в еданах, с этой буквой – в градусах. Пример: sin(x) и sind(x). Имена некоторых тригонометрических и гиперболических в Матлабе отличаются от общепринятых. Например, tg(x) записывается как tan(x), косинус гиперболический ch(x) записывается как cosh(x). Для правильной записи следует посмотреть список этих функций в Справке [19].

11) Переменные в круглых скобках после имени объекта считаются в Матлабе как аргументы только для встроенных функций и для функции пользователя, сформированной в виде М-файла-функции и зарегистрированной в Матлабе. В остальных случаях эти переменные считаются Матлабом индексами имени объекта с соответствующей индексам обработкой при вводе. Поэтому определение функции в виде присваивания нельзя создавать ни в Комокне ни в сессфайле: при вводе в Матлаб набранного такого присваивания последний выдаёт в Комокно сообщение о запрете его создания в упомянутых контекстах.

12) М-файл-функция представляет собой конструкцию, дающей Пользователю возможность встраивать в Матлаб собственные функции. Шаблон простой М-файл-функции имеет следующий вид.

Function [pr1, pr2...prN1] = funname(ar1,ar2 ... arN2) – шаблон обязательного текста 1-й строки М-файла (pr1, pr2...prN1 – выходные переменные, число которых N1, ar1,ar2 ... arN2 – аргументы функции, число которых N2), function – ключевое слово, отображаемое Матлабом синим цветом.

- % Основной комментарий

- Набор исходных данных. Каждое присваивание в наборе должно заканчиваться точкой с запятой.

Pr1 = выходное выражение 1;

pr2 = выходное выражение 2;

.....

prN1 = выходное выражение N1;

Если выходная переменная одна, то квадратные скобки не ставятся.

Основной комментарий – это справочные сведения о назначении функции, о допустимых значениях её аргументов, о видах функциональных команд на основе данной функции и т.п. Эти сведения выдаются Матлабом в Комокно командой «help funname» - вывод основного комментария функции funname.

Имя функции «funname» должно начинаться обязательно с буквы, его длина не должна превосходить 63 знаков, и оно должно совпадать с именем сохраненного М-файла (funname.m) без расширения m. Аргументами функции могут быть: скаляры, векторы, матрицы. Все аргументы в векторном или в матричном виде должны иметь одинаковый размер при покомпонентных вычислениях. Для вычисления с различающимися размерами и типами (вектор или матрица) следует использовать цикл for с вставленной в него функцией funname – см. на пример, цикл for для параметра P1, от которого зависят аргументы f и A в М-файл-функции linprog на рис. 2.5.

Чтобы функция работала, её нужно зарегистрировать в Матлабе. В окне Редактора командами «File_Set Path» вызывается панель регистрации «Set Path». Нажатием кнопки «Add Folder» вызывается окно поиска, в котором указывается папка с регистрируемой функцией, а затем на панели регистрации нажимается кнопка «Save» и кнопка «Close», завершающая регистрацию.

Функция запускается на вычисление при наличии значений всех её аргументов в Вычпространстве или в виде набора присваиваний для всех аргументов, расположенных в строке слева от функциональной команды или в строках над ней. Варианты запуска функции (функциональных команд) согласно её определения: [pr1, pr2...prN1] = funname(ar1,ar2 ... arN2), pr1 = funname(ar1,ar2 ... arN2), [pr1, pr2] = funname(ar1,ar2 ... arN2), [pr1, pr2,pr3] = funname(ar1,ar2 ... arN2) ... и funname(ar1,ar2 ... arN2). Во всех случаях, за исключением последнего, результаты выданы с именами используемых выходных переменных. Последний вариант команды аналогичен 2-у в списке, но выдаётся он не с именем переменной pr1, а с системной переменной ans.

1.2.3. Вычисления и сохранение результатов

И) Вычисления в Комокне. Пользователь активизирует его (переходит в окно, вставляет курсор в место ввода, помеченное символом >>), набирает на клавиатуре вычисляемое выражение, редактирует его (если нужно) и завершает ввод нажатием клавиши Enter. Матлаб выдаст результат, присвоив ему переменную с именем ans (ans = результат), если имя результату не было введено заранее: в виде имени вычисляемой переменной без знака присвоения или в виде нескольких вычисляемых имён, отделяемых друг от друга запятой.

При наборе данных в Комокне Матлаб не контролирует правильность ввода: ошибка отмечается Матлабом после запуска введённого контекста. Вычисленный контекст нельзя редактировать. Для редактирования текст копируется в текущую точку ввода Комокна.

Если вставляемый текст набирается вручную и занимает более одной строчки, то переход к следующей строке производится нажатием 2 клавиш: Shift+Enter. Нажатием клавиш ▲, ▼ можно построчно вставлять в новое место ввода ранее вычисленный текст. Многострочный текст так нельзя вставлять, т.к. очередная вставлен-

ная строка замещает прежнюю.

Так как Комокно не сохраняет введенные данные и полученные результаты, то весь вычислительный контекст следует сохранять в сесфайле.

К) Вычисления в окне Редактора. Вычислительный текст набирается в сесфайле в виде командования, содержащего исходные данные и команды или функции по их обработке. Исходные данные должны следовать раньше команд и функций. При наборе с клавиатуры следует обращать внимание на заметки Матлаба по вводу, выдаваемых в полосу замечаний набором линий, и в виде выделения сомнительных объектов желтым цветом. При подводе курсора к такому месту Матлаб выдаст рекомендацию по устранению замечания.

Набранный текст выделяется, и клавишей f9 (для ноутбука – клавишами fn+f9) запускаются вычисления. Запускать вычисления при выделенном контексте можно также контекст-командой «Evaluate Selection». Матлаб копирует в Комокно выделенный текст, и ниже его помещает результаты вычислений или сообщение об ошибке, если таковая была в командовании.

Л) При необходимости результаты копируются в сесфайл, выделяются и командами Редактора «Text_Comment» закомментируются, т.е. в начале каждой из выделенных строк проставляется знак процентов (%). Если этого не сделать, то Матлаб выставит множество замечаний по тексту результата: точки с запятой не проставлены, многие объекты не являются правильными выражениями Матлаба и т.п.

В течение вычислительной сессии следует периодически пересохранять сесфайл и файл Вычпространства. При проблемном вводе возможно зависание Матлаба с потерей несохраненной части результатов работы. По умолчанию Матлаб сохраняет копии файлов с расширением «.m», приписывая копии расширение «.asv». Минимальная периодичность сохранения 1 мин. В случае утраты М-файла вместо него можно использовать копию, сменив расширение asv на m.

Копирование файла можно отключить: вызвать командами «File_Preferences_Autosave» вкладку автосохранения и убрать галочку в квадрате «Enable autosave in the MATLAB Editor». На этой же вкладке можно установить другую периодичность автосохранения в прямоугольнике «Save every:».

1.3. Функции линейного программирования

В пакете Матлаб есть две функции линейного программирования: linprog(...) и bintprog(...). По умолчанию обе являются минимизаторами, но могут быть и максимизаторами, если ц-функцию перед запуском решения умножить на -1, и в результате решения найденное значение ц-функции также умножить на -1. Первая функция носит название: «Функция линейного программирования» [17], вторая: «Функция линейного программирования в целых числах с двоичным ц-вектором».

Символическая запись решаемой проблемы для функции linprog(...):

$$\min_x f(x) = \min_x (f^T \cdot x) \text{ при условиях } \begin{cases} A \cdot x \leq b, \\ Aeq \cdot x = beq, \\ lb \leq x \leq ub, \\ x = x0 \text{ (по выбору)}, \end{cases} \quad (1.1)$$

где f^T , b , beq , lb , ub , x , $x0$ – векторы; f^T – вектор-столбец коэффициентов ц-функции $f(x)$, A , Aeq – матрицы. Размерность: векторов x , f^T , lb , ub , $x0$ – n ; векторов b , beq – n . Размер матриц A , Aeq – $n \times n$.

Аналогичный вид записи для bintprog(...):

$$\min_x f(x) = \min_x (f^T \cdot x) \text{ при условиях } \begin{cases} A \cdot x \leq b, \\ Aeq \cdot x = beq, \\ x \text{ – двоичный вектор,} \\ x = x0 \text{ (по выбору)}, \end{cases} \quad (1.2)$$

где f^T , b , x , $x0$ – векторы; f^T – вектор-столбец коэффициентов ц-функции $f(x)$, A , Aeq – матрицы. Размерность: векторов f^T , x , $x0$ – n ; векторов b , beq – n . Размер матриц A , Aeq – $n \times n$. Векторы x , $x0$ – двоичные: состоят из нулей и единиц; остальные объекты включают только целые числа.

В монографии функция bintprog(...) далее не рассматривается.

2. Функция линейного программирования `linprog(...)` и линпрогооптимизаторы на её основе

2.1. Ключевые термины раздела 2.

LINPROG(...)	– Функция для решения задач линейного программирования, т.е. для поиска оптимума линейной ц-функции. Имеет 10 вариантов записи (для Матлаб R 2009b). Вариант с наибольшим числом аргументов и выходных элементов: <code>[x,fval,exitflag,output,lambda]= linprog(f,A,b,Aeq,beq,lb,ub,x0,optimset(...))</code> . См. табл. 2.1 в текущем разделе Методуказаний. Linprog вычисляет глобальный оптимум!
ВЫЧПРОСТРАНСТВО, WORKSPACE	– Окно, в котором хранятся вычислительные объекты текущей сессии Матлаба. Для повторного использования вычобъектов в новой сессии окно ель изается Пользователем в файле с расширением «.mat» (мат-файле) в действующей папке, и этот файл активизируется двойным щелчком ЛКМ на его имени.
ЗАДАЧА ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ, ЛИНПРОГЗАДАЧА	– Поиск оптимума линейной ц-функции – см. например рис.2.1.
ЗАКОММЕНТИРОВАТЬ СТРОКУ/ СТРОКИ КО- МАНД	– Поставить знак проценты (%) в 1-й позиции строки/ строк. В редакторе-отладчике существует команда закомментирования выделенных строк: «Text_Comment». Закомментированные строки не выполняются Матлабом при запуске программы. Закомментировать можно группу команд в строке, поставив вручную перед ней знак %.
ИНЖ-ТАБЛИЦА	– Таблица в формате rtf (Уорд-таблица) или html.
КРУПНО-РАЗМЕРНОСТ- НЫЙ АЛГОРИТМ, КР- АЛГОРИТМ, LARGE-SCALE ALGORITHM	– Алгоритм для оптимизации ц-функци большой размерности, когда число компонентов ц-вектора равно нескольким десяткам, сотням и более штук. Алгоритм опирается на свойство разреженности матриц из описания задачи линейного программирования, в частности, разреженными должны быть матрицы A и Aeq (см. термин <code>linprog(...)</code>). Алгоритм может применяться для задачи средней и малой размерностей, если матрицы её описания разреженные. КР-алгоритм в <code>linprog</code> называется Линейный для Внутренних Точек Оптимизатор (ЛВТ-оптимизатор) – LIPSOL (Linear Interior Point Solver). Алгоритм действует в <code>linprog</code> по умолчанию. См. разд. 2.3 по вычислению 5000-компонентного минимизационного вектора с матрицей A в 25 млн элементов с числом ненулевых элементов 2500 или разд. 2.4.3 по вычислению 7000-компонентного минимизационного вектора с матрицей A в 49 млн элементов. Время решения в 1-м случае составляло 15,62 с, а во 2-м – 0,372 с.
ЛИНЕЙНАЯ Ц-ФУНКЦИЯ	– $f_c(x) = f_1 \cdot x_1 + f_2 \cdot x_2 + \dots + f_N \cdot x_N = f \cdot x$ – скалярное произведение векторов f и x.
ЛИНПРОГ- ОПТИМИЗАТОР	– Командное указание с использованием ПМФ <code>linprog</code> для решения ЛП-задачи
ЛИНПРОГОПТИМУМ, ЛИНПРОГМИН, ЛИН- ПРОГМАКС	– Оптимум задачи линейного программирования.
ОПТИМИЗАЦИОННОЕ УКАЗАНИЕ, ОПТИМ- УКАЗАНИЕ	– Набор исходных данных для оптимизационных вычислений <code>(f,A,b,Aeq,beq,lb,ub,x0,optimset(...))</code> – все или отдельные элементы: f – обязательно, A и b, Aeq и beq вставляются (исключаются) парой), Управляющие параметры, линпрогооптимизатор соответствующего вида и набор выходных объектов, отражающий результаты оптимизации или функции от них, например <code>[A*xmin,b']</code> – сравнение фактических ель иза левых частей неравенств (4.3) с вектором b' их правых частей (см. рис. 2.3, 2.12).
ПАРАМЕТР Ц-ФУНКЦИИ	– Скалярная или векторная переменная, для значений которой вычисляются по ц-переменной оптимумы ц-функции командным или пользовательско-программным способами.
ПОИСКМИНФУНКЦИЯ, ПМФ, ОПТИМИЗАТОР	– Функция Матлаба, выполняющая поиск локального минимума (максимума при умножении ц-функции на «-1») Целевой функции (ц-функции) для заданных условий. Функция может выдавать: координаты минимума и значение ц-функции в нём, характер окончания поиска в форме значения переменной <code>exitflag</code> , характеристику процесса поиска в форме значений переменной <code>output</code> и описание ряда сопутствующих вычислительных объектов. Таких функций в Матлабе 10. Они составляют Инструментарий оптимизации – Optimization Toolbox.

ПОЛНАЯ МАТРИЦА, FULL MATRIX	– Матрица с преимуществом ненулевых элементов. Операции с матрицами производятся без учета нулей в них.
ПТ, ПУНКТ (ПИКА)	– Единица измерения высоты шрифта, толщины линии или размера маркера. 1 пт = 0,353 мм.
РАЗРЕЖЕННАЯ МАТРИЦА, SPARSE MATRIX	– Матрица с преимуществом нулей, т.е. нулей может быть в несколько раз (зачастую в 10, 100 и более раз – см. разд. 2.3) больше ненулевых элементов. В Матлабе есть набор операций, позволяющий хранить и обрабатывать только ненулевые элементы разреженных матриц, что значительно сокращает объём памяти и время выполнения программ с ними. См. пример на рис. 2.1.
РАСКОММЕНТИРОВАТЬ СТРОКУ/ СТРОКИ КОМАНД	– Убрать знак проценты (%) в 1-й позиции строки/ строк. В редакторе-отладчике существует команда раскомментирования выделенных строк: «Text_Uncomment». Раскомментированные строки будут выполняться при запуске программы.
СЕССИОННЫЙ ФАЙЛ, СЕССФАЙЛ	– Текстовый М-файл (не М-файл-функция), в котором сохраняются команды, указания и результаты вычислительных сессий Пользователя. Типовое имя для файла: «SessFamilijaPolzovatelja». Программа, указания или их фрагмент для вычисления выделяется, и нажимается клавиша F9 (на ноутбуке – fn+f9): Матлаб копирует выделенное в Командное окно, производит вычисления (если выделенный набор не содержит ошибок) и выдаёт в Командное окно результат. При наличии ошибок в наборе в Командное окно выдаётся сообщение об ошибке.
СРЕДНЕ-РАЗМЕРНОСТНЫЙ АЛГОРИТМ, СР-АЛГОРИТМ, MEDIUM-SCALE ALGORITHM	– Алгоритм для оптимизации ц-функции средней и малой размерностей. Использует полные матрицы из описания задачи: нули будут обрабатываться так же, как и ненулевые элементы. В linprog применяются 2 СР-алгоритма: Последовательностного Квадратичного Программирования (ПКВП-алгоритм, Sequential Quadratic Programming – SQP-алгоритм) и Симплекс-алгоритм. Алгоритмы становятся действующими в linprog-оптимизаторе по командам: options = optimset('LargeScale', 'off') – для ПКВП-алгоритма, options = optimset('LargeScale', 'off', 'Simplex', 'on') – для Симплекс-алгоритма.
СТИЛИЗАТОР	– Набор параметров стиля текста, графики или рисунка.
СТОЛБЕЦ СОВЕТОВ	– Столбец у правой границы окна Редактора-отладчика, включающий набор желтых и оранжевых меток. При подводе мышки к метке Матлаб выдаёт номер строки и совет или сообщение об ошибке по поводу текста строки.
УНИВЕРСАЛЬНОЕ ЛИНПРОГУКАНИЕ	– Линпрогуказание LinProgUnivUkz (рис. 2.4), обеспечивающее параметрическое управление типом оптимизации (мин/ макс) и совмещение простой нумерации и уникальной (неповторимой) идентификации всех переменных и объектов (входных/ выходных) совместно с универсальным линпрогоптимизатором LinProgUniversAvtomatMU (рис. 2.5), задействованных в текущем его пуске. Результат вычисления линпрогминимума по этому указанию – на рис. 2.6.
УНИВЕРСАЛЬНОЕ ТАБЛИЧНОЕ ЛИНПРОГУКАНИЕ	– Линпрогуказание LinProgUnivUkz_Tab (рис. 2.12), обеспечивающее параметрическое управление типом оптимизации (мин/ макс), совмещение простой нумерации и уникальной (неповторимой) идентификации всех переменных и объектов (входных/ выходных) и выдачу итогтаблицы результатов пуска совместно с универсальным линпрогоптимизатором LinProgUniversAvtomatMU_Tab (рис. 2.13), задействованных в текущем его пуске. Итогтаблица имеет идентификатор, равный объектам пуска, сохраняется в html-формате и в заголовке имеет перечень всех объектов, задействованных в пуске оптимизации. Результат вычисления линпрогминимума по этому указанию – на рис. 2.14, А (Комокно) и Б (Итогтаблица) .
ЦЕЛЕВАЯ Ц-ОБЛАСТЬ	– Область изменения ц-вектора x. Задаётся границами для компонент x: $lb \leq x \leq ub$ (lb, ub – векторы одинаковой с x размерности), причем нижняя граница, верхняя или обе вместе могут отсутствовать. Если отсутствуют отдельные компоненты границ, то они замещаются значениями: -inf (для нижней границы) и inf (для верхней).
ЦЕЛЕВАЯ ПЕРЕМЕННАЯ, Ц-ПЕРЕМЕННАЯ, Ц-ВЕКТОР	– Скалярная или векторная переменная (аргумент ц-функции, обозначаемый как x), по которой вычисляется оптимумы ц-функции.
ЯЧЕК-ТАБЛИЦА	– Таблица в ячейковом формате Матлаба. Ячейковый формат позволяет объединять (склеивать) массивы с разнородными типами данных. Например, числовые и массивы слов (имён).

2.2. Описание функции линейного программирования `linprog(...)`

Функция `linprog(...)` предназначен для решения задач линейного программирования, т.е. для минимизации линейной ц-функции $fc(x)$ в заданной области ц-переменной x для определенных наборов линейных ограничений. Векторно-матричная запись общей задачи линейного программирования:

$$\min_x fc(x) = \min(f^T \cdot x), \quad f = [f1 \ f2 \dots fN], \quad x = [x1 \ x2 \dots xN] \quad (2.1)$$

в заданной ц-области x

$$lb \leq x \leq ub, \quad (2.2)$$

при ограничениях – линейных неравенствах и линейных равенствах – соответственно:

$$Ax \leq b, \quad (2.3)$$

$$Aeq \cdot x = beq. \quad (2.4)$$

f^T , b , beq , lb , ub , $x0$ – векторы; f^T – вектор-столбец коэффициентов ц-функции $fc(x)$, A , Aeq – матрицы.

Размерность: векторов x , f^T , lb , ub , $x0$ – n ; векторов b , beq $\leq n$. Размер матриц A , Aeq $\leq n \cdot n$.

Для решения задачи (2.1)...(2.4) применяется поискоминфункция `linprog(...)`. В зависимости от вида ограничений, полноты описания заданной области, количества аргументов и выходных переменных команда на поиск минимума имеет 10 вариантов записи – табл. 2.1.

Таблица 2.1. Варианты функции `linprog(...)`

Вариант	Примечание
1. $x = \text{linprog}(f,A,b)$	Линейные неравенства (4.3)
2. $x = \text{linprog}(f,A,b,Aeq,beq)$	Линейные неравенства (4.3) и равенства (4.4)
3. $x = \text{linprog}(f,A,b,Aeq,beq,lb,ub)$	Линейные неравенства (4.3), равенства (4.4) и ц-область (4.2)
4. $x = \text{linprog}(f,A,b,Aeq,beq,lb,ub,x0)$	Линейные неравенства (4.3), равенства (4.4), ц-область (4.2) и начвектор $x0$
5. $x = \text{linprog}(f,A,b,Aeq,beq,lb,ub,x0,options)$	Линейные неравенства (4.3), равенства (4.4), ц-область (4.2), начвектор $x0$ и изменённые уставки в структуре <code>options</code>
6. $x = \text{linprog}(\text{problem})$	В версии R2008a и ниже – не работает
7. $[x,fval] = \text{linprog}(...)$	Справа – правая часть любого варианта 1...5
8. $[x,fval,exitflag] = \text{linprog}(...)$	“
9. $[x,fval,exitflag,output] = \text{linprog}(...)$	“
10. $[x,fval,exitflag,output,lambda] = \text{linprog}(...)$	“

Наиболее полный вид этой команды – (2.5):

$$[xm,ym,efm,outm,lam] = \text{linprog}(f,A,b,Aeq,beq,lb,ub,x0,optimset(...)). \quad (2.5)$$

Значения аргументов команды должны быть заданы перед командой. Размещение имён аргументов в команде фиксировано. Если какой-либо аргумент (например, $x0$) или их пара (например, A и b) не используются при оптимизации, на место имени каждого неиспользуемого аргумента ставятся квадратные скобки – `[]` (см. рис. 2.2). Если опускается крайний справа аргумент, то на его место можно не ставить скобки, а заменить аргумент замыкающей скобкой – см. рис. 2.2, A (опущены аргументы $ub,x0$ и `options`). Обозначения имён аргументов может варьироваться, но при знакомстве с оптимизатором целесообразно использовать матлабовские имена.

Буква `m` обозначает выходные переменные, относящиеся к минимуму. Имена выходных переменных сокращены по сравнению с данными Справки: `ym` = `fval`, `efm` = `exitflag`, `outm` = `output`, `lam` = `lambda`. Это сокращение не влияет на содержание выдаваемых данных.

Размещение имён выходных переменных так же, как и аргументов – фиксировано. В списке выходных переменных нельзя опускать имена внутри списка, разрешается опускать крайние справа имена, ставя на их место замыкающую скобку списка – см. 1-й столбец табл. 2.1.

Аргумент `optimset(...)` производит изменение величин выбранных уставок функции (при необходимости). Если уставки не меняются, то этот аргумент опускают, т.е. скобку ставят сразу за $x0$.

Для вычисления максимума команда должна быть записана в виде (2.6):

$$[xma, ypr, efma, outma, lama] = \text{linprog}(-f, A, b, Aeq, beq, lb, ub, x0, \text{optimset}(...)), \quad yma = -ypr, \quad (2.6)$$

т.е. вектор f следует умножить на -1 , и на такую же величину умножается ypr , чтобы получить истинное максимальное значение ЦФ $fc(x)$.

Linprog вычисляет глобальный оптимум!

2.2.1. Алгоритмы функции linprog

В функции `linprog` применяются два типа минимизационных алгоритма: крупно-размерностный (large scale – КР-алгоритм) и средне-размерностный (medium-scale – СР-алгоритм) [17]. В КР-алгоритме применяются операции с разреженными матрицами (матрицы, состоящие в основном из нулей), что позволяет резко сократить память для их хранения и время выполнения операции над ними (хранение и операции производятся только для ненулевых элементов). В СР-алгоритме используются полные матрицы, и все операции выполняются для каждого элемента матрицы. Для задач большой размерности (при числе компонентов c -вектора x в несколько десятков, сотен и более) следует применять КР-алгоритм. Для средней и малой размерности c -вектора x – СР-алгоритм. СР-алгоритм можно применять для задач малой и средней размерности, если матрицы задачи, в частности A и Aeq являются разреженными.

КР-алгоритм в `linprog` называется «Линейный для Внутренних Точек Оптимизатор» (ЛВТ-оптимизатор) – LIPSOL (Linear Interior Point Solver). Алгоритм действует в `linprog` по умолчанию. В `linprog` также используются 2 СР-алгоритма: Последовательностного Квадратичного Программирования (ПКвП-алгоритм, Sequential Quadratic Programming – SQP-алгоритм) и Симплекс-алгоритм. Алгоритмы становятся действующими в `linprog`-оптимизаторе по командам:

$$\left. \begin{aligned} \text{options} = \text{optimset}(\text{'LargeScale'}, \text{'off'}) & \text{ – для ПКвП-алгоритма,} \\ \text{options} = \text{optimset}(\text{'LargeScale'}, \text{'off'}, \text{'Simplex'}, \text{'on'}) & \text{ – для Симплекс-алгоритма.} \end{aligned} \right\} \quad (2.7)$$

Каждая из этих команд должна быть записана выше оптим-команды (2.5) или (2.6). Можно также правую часть из (2.7) вставить вместо аргумента `options` в (2.5) или (2.6), не записывая команды (2.7).

2.2.2. Справочные команды и программа функции

В набор входят 4 команды.

`Open('linprog')` – вывод кода `linprog` в окно Редактора. Результат работы команды приведен в Приложении 2: программа `linprog` на 337 строках.

Тип `linprog` – вывод кода `linprog` в командное окно. Результат аналогичен первой команде, но без подкрашивания зелёным цветом справочных строк и без обеспечения редакционных функций, таких, например, как свёртывание тел циклов и условий, т.е. просто текст программы.

`Dos linprog` – вывод основной справки по `linprog` в окно справки (в `help`-окно), в которой детально описываются объекты оптимизатора (включая аргументы и выходные переменные), опции, настройки, алгоритмы и примеры применения.

`Help linprog` – вывод локальной справки из кода программы функции (строки 1...77, см. Приложение 2) в командное окно.

2.2.3. Непараметрические и параметрические линпрогооптимизаторы на основе функции linprog

Линпрогооптимизатором называется программа, содержащая условия оптимизации (2.2)...(2.4), вариант функции `linprog`, соответствующий наборам условий оптимизации и выходных переменных (см. табл. 2.1), и 2 текстовых файла Матлаба, обеспечивающих удобство коррекции условий линпрогооптимизации в случае неудачного решения, вывод результатов в табличной форме, совместимой с типовыми текстредакторами (например, с Уордом) и автоматического сохранения условий оптимизации в текстовом файле Матлаба при решении ряда или множества линпрогозадач.

Линпрогооптимизаторы разделяются на непараметрические и параметрические с одним (однопараметрические), с двумя (двупараметрические) и более параметрами. Объектам и терминам однопараметрической линпрогооптимизации целесообразно присваивать в начале слова букву «р» (от английского *parameter*) для отличия

их от непараметрических объектов. Например, вместо термина «однопараметрическая линпрогооптимизация» следует с предварительным пояснением использовать термин «р-линпрогооптимизация». К именам файлов и переменных «р» следует присоединять без тире. Аналогично для объектов двухпараметрической линпрогооптимизации следует вставлять набор «r1p2». Например, r1p2-линпрогооптимизация.

Линпрогуказанием называется программы, содержащая условия и настройки линпрогооптимизации, функцию linprog или текстфайл с этой функцией и с дополнительными возможностями по сохранению объектов условий оптимизации.

В монографии опубликованы, описаны и приведены процедуры использования с примерами применения следующих линпрогуказаний и линпрогооптимизаторов.

- 1) Линпрогуказание с ручным вводом номеров объектов и выходных данных (разд. 2.4.1).
- 2) Универсальный линпрогооптимизатор с параметрическим вводом типа оптимизации, автоматической уникальной нумерацией имён объектов и выходных данных (разд. 2.4.2), состоящий из Универсального линпрогуказания LinProgUnivUkz и Универсального линпрогооптимизатора LinProgUniversAvtomatM.
- 3) Универсальный линпрогооптимизатор с выдачей единой итог-таблицы запуска оптимизации (разд. 2.4.4), состоящий из Универсального линпрогуказания LinProgUnivUkz_Tab и Универсального итогтабличного линпрогооптимизатора LinProgUniversAvtomatMU_Tab.
- 4) Однопараметрических р-линпрогооптимизаторов с ручной нумерацией объектов задачи pLinProgMinRuch.m и pLinProgMaxRuch.m (разд.3.1), каждый из которых включает матлабовские текстфайлы Формирователя отладочной html-таблицы pTabLinProgRuchOtladHtm и Универсального строителя html-итогтаблиц pLinProgRuchP1OptTabFormer.
- 5) Универсальный р-линпрогооптимизатор pLinProgTabGenOptimer (разд. 3.6), включающий 2 матлабовских текстфайла: р-отладтаблицы pLinProgUnverOtlad для коррекции условий оптимизации в случае неудачного решения для части или для всех значений параметра р и строителя р-итогтаблицы и копировщика в действующую папку файла этого оптимизатора с именем заголовка итогтаблицы pLinProgUnverP1TabStroyker.

Кратка характеристика оптимизаторов.

1. Линпрогуказание-оптимизатор с ручным вводом объединяет условия оптимизации (2.2)...(2.4) в формате Матлаба, функцию linprog и выходные объекты pg и gv, используемые при необходимости для коррекции условий линпрогооптимизации. Это наиболее простой оптимизатор: для несложных условий программа занимает всего 10 строк – см. рис. 2.3. Имеет 2 формы: для минимума и для максимума. Достоинство – простота. Недостатки: необходимость ручной нумерации имён условий, выходных переменных и объектов при выполнении нескольких задач оптимизации с сохранением условий всех задач; черновой формат результата, не обеспечивающий прямую вставку результата в текстовый документ.

2. Универсальный линпрогооптимизатор обладает параметрическим вводом с программным выбором типа оптимизации(мин/макс), автоматическими раздельной (для мин/макс) нумерацией и присвоением уникальных идентификаторов всем объектам и выходным переменным в Вычпространстве, и выдаёт в Комокно значения текущей линпрогооптимизации: нумерованных и отидентифицированных выходных переменных x,y,ef и выходных объектов pg и gv. Он состоит из Универсального линпрогуказания LinProgUnivUkz (рис. 2.4), в состав которого включен текстфайл Универсального линпрогооптимизатора LinProgUniversAvtomatM (рис. 2.5). Все настройки (линейная нумерация объектов, выбор типа оптимизации, изменение настроек функции linprog, ввод условий оптимизации) выполняются в линпрогуказании. В оптимизаторе LinProgUniversAvtomatM нет настроек: присвоение уникальных идентификаторов выполняется автоматически. Объём составного линпрогооптимизатора – 70 строк.

Достоинства: возможность решения множества задач с одним и тем же оптимизатором, не заботясь о нумерации объектов задачи и результата – это всё будет происходить автоматически, причём, ц-функция каждой задачи будет получать дополнительно к идентификатору дату и время выполнения линпрогооптимизации.

Недостатки: черновой формат результата, не обеспечивающий прямую вставку результата в текстовый документ и отсутствие возможности быстрого сравнения результатов ряда задач из-за дробного объектового состава результата и индивидуального способа сохранения объектов в Вычпространства.

3. Универсальный итогтабличный линпрогооптимизатор состоит из Универсального линпрогуказания LinProgUnivUkz_Tab (рис.2.12) и включённого в него текстфайла Универсального итогтабличного линпрогооп-

тимизатора LinProgUniversAvtomatMU_Tab (рис. 2.13, А). Этот оптимизатор создан на основе оптимизатора из п. 2: в линпрогуказание LinProgUnivUkz_Tab добавлена переключатель-команда управления выдачей итогтаблицы (выдавать/не выдавать), а в оптимизатор – строки формирования итогтаблицы в html-формате (см. рис. 2.15,А). Имя итогтаблицы содержит такой же идентификатор, что и имя ц-функции. В заголовке таблицы перечислены имена всех условий, задействованных в данном решении. Это позволяет при необходимости оперативно вызывать все условия в Комокно и также оперативно удалять условия из Вычпространства для его разгрузки при большом числе задач оптимизации командой «cleag name1 name2 name3 ...», в которой имена «name1 name2 name3 ...» - имена удаляемых условий, скопированные из заголовка итогтаблицы.

Достоинства: возможность решения множества задач с одним и тем же оптимизатором, не заботясь о нумерации объектов задачи и результата; выдача результата в итогтаблице html-формата, которую можно вставлять в текстовый документ; оперативное сравнение результатов ряда задач по их итогтаблицам; возможность оперативной манипуляции с условиями задачи, сохраненными в Вычпространстве.

Недостатки: по упомянутым выше критериям – отсутствуют.

4. Однопараметрические р-линпрогоптимизаторы с ручной нумерацией объектов оптимизатора pLinProgMinRuch.m (рис. 3.1) и pLinProgMaxRuch.m (рис.3.2). В состав каждого оптимизатора входят: текст-файл pTabLinProgRuchOtladHtm Построителя отладочной html-таблицы с объектами prnm, prvm (рис. 3.6) и текстфайл pLinProgRuchP1OptTabFormer Универсального построителя html-итогтаблиц ручных р-линпрогоптимизаторов (рис.3.10). В р-линпрогоптимизаторах объекты задачи (условия и выходные величины) не сохраняются в Вычпространстве на постоянной основе. Их сохранение носит оперативный характер: следующий запуск очередной задачи не меняя имён объектов изменяет их значения. Это сделано потому, что значения объектов представляют собой наборы чисел, не отражающих позиций вставки параметра р, что не позволяет воссоздать исходную структуру условий оптимизации задачи. Взамен построитель pLinProgRuchP1OptTabFormer наряду с итогтаблицей создаёт копию р-линпрогоптимизатора с тем же именем, что у итогтаблицы (см. рис. 3.8 и 3.9).

Достоинства: сохранение результата в html-таблице; сохранение условий задачи в m-файле, однозначно связанным с файлом html-таблицы; возможность набора множества задач одного типа р-линпрогоптимизации в одном и том же оптимизаторе, полагаясь на автоматическое сохранение условий каждой задачи.

Недостатки: наличие двух р-оптимизаторов (для минимума/ максимума), невозможность в одном оптимизаторе выполнить задачу другого типа р-линпрогоптимизации.

5. Универсальный р-линпрогоптимизатор pLinProgTabGenOptimer (рис. 3.12) включает 2 матлабовских текстфайла: формирователя р-отладтаблицы pLinProgUnverOtlad (рис. 3.17) и построителя р-итогтаблиц pLinProgUnverP1TabStroyker (рис. 3.15), который также копирует файл линпрогоптимизатора в действующую папку с присвоением ему имени файла р-итогтаблицы (см. рис. 3.13 и 3.14).

Достоинства: сохранение результата в html-таблице; сохранение условий задачи в m-файле, однозначно связанным с файлом html-таблицы; возможность набора множества задач разного типа р-линпрогоптимизации в одном и том же оптимизаторе; сокращение набора оптимизаторов с 2 до одного.

Недостатки: по упомянутым выше критериям – отсутствуют

Рекомендации по применению.

Для непараметрической линпрогоптимизации следует использовать Универсальный итогтабличный линпрогоптимизатор, состоящий из Универсального линпрогуказания LinProgUnivUkz_Tab (рис.2.12) и включённого в него текстфайла Универсального итогтабличного линпрогоптимизатора LinProgUniversAvtomatMU_Tab (рис. 2.13, А).

Для однопараметрической линпрогоптимизации следует использовать Универсальный р-линпрогоптимизатор pLinProgTabGenOptimer (рис. 3.12) с включёнными в него двух матлабовских текстфайлов: формирователя р-отладтаблицы pLinProgUnverOtlad (рис. 3.17) и построителя р-итогтаблиц pLinProgUnverP1TabStroyker (рис. 3.15).

Упомянутые линпрогоптимизаторы 1...5 можно скачать с электронного варианта монографии (имя – LinProg_Matlab_2016Rykin), помещённого на сайте кафедры Управление проектами Института Компьютерных наук и технологий Санкт-Петербургского политехнического университета Петра Великого по адресу: http://www.iit.spb.ru/2005/ins_inn_material/baza_general_document.php в папке PosobiyaPoKomputerKonstr-

2.3. Вычислительные возможности linprog. Поиск 5000-компонентного минимизационного вектора

Определение минимума линейной ц-функции $fc(x)$ вида (2.1) (со случайным равномерным выбором коэффициентов в диапазоне 0...5) для 5000-компонентного вектора x при 50 ограничениях-неравенствах (2.3), сформированных для первых 50 компонент вектора x для ц-области (2.2). Матрица A задачи является разреженной: число элементов в ней 1 млн, а число ненулевых элементов $50 \times 50 = 2500$, т.е. нулей больше в 2000 раз. Решение задачи представлено на рис. 2.1: время решения 15,62 с. Назначение отдельных элементов оптим-указания – в его построчных комментариях. Применяется по-умолчанию КР-алгоритм, оптимизированный для решения задач с разреженными матрицами.

```
tic
n =5000;n1=50;n2=50;Ar=3.*rand(n1,n2);A1o=zeros(n1,n-n2);Apr=cat(2,Ar,A1o);
A2o=zeros(n-n1,n);A=cat(1,Apr,A2o);
b=[15.*rand(n1,1);zeros(n-n1,1)]; % первые n1 компонент числа, равномерно
    % распределенные в диапазоне 0...15; а остальные 4950 - нули
f=(-1).^fix(10.*rand(1,n)).*6.*rand(1,n); % равномерное распределение n компонент
    % в диапазоне -3...0 с умножением некоторых на -1, когда
    % fix(10.*rand(1,n))=нечётному числу
lbv=-3.*rand(1,n); % равномерное распределение n компонент в диапазоне -3...0
ubv=4.*rand(1,n); % равномерное распределение n компонент в диапазоне 0...4
[xv,yv,exfv,outv]=linprog(f,A,b,[],[],lbv,ubv)
toc
Optimization terminated.
xv =
    -0.0293
   -2.0798
    0.2145
   -0.5843
    2.4094
    3.3837
    2.1693
    3.8681
   -0.1582
   -2.0770
   -0.2050
    ... % Пропущено 4980 строк
    2.7546
   -0.5242
    3.1224
    1.7615
    3.0184

yv =
   -2.6342e+004
exfv =
    1
outv =
    iterations: 11
    algorithm: 'large-scale: interior point'
    cgiterations: 0
    message: 'Optimization terminated.'
```

Рис. 2.1. Вычисление минимума линейной ц-функции (2.1) для ц-вектора с 5000 компонент при наличии ограничений-неравенств (2.3) с участием первых 50 компонент ц-вектора x . Время вычисления 15,62 с для процессора Пентиум Т2080, 1,73 ГГц, ОЗУ 1,5 Гб

2.4. Линпрогуказания и линпрогоптимизаторы с использованием функции linprog

Рассмотрим линпрогзадачи: определить 5-компонентный вектор x , оптимизирующий линейную ц-функцию $fc(x)$ (2.8), и вычислить её оптим-значения посредством штатных алгоритмов поиска (см. разд. 2.2.1) для следующих условий.

$$Fc(x) = -5x(1) + 4x(2) - 6x(3) - 3x(4) + 9x(5) = f' \cdot x, \quad (2.8)$$

(где f и x – векторы, равные в символике Матлаба: $f = [-5 \ 4 \ -6 \ -3 \ 9]$, $x = [x(1) \ x(2) \ x(3) \ x(4) \ x(5)]$) при ограничении типа неравенств (2.9) и равенств (2.10), записанных с компонентами ц-вектора в виде независимых переменных :

$$\left. \begin{aligned} x_1 - x_2 + x_3 + 2x_4 + 3x_5 &\leq 20 \\ 3 \cdot x_1 + 2 \cdot x_2 + 4 \cdot x_3 + x_4 - 3x_5 &\leq 42 \\ 3 \cdot x_1 + 2 \cdot x_2 + 4 \cdot x_4 + 3x_5 &\leq 30 \\ x_1 + 3 \cdot x_2 + 5 \cdot x_3 + 4 \cdot x_4 - 2x_5 &\leq 15 \\ 3 \cdot x_1 + 4 \cdot x_2 + 7 \cdot x_3 + 2 \cdot x_4 - 1.5x_5 &\leq 23 \end{aligned} \right\} \quad (2.9)$$

$$\left. \begin{aligned} 1.5 \cdot x_2 + 2 \cdot x_5 &= 7, \\ 3 \cdot x_1 + x_2 - 0.1 \cdot x_4 &= 2; \end{aligned} \right\} \quad (2.10)$$

для следующих областей изменения ц-вектора x :

$$\left. \begin{aligned} \text{а) } 0 \leq x(1), 0 \leq x(2), 0 \leq x(3), 0 \leq x(4), 0 \leq x(5), \\ \text{б) } 0 \leq x(1) \leq 6.5; 0 \leq x(2) \leq 4.5; 0 \leq x(3) \leq 5; 0 \leq x(4) \leq 8.2; 0 \leq x(5) \leq 3.1. \end{aligned} \right\} \quad (2.11)$$

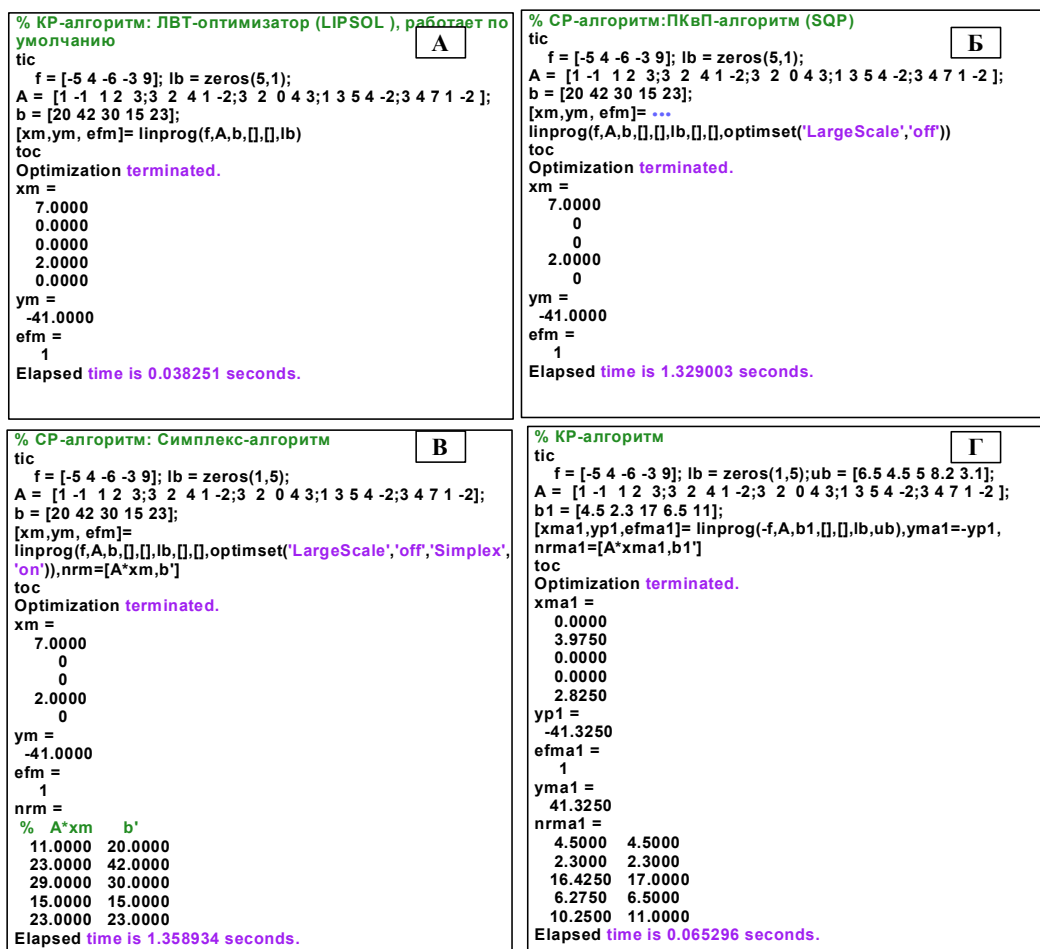


Рис. 2.2. Решения линпрогзадач посредством простых линпрогуказаний : минимизации ц-функции (2.8) с ограничениями-неравенствами (2.9) в ц-области (2.11 а) для 3 типов алгоритмов оптимизатора linprog (А, Б и В); максимизации с неравенствами (2.9) в ц-области (2.11 б). zeros(1,5)=[0 0 0 0 0], nrm, nrma1 – объекты для проверки степени выполнения неравенств (2.9). Время поиска минимума с учётом разреженности матриц (рис. 2.2 А) в 30 раз меньше, времени поиска без её учёта (рис. 2.2 Б и В), т.е. с использованием полных матриц.

Варианты решения линпрогзадач с одновременным вычислением времени решения каждой операторами tic-toc представлены: минимизации $[xm, ym, efm] = \text{linprog}(f, A, b, [], [], lb)$ для 3 алгоритмов – рис. 2.2 А, Б, В;

максимизации $[xma, yp, efma] = \text{linprog}(-f, A, b, [], [], lb, ub)$, $yma = -yp$ для КР-алгоритма – рис. 2.2 Г; ель изации посредством оптимизатора (2.6) с выводом всего набора выходных переменных ПМФ linprog – рис. 2.2.

Аргументы и выходные переменные в линпрогуказаниях на рис. 2.2 и 2.3. f – вектор коэффициентов ц-функции (2.8), lb – вектор нижних границ ц-области, состоит из левых частей неравенств (2.11 а), ub – вектор верхних границ ц-области, состоит из правых частей неравенств (2.11 б). A – матрица коэффициентов неравенств (2.9), b – вектор правых частей неравенств (2.9). xm , ym – имена для минимизационных значений ц-вектора и ц-функции и, efm – флаг окончания процесса минимизации. Aeq – матрица коэффициентов равенств (2.10), beq – вектор правых частей равенств (2.10). xma , yma – имена для максимизационных значений ц-вектора и ц-функции и, $efma$ – флаг окончания процесса максимизации, ngm , $ngma1$ – объекты для проверки степени выполнения неравенств (2.9).

Как видно из рис. 2.2 время поиска минимума с учётом разреженности матриц (рис. 2.2,А) в 30 раз меньше, времени поиска без её учёта (рис. 2.2,Б и В), т.е. с использованием полных матриц.

2.4.1. Шаблон линпрогуказания с ручным вводом номеров объектов и выходных данных

При решении подряд нескольких задач линейного программирования для предотвращения стирания очередным результатом предыдущего возникает необходимость в нумерации исходных объектов задачи (f , A , b и т.д.) и выходных переменных (xop , yop и $efop$). Это делается вручную (программный вариант нумерации – см. разд. 2.4.2) при формировании линпрогуказания – см. рис. 2.3 (нумерованные строки), на котором представлены шаблоны вычисления линейных максимума и минимума в наиболее полном функциональном оснащении, за исключением аргумента `options`, вставляемого в крайнюю правую позицию линпрогоптимизатора. Заполняя эти шаблоны следует руководствоваться пп. 3)...7)

А	Б
<pre> 1 tic 2 f2 = [-5.3 2.2 -6 -3 9]; 3 A2 = [1 -1 1 2 3; 3 2 4 1 -2; 3 2 0 4 3; 1 3 5 4 -2; 3 4 7 1 -2]; 4 b2 = [20 42 30 15 23]; 5 Aeq2 = [1.7 1.5 0 0 2; 3 1 4.2 0.8 0]; 6 beq2 = [5.3 2.9]; 7 lb2=zeros(1,5); ub2=ones(1,5)*10; 8 [xm2,ym2,efm2,outm2,lam2]= linprog(f2,A2,b2,Aeq2,beq2,lb2,ub2) 9 nrm2=[A2*xm2,b2']; rvm2=[Aeq2*xm2,beq2'] 10 Tsch=toc Optimization terminated. xm2 = 0.0000 2.9000 0.0000 0.0000 0.4750 ym2 = 10.6550 efm2 = 1 outm2 = iterations: 7 algorithm: 'large-scale: interior point' cgiterations: 0 message: 'Optimization terminated.' constrviolation: 1.5543e-014 lam2 = ineqlin: [5x1 double] eqlin: [2x1 double] upper: [5x1 double] lower: [5x1 double] nrm2 = -1.4750 20.0000 4.8500 42.0000 7.2250 30.0000 7.7500 15.0000 10.6500 23.0000 rvm2 = 5.3000 5.3000 2.9000 2.9000 Tsch = 0.6594 </pre>	<pre> 1 tic 2 f = [-5 4 -6 -3 9]; 3 A = [1 -1 1 2 3; 3 2 4 1 -2; 3 2 0 4 3; 1 3 5 4 -2; 3 4 7 1 -2]; 4 b1 = [4.5 2.3 17 6.5 11]; 5 Aeq = [0 1.5 0 0 2; 3 1 0 0 0]; 6 beq = [7 2]; 7 lb = zeros(1,5); ub = [6.5 4.5 5 8.2 3.1]; 8 [xma2,yp2,efma2,outma2,lama2]= linprog(-f,A,b1,Aeq,beq,lb,ub) 9 yma2=-yp2,nрма2=[A*xma2,b1'],rvma2=[Aeq*xma2,beq'] 10 toc Optimization terminated. xma2 = 0.0465 1.8605 0.0000 0.0000; 2.1047; yp2 = -26.1512 efma2 = 1 outma2 = iterations: 7 algorithm: 'large-scale: interior point'; cgiterations: 0 message: 'Optimization terminated.' lama2 = ineqlin: [5x1 double] eqlin: [2x1 double] upper: [5x1 double] lower: [5x1 double] yma2 = 26.1512 nрма2 = % A*xma2 b' 4.5000 4.5000 -0.3488 2.3000 10.1744 17.0000 1.4186 6.5000 3.3721 11.0000 rvma2 = % Aeq*xma2 beq' 7.0000 7.0000 2.0000 2.0000 Elapsed time is 2.041143 seconds. </pre>

Рис. 2.3. Вычисление минимума (А) и максимума (Б) ЦФ (2.8) при ограничениях (2.9) и (2.10) в ц-области (2.11 б) с выводом всего набора выходных переменных функции linprog при поиске максимума. Командуказание в строках 1...10 является эталонным при поиске линпрог-минимума/максимума: в конкретном поиске неиспользуемые аргументы могут быть удалены или закомментированы в строках указания, а их имена в списке линпрог-оптимизатора замещены квадратными скобками или закрывающей скобкой списка согласно правилу из разд. 2.2. Пронумерованные затенённые строки можно копировать с электронной версии Пособия в качестве шаблонов для решения задач линпрогоптимизации.

Разд. 2.5. Выходные переменные `out` и `lam` при типовом поиске оптимума можно не включать в список. Выход-

ные объекты – `ngm`, `ngma` (неравенства) и `gvm`, `gvma` (равенства) – при успешном поиске ($ef=1$) также можно не включать в указание. В случае неудачного поиска эти объекты нужно включать для определения источников ошибок и путей коррекции линпрогзадачи для её успешного решения – см. разд. 2.7.1.4.

2.4.2. Универсальный линпрогооптимизатор с переключательным вводом типа оптимизации, номеров объектов и выходных данных

Ручная нумерация объектов линпрогооптимизации подходит при решении 2...4 задач. При большем их числе целесообразно перейти к программному способу нумерации. На рис. 2.4...2.6 представлен вариант программного способа (шаблона) и результатов его применения, состоящий из Универсального линпрогуказания `LinProgUnivUkz` и Универсального линпрогооптимизатора `LinProgUnivesAvtomatMU`.

```
%
% Универсальное линпрогуказание LinProgUnivUkz
% Настройки универсального линпрогооптимизатора LinProgUnivesAvtomatMU
tic
NumLPmri=0; % Нумератор объектов линпрогмина очистить при необх. Новой нумерации. После 1 пуска – закомментировать.
NumLPma=0; % Нумератор объектов линпрогмакса очистить при необх. Новой нумерации. После 1 пуска – закомментировать.
% Lmi_Lma=1; % Поиск линпрогмина. Закоммент. При поиске линпрогмакса.
Lmi_Lma=-1; % Поиск линпрогмакса. Закоммент. При поиске линпрогмина.
% options=optimset('param1',число,'param2','строк-значение',...); % Изменение настроек линпрогфункции Матлаба
linprog(...):раскомментировать строку и заполнить шаблон optimset(...).

% Аргументы линпрогфункции Матлаба linprog(...)
fo = [-5.6 4.4 6 -3 9];
Ao = [1 -1 1 2 3; 3 2 4 1 -2; 3 2 0 4 3; 1 3 5 4 -2; 3 4 7 1 -2]; bo = [6 2.3 17 6.5 11];
% Ao=[]; bo=[];
Aeqo = [0 1.5 0 0 2; 3 1 2.5 0 0.7; 4.1 -2.3 11 3.4 4.7]; beqo = [7.82 3.62 7.62];
% Aeqo=[]; beqo=[];
Lbo = zeros(1,5); Ubo = [6.5 4.5 5 8.2 3.1];

% Запуск универсального линпрогооптимизатора.
LinProgUnivesAvtomatMU % Текст-файл универсального линпрогооптимизатора.
Точ
```

Рис. 2.4. Универсальное линпрогуказание с программным выбором типа оптимизации(мин/макс), с возможностью отдельной (для мин/макс) нумерации и уникальной идентификацией всех объектов и выходных переменных и с выдачей в Комокно значений текущей линпрогооптимизации: нумерованных и идентифицированных выходных переменных `x`, `y`, `ef` и выходных объектов `ng` и `gv`. Исполнительная часть линпрогуказания заключена в вызываемом текст-файле `LinProgUnivesAvtomatMU` (рис. 2.5).

В шаблоне входная часть и исполнительная принадлежат разным программам, что исключает возможность непреднамеренного внесения ошибки в исполнительный код при вводе настроек или аргументов.

В команду `линказани` объединены управляющие настройки линпрогооптимизатора, аргументы и команда запуска. Назначение настроек приведено в комментариях. Поскольку обе команды обнуления нумераторов (`NumLPmri`/ `NumLPma`) ель изации вать, оптитмизатор работает в режиме возрастания номеров объектов на 1 с каждым запуском. Если одну из команд раскомментировать и не закомментировать после запуска, то все объекты оптимизации, соответствующей незакомментированной команде, будут иметь номер 1, не зависящий от количества запусков.

Линпрогуказание на рис. 2.4 настроено на поиск максимума, т.к. раскомментирована команда `Lmi_Lma=-1`. Если обе команды будут раскомментированы, то оптимизатор будет тоже искать максимум, поскольку его команда следует непосредственно за командой минимума, отменяя её.

Команда `options` предназначена для изменения настроек оптимфункции Матлаба `linprog(...)`, выбираемых из 6 категорий согласно табл. 2.1 из разд. 2.6. Общий вид команды изменения уставок и пример её применения для изменения числа итераций в алгоритме (с 85 на 100) и уставки дисплея (с `final` на значение `iter`):

```
(2.12) options = optimset('param1',число,'param2', 'строк-значение'...),
options = optimset('MaxIter',100, 'Display', 'iter').
```

Численное значение параметра ставится без апострофов, а именованное (строк-значение) – в апострофах. Значения аргументов `fo`...`Ubo` проведены для учебного примера. Их можно менять как в отношении

размерности задачи (увеличивать/ уменьшать размерность), так и по конкретному виду линейных ограничений и границ ц-области (Lbo, Ubo). Нельзя менять имена объектов, т.к. они задействованы в программе оптимизатора. Результаты работы линпрогооптимизатора показаны на рис. 2.6, А и Б. Пояснения по результатам – ниже в этом разделе.

Универсальный линпрогооптимизатор LinProgUniversAvtomatMU показан на рис. 2.5. Назначение команд указано в построчных комментариях. Подробное содержание использованных команд – в Приложении 3 (копии из Справки, английский текст). Пояснения по ряду команд линпрогооптимизатора.

```

1 % Универсальный линпрогооптимизатор LinProgUniversAvtomatMU
2 % С переключаемым выбором типа оптимизации (мин/макс), с возможностью раздельной (для мин/макс)
3 % нумерации всех объектов и выходных переменных и с выдачей в Комокно значений текущей
4 % линпрогооптимизации: нумерованных выходных переменных x,y,ef и нумер. вых.объектов ng и gv.
5 % Запускается только в составе универсального линпрогуканзания или самостоятельно кнопкой "Save and Run" или
6 % клавишей f9 (при выделении) при наличии в Вычпространстве всех исходных объектов:
7 % NumLPrmi, NumLPrma, Lmi_Lma, fo, Ao, bo, Aeqo, beqo, LBo, UBo.
8 tic; % Запуск таймера программы.
9 a=Lmi_Lma;
10 NomTStr=num2str(rem(now,1)); % Формирование уникальной части номера в виде строк-переменной из временного
11 NomTStr(1)=[]; NomTStr(1)='_'; %; момента выполнения команды с удалением 1-го нуля и с замен. след. точки на нижн. подчёрк.
12 NumerLPrmi=1; % Нумеровать объекты линпрогмина (Об ЛПрМи) для сеансов программы; 0 - не нумеровать.
13 NumerLPrma=1; % Нумеровать объекты линпрогмакса (Об ЛПрМа) для сеансов программы; 0 - не нумеровать.
14 if (NumerLPrmi==1) && (NumerLPrmi==0) && (Lmi_Lma==1) % При нумерац. Об ЛПрМи, очищен. нумерат. и линпрогминимиз.
15 NumLPrmi=1; % присвоить нумератору Об ЛПрМи номер 1.
16 elseif (NumerLPrmi==1) && (NumerLPrmi>=1) && (Lmi_Lma==1) % При нумерац. Об ЛПрМи, заполн. нумерат. и линпрогминимиз.
17 NumLPrmi=NumLPrmi+1; % увеличить номер нумератора Об ЛПрМи на 1.
18 end
19 NumLPrmStr=[num2str(NumerLPrmi) NomTStr]; % Присоединение к строк-номеру минобъекта уникальной части.
20 if (NumerLPrma==1) && (NumerLPrma==0) && (Lmi_Lma==1) % При нумерац. Об ЛПрМа, очищен. нумерат. и линпрогмаксимиз.
21 NumLPrma=1; % присвоить нумератору Об ЛПрМа номер 1.
22 elseif (NumerLPrma==1) && (NumerLPrma>=1) && (Lmi_Lma==1) % При нумерац. Об ЛПрМа, заполнен. нумерат. и линпрогмаксимиз.
23 NumLPrma=NumLPrma+1; % увеличить номер нумератора Об ЛПрМа на 1.
24 end
25 nro=[]; rvo=[]; % Очистка выходных объектов от результата предыдущего вычисления.
26 NumLPrmaStr=[num2str(NumerLPrma) NomTStr]; % Присоединение к строк-номеру максобъекта уникальной части.
27 [xo,yo,efo]=linprog(a*fo,Ao,bo,Aeqo,beqo,LBo,UBo,options); % Вычисление линпрогооптимума.
28 % Вычисление выходных объектов: векторов огранич.-неравенств и огранич.-равенств.
29 if isempty(Ao) % Проверка пустого значения матрицы Ao.
30 nro=[];
31 else
32 nro=[Ao*xo, bo'];
33 end
34 if isempty(Aeqo) % Проверка пустого значения матрицы Aeqo.
35 rvo=[];
36 else
37 rvo=[Aeqo*xo, beqo'];
38 end
39 dateK=date; dateK(3)='_';dateK(7)='_'; % Замена дефисов в дате на нижн. подчёрк. для прав. работы assignin для Ny.
40 if Lmi_Lma==1
41 Nf=['fmi_' NumLPrmStr];NA=['Ami_' NumLPrmStr];Nb=['bmi_' NumLPrmStr];NAeq=['Aeqmi_' NumLPrmStr]; % Присвоение
42 Nbeq=['beqmi_' NumLPrmStr];NLB=['ANLBmi_' NumLPrmStr];NUB=['UBmi_' NumLPrmStr]; % номеров именам объектам текущей об-
43 Nx=['xmi_' NumLPrmStr];Ny=['ymi_' NumLPrmStr '_' dateK];Nef=['efmi_' NumLPrmStr]; % линпрогминимизации, если она задана.
44 Nnr=['nrmi_' NumLPrmStr];Nrv=['rvmi_' NumLPrmStr];
45 assignin('base', Nf, fo);assignin('base', NA, Ao);assignin('base', Nb, Ao);assignin('base', NA, bo); % Присвоение объектам текущей
46 assignin('base', NAeq, Aeqo);assignin('base', Nbeq, beqo);assignin('base', NLB, LBo);assignin('base', NUB, UBo); % линпрогминимизации,
47 assignin('base', Nx, xo);assignin('base', Ny, yo);assignin('base', Nef, efo); % если она задана, нумерованных имён и запись их
48 assignin('base', Nnr, nro);assignin('base', Nrv, rvo); % с именами в Вычпространство.
49 evalin('base', Nx), evalin('base', Ny), evalin('base', Nef), evalin('base', Nnr), evalin('base', Nrv) % Выдача в Комокно текущих пронумер.
50 % значений линпрогминимиз. x,y,ef,ng и gv, если она задана.
51 end
52 if Lmi_Lma==1
53 Nf=['fma_' NumLPrmaStr];NA=['Ama_' NumLPrmaStr];Nb=['bma_' NumLPrmaStr];NAeq=['Aeqma_' NumLPrmaStr]; % Присвоение
54 Nbeq=['beqma_' NumLPrmaStr];NLB=['ANLBma_' NumLPrmaStr];NUB=['UBma_' NumLPrmaStr]; % номеров именам объектам текущей
55 Nx=['xma_' NumLPrmaStr];Ny=['yma_' NumLPrmaStr '_' dateK];Nef=['efma_' NumLPrmaStr]; % линпрогмаксимизации, если она задана.
56 Nnr=['nrma_' NumLPrmaStr];Nrv=['rvma_' NumLPrmaStr];
57 assignin('base', Nf, fo);assignin('base', NA, Ao);assignin('base', Nb, Ao);assignin('base', NA, bo); % Присвоение объектам текущей
58 assignin('base', NAeq, Aeqo);assignin('base', Nbeq, beqo);assignin('base', NLB, LBo);assignin('base', NUB, UBo); % линпрогмаксимизации,
59 assignin('base', Nx, xo);assignin('base', Ny, -yo);assignin('base', Nef, efo); % если она задана, нумерованных имён и запись их
60 assignin('base', Nnr, nro);assignin('base', Nrv, rvo); % с именами в Вычпространство.

```

Рис. 2.5. Универсальный линпрогооптимизатор LinProgUniversAvtomatMU с переключаемым выбором типа оптимизации (мин/макс), с возможностью раздельной (для мин/макс) нумерации всех объектов и выходных пере-

менных и с выдачей в Комокно значений текущей линпрогооптимизации: нумерованных выходных переменных x, y, ef и нумерованных выходных объектов ng и gv .

Команда «gem(pow,1)» в 10-й строке оптимизатора является производной от команды «pow», в которой закодированы в виде одного числа дата и время её исполнения (см. разд. 6 Приложения 3), производной в виде закодированного числа времени исполнения. Типичный результат исполнения команды pow и 2 её производных представлен в выражениях (2.13):

$$\left. \begin{aligned} nn1 &= now \\ nn2 &= floor(nn1) \\ nn3 &= rem(nn1, 1) \\ nn1 &= 7.3642e+005; nn2 = 736420; nn3 = 0.7954. \end{aligned} \right\} \quad (2.13)$$

Результат команды $gem(nn1, 1)$ меньше 1 и имеет суточный период, равный 86400 сек. Повторение команд через сутки в одну и ту же секунду практически невозможно, что и обеспечивает уникальность числа-результата рассматриваемой команды. Для использования числа-результата из него удаляются ноль и точка командами 11-й строки оптимизатора.

Форомирование 2 рядов простых порядковых номеров (один – для минимума, другой – для максимума) производится двумя группами if-команд – строки 14...18 и 20...24 на рис. 2.5. К этим номерам в строках 19 и 25 присоединяются уникальные идентификаторы. Каждый из этих 2 составных номеров присваивается именам всех входных и выходных объектов при выполнении «своей» оптимизации – строки 41...44 – для минимума, 53...56 – для максимума. Кроме того, в номера ц-функции Nu добавляется ещё дата выполнения оптимизации – строки 43 и 55. Пример представления минимизационного вектора, ц-функции и матрицы коэффициентов неравенств в Вычпространстве с описанными идентификаторами – выражения (2.14):

$$\left. \begin{array}{l} xmi_4_92241' = \\ 0.0000 \\ 1.8589 \\ 0.0000 \\ 0.0210 \\ 2.5158 \end{array} \right\} \left. \begin{array}{l} ymi_4_92241_30_Mar_2016 = \\ 30.7585 \\ \text{Ami_4_92241} \quad 1 \times 5 \quad [6, 2.3000, 17, 6.5000, 11] \quad 2.3000 \quad 17 \quad \text{double} \end{array} \right\} \quad (2.14)$$

В этих выражениях величина 4_92241 – составной номер объекта: 4 – порядковый номер, 92241 – уникальный идентификатор. Упомянутые имена с идентификаторами посредством команд assignin помещаются как переменные в Вычпространство с одновременным приписыванием к ним значений соответствующих объектов выполняемой линпрогооптимизации. Посредством команд evalin в строках 49 и 61 выходные переменные и объекты ng и gv последнего запуска оптимизации из Вычпространства открываются в Комокне – см. рис. 2.6, А.

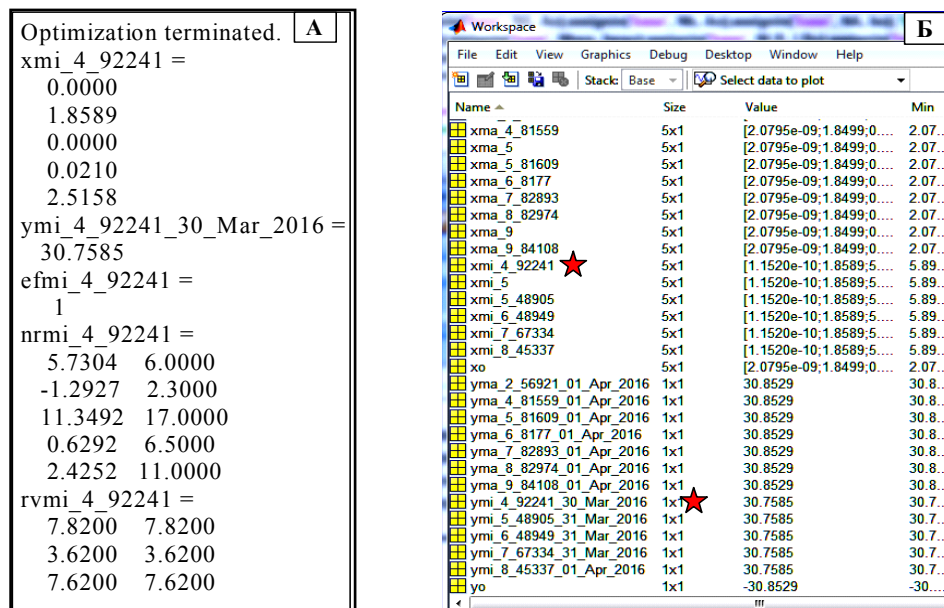


Рис. 2.6. Результаты линпрогминимизации, выведенные в Комокно (А) и фрагмент Вычпространства (Б)

с π -функцией `umi_4_92241_30_Mar_2016` и оптимизационным вектором `xmi_4_92241` (отмечены звёздочками) из этих результатов.

Выходные объекты `pg` и `gv` являются 2-столбцовыми матрицами – см. рис. 2.6, А. Их вторые столбцы – значения векторов `bo` и `beqo` выполненной оптимизации, которые переданы в Вычпространство с именами соответственно (для рис. 2.6, А): `bmi_4_92241` и `beqmi_4_92241`. Первые столбцы этих объектов – значения ограничений неравенств и равенств, матрицы коэффициентов которых `Ao` и `Aeqo` переданы в Вычпространство с именами соответственно: `Ami_4_92241` и `Aeqmi_4_92241`.

Для восстановления времени выполнения линпрогооптимизации по уникальному идентификатору следует набрать 2 команды типа (2.15):

```
DatChis = datenum(2016, 03, 30)
% DatChis = 736419
DatVrem = datestr(736419.92241)
% DatVrem = 30-Mar-2016 22:08:16.
```

(2.15)

Закомментированы результаты выполнения команд 1-й и 3-й строк. Причём, в скобках 1-й команды стоит дата из имени π -функции из рис. 2.6, А (`umi_4_92241_30_Mar_2016`). Результат выполнения 1-й команды – целое число, соответствующее введённой дате. Аргумент 2-й команды – дробное число, целая часть которого равна результату 1-й команды, а цифры дроби равны цифрам уникального идентификатора. В примере идентификатору 92241 соответствует время 22ч: 8м: 16с – это время выполнения команды `NomTStr=num2str(rem(now,1))` в 10-й строке оптимизатора на рис. 2.5.

Детальное описание команд `datenum` и `datestr` представлено в Приложении 3.

В строках 29...38 производится формирование выходных объектов текущей оптимизации `pg` и `gv`. Такая процедура обусловлена возможностью использования значений: `Ao=[]`, `Aeqo=[]`, `bo=[]`, `beqo=[]`. Одновременно всех или порознь. Для этих случаев произведения `Aeqo*хо` и `Ao*хо` не работают (Матлаб выдаёт ошибку). Ошибочны также и условия `Ao==[]`, `Aeqo==[]`, что заставило применить команду `isempty(Ao)` (Приложение 3).

2.4.3. Применение универсального линпрогооптимизатора для решения задач с размерностью 7000

2.4.3.1. Линпрогминимум для `Ao 7000x7000` с числом ненулевых значений `100x50`

Набор задачи – на рис. 2.7.

```
%
% Универсальное линпрогуказание LinProgUnivUkz
% Настройки универсального линпрогооптимизатора LinProgUnivesAvtomatMU
tic
NumLPmi=0; % Нумератр объектов линпрогмина очистить при необх. Новой нумерации. После 1 пуска – заком-
ментировать.
NumLPma=0; % Нумератор объектов линпрогмакса очистить при необх. Новой нумерации. После 1 пуска – за-
комментировать.
Lmi_Lma=1; % Поиск линпрогмина. Закоммент. при поиске линпрогмакса.
% Lmi_Lma=1; % Поиск линпрогмакса. Закоммент. При поиске линпрогмина.
% options=optimset('param1',число,'param2','строка-значение',...); % Изменение настроек линпрогфункции Матлаба
linprog(...): раскомментировать строку и заполнить шаблон optimset(...).
% Аргументы линпрогогфункции Матлаба linprog(...)
% fo = [-5.6 4.4 6 -3 9];
% Ao = [1 -1 1 2 3; 3 2 4 1 -2; 3 2 0 4 3; 1 3 5 4 -2; 3 4 7 1 -2]; bo = [6 2.3 17 6.5 11];
% Ao=[]; bo=[];
% Aeqo = [0 1.5 0 0 2; 3 1 2.5 0 0.7; 4.1 -2.3 11 3.4 4.7]; beqo = [7.82 3.62 7.62];
% Aeqo=[]; beqo=[];
% Lbo = zeros(1,5); Ubo = [6.5 4.5 5 8.2 3.1];
% Возможности Универсального линпрогооптимизатора. Вычисление для n =7000
n =7000; n1=100; n2=50; Ar=3.*rand(n1,n2); A1o=zeros(n1,n-n2); Apr=cot(2,Ar,A1o); % Формирование матрицы Ao.
A2o=zeros(n-n1,n); Ao=cot(1,Apr,A2o); % Формирование матрицы Ao.
Bo=[14.*rand(n1,1); zeros(n-n1,1)]; % Первые n1 компонент числа, равномерно
% распределенные в диапазоне 0...14; а остальные 6900 – нули.
Fo=(-1).^fix(10.*rand(1,n)).*6.*rand(1,n); % равномерное распределение компонент
% в диапазоне -2...0 с умножением некоторых на -1, когда fix(10.*rand(1,n))=нечетному числу.
Aeqo=[]; beqo=[];
Lbo=-2.*rand(1,n); % равномерное распределение компонент в диапазоне -2...0
Ubo=5.*rand(1,n); % равномерное распределение компонент в диапазоне 0...5
% Запуск универсального линпрогооптимизатора.
LinProgUnivesAvtomatMU % Текст-файл универсального линпрогооптимизатора, рис. 2.5.
toc
```

Рис. 2.7. Универсальное линпрогуказание, настроенное для минимума. Размерность задачи `n =7000`. Мат-

рица A_0 содержит первые 100×50 ненулевых элементов (5000), а остальные 48 995 000 – нули, т.е. нулей больше в 9799 раз. Матрица A_{eq0} не используется. Закомментированные элементы соответствуют оптимизатору рис. 2.4.

Вектор f_0 имеет 7000 компонент, равномерно случайно распределённых в диапазоне ± 6 . Матрица A_0 имеет 49 млн. компонент, из которых только первые 100×50 ненулевые, равномерно случайно распределённые в диапазоне $0 \dots 3$. Вектор b_0 имеет 100 ненулевых компонент, равномерно случайно распределённых в диапазоне $0 \dots 14$, а остальные 6900 – нули. Нижняя граница ц-области L_{b0} 7000-компонентный вектор с равномерно случайно распределёнными компонентами в диапазоне $-2 \dots 0$. А верхняя граница имеет компоненты, распределённые в диапазоне $0 \dots 5$. Вектор b_0 , формируемый исходной командой в этом указании (в отличие от указания рис. 2.4) имеет столбцовую структуру. Чтобы не вносить изменения в линпрогооптимизатор рис. 2.5, вектор b_0 подвергается дополнительной операции – транспонированию (символ – апостроф «'»), превращающий вектор-столбец в вектор-строку.

Для того, чтобы основные результаты оптимизации (x , y , ef) поместились в Комокно количество отображаемых в нём строк было увеличено с 5000 (по умолчанию) до 8000. Командами меню «File_Preferences» зывается панель настроек Матлаба и в поле «Number of ...» (см. рис. 2.10) устанавливается нужное значение. В дополнение к приведенному в линпрогооптимизаторе LinProgUniversAvtomatMU (рис. 2.5) закомментирована часть строки 49 для невыдачи в Комокно переменных pro и rvo , и эта строка выглядит так: `evalin('base', Nx), evalin('base', Ny), evalin('base', Nef), %evalin('base', Nnr), evalin('base', Nrv).`

Результат вычисления показан на рис. 2.8. В Комокно Матлаб выдал данные (рис. 2.8, А), состоящие из

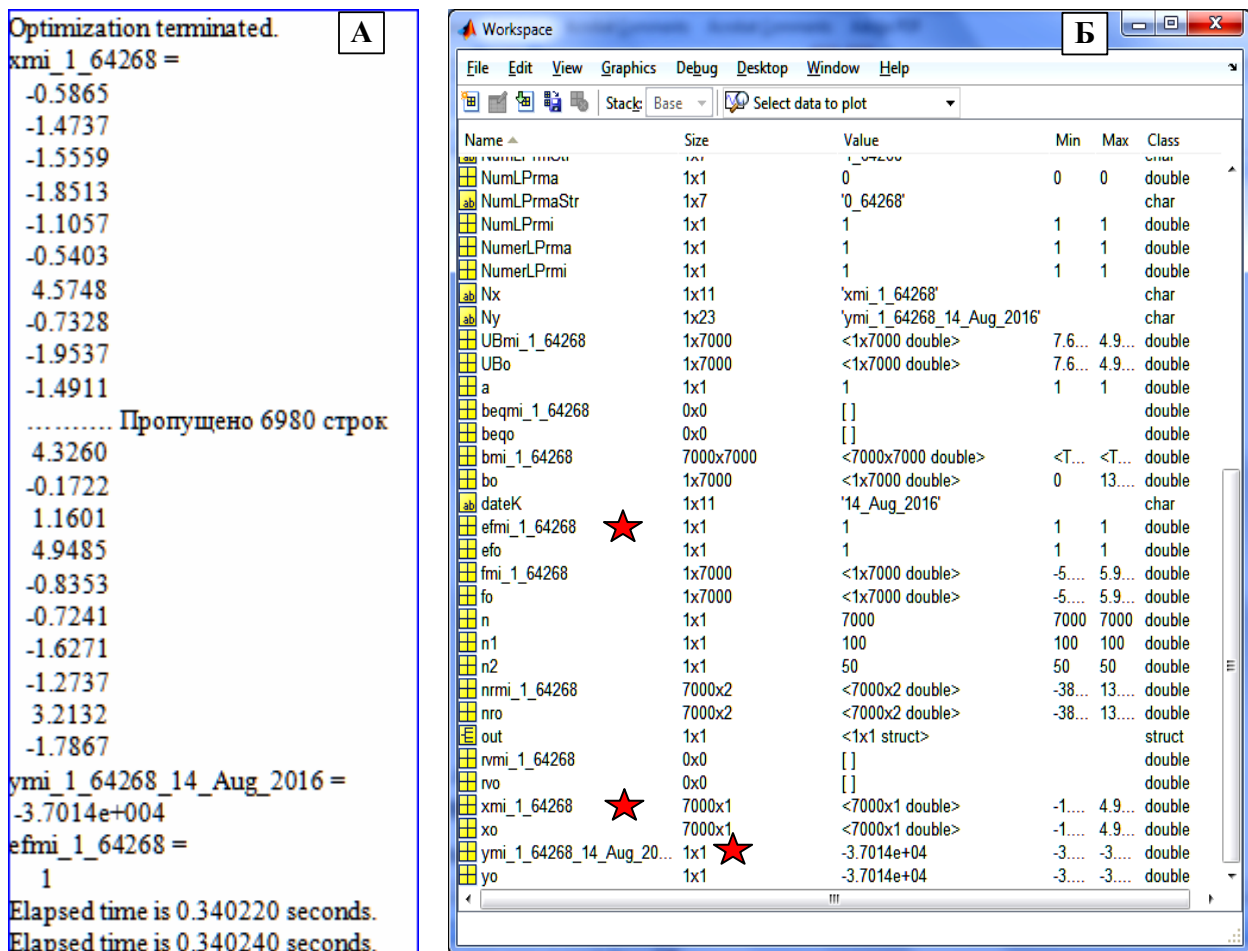


Рис. 2.8. Результат вычисления линпрогминимума по команд-указанию рис. 2.7 с использованием универсального линпрогооптимизатора LinProgUniversAvtomatMU рис. 2.5, в котором закомментирована часть строки 49 для невыдачи в Комокно переменных pro и rvo . Строка выглядит так: `evalin('base', Nx), evalin('base', Ny), evalin('base', Nef), %evalin('base', Nnr), evalin('base', Nrv).` Кроме того, число строк в Комокне установлено 8000 вместо 5000 по умолчанию.

Оптимизационного вектора xmi_1_64268 в 7000 строк (показаны 10 первых и 10 последних), минимального значения ц-функции $y_{mi_1_6428_14_Aug_2016}$ и флага оптимизации $ef_{mi_1_64268}$. Выданные в Комокно переменные в Вычпространстве (рис.2.8, Б) отмечены звёздочками. Окно Вычпространства (рис. 2.8, Б) отобра-

жает все объекты только одного запуска линпрогминимизации. Объем сохранённого файла WS_7000_AoMin.mat равен 3 МБ.

Время вычисления линпрогминимума задачи составляет 0,34 с. Системная плата: Hewlett-Packard HP Pavilion g6 Notebook PC. Тип ЦП: Mobile DualCore Intel Core i5, 2900 MHz (29 x 100), системная память 4000 МБ (DDR3-1600 DDR3 SDRAM).

2.4.3.2. Линпрогмаксимум, Ao из 1-й задачи, Aeqo с ненулевыми значениями 300x500

Набор задачи – на рис. 2.9. Результаты – на рис. 2.11. Как следует из рис. 2.11, А время вычисления

```
%
% Универсальное линпрогуказание LinProgUnivUkz

% Настройки универсального линпрогоптимизатора LinProgUnivesAvtomatMU
tic
NumLPmi=0; % Нумератр объектов линпрогмина очистить при необх. Новой нумерации.После 1 пуска – заком-
ментировать.
NumLPma=0; % Нумератор объектов линпрогмакса очистить при необх. Новой нумерации.После 1 пуска – за-
комментировать.
% Lmi_Lma=1; % Поиск линпрогмина.Закоммент. при поиске линпрогмакса.
Lmi_Lma=-1; % Поиск линпрогмакса. Закоммент. При поиске линпрогмина.
% options=optimset('param1',число,'param2','строка-значение',...); % Изменение настроек линпрогфункции Матла-
ба linprog(...): раскомментировать строку и заполнить шаблон optimset(...).

% Аргументы линпрогогфункции Матлаба linprog(...)
% fo = [-5.6 4.4 6 -3 9];
% Ao = [1 -1 1 2 3; 3 2 4 1 -2; 3 2 0 4 3; 1 3 5 4 -2; 3 4 7 1 -2 ]; bo = [6 2.3 17 6.5 11];
% Aeqo = [0 1.5 0 0 2; 3 1 2.5 0 0.7; 4.1 -2.3 11 3.4 4.7]; beqo = [7.82 3.62 7.62];
% Lbo = zeros(1,5); Ubo = [6.5 4.5 5 8.2 3.1];

% Возможности Универсального линпрогоптимизатора. Вычисление для n =7000

n =7000;n1=100;n2=50;Ar=3.*rand(n1,n2);A1o=zeros(n1,n-n2);Apr=cat(2,Ar,A1o);
A2o=zeros(n-n1,n);Ao=cat(1,Apr,A2o);
bo=[14.*rand(n1,1);zeros(n-n1,1)]; % Первые n1 компонент числа, равномерно
% распределенные в диапазоне 0...14; а остальные 6900 – нули.
Fo=(-1).^fix(10.*rand(1,n)).*6.*rand(1,n); % равномерное распределение компонент
% в диапазоне -2...0 с умножением некоторых на -1, когда
% fix(10.*rand(1,n))=нечетному числу
n3=300;n4=500;Ar1=4.*rand(n3,n4);A3o=zeros(n3,n-n4);Apr1=cat(2,Ar1,A3o); % Формирование матрицы Aeqo.
A4o=zeros(n-n3,n);Aeqo=cat(1,Apr1,A4o); % Формирование матрицы Aeqo.
Beqo=[10.*rand(n3,1);zeros(n-n3,1)]; % Первые n3 компонент числа, равномерно
% распределенные в диапазоне 0...10; а остальные 6700 – нули.
Lbo=-2.*rand(1,n); % равномерное распределение компонент в диапазоне -2...0
Ubo=5.*rand(1,n); % равномерное распределение компонент в диапазоне 0...5

% Запуск универсального линпрогоптимизатора.
LinProgUnivesAvtomatMU % Текст-файл универсального линпрогоптимизатора, рис. 2.5.
toc
```

Рис. 2.9. Универсальное линпрогуказание, настроенное для максимум. Размерность задачи $n = 7000$. Матрица Ao из предыдущей задачи. Матрица Aeqo в левом верхнем углу содержит 300x500 ненулевых значений, остальные 48 985 000 – нули. Помимо Aeqo и beqo остальные объекты – из задачи рис. 2.7.

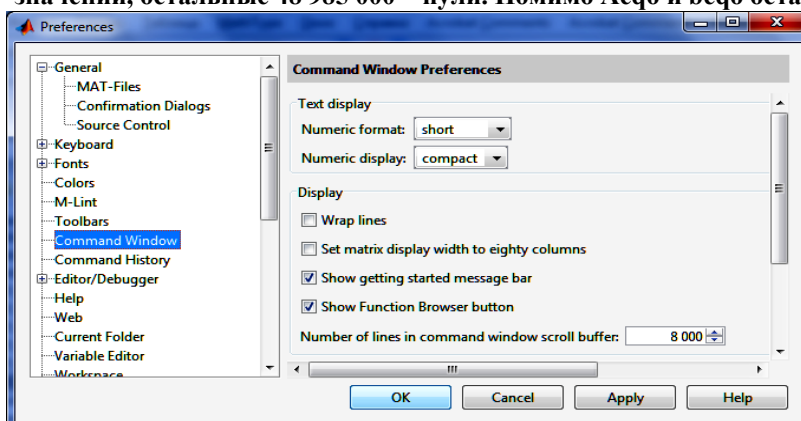


Рис. 2.10. Настройка числа строк в Командном окне на 8000. По умолчанию – 5000, максимальное число – 25 000.

Линпрогмаксимума составило 1,97 с, что превышает время решения предыдущей задачи в 5,8 раз. Увеличение

времени связано с возросшим объёмом обрабатываемой информации из-за наличия ограничений-равенств $Aeqo$ и $beqo$. Файл Вычпространства этого вычисления $WS_7000_AoAeqoMax.mat$ занимает объём 3 МБ, как и в предыдущей задаче.

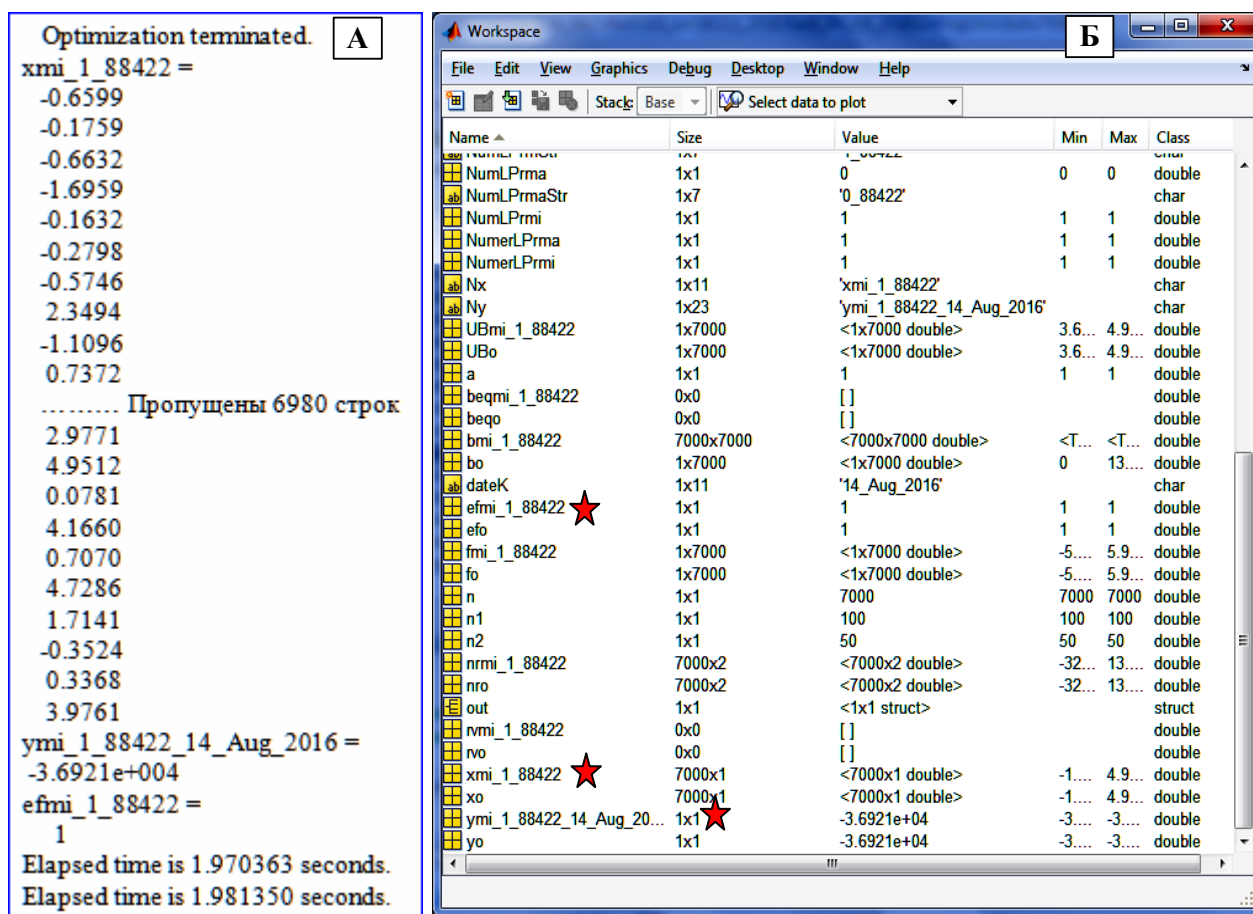


Рис. 2.11. А – результат вычисления линпрогмаксимума по командоваанию рис. 2.9 с использованием универсального линпрогоптимизатора рис. 2.5. Звездочками отмечены переменные из Комокна. Вычпространство (Б) содержит объекты только этого вычисления и занимает объём 3 МБ.

2.4.4. Универсальный линпрогоптимизатор с выдачей единой итог-таблицы запуска оптимизации

Универсальный линпрогоптимизатор $LinProgUnivAvtomatMU$ рис. 2.5 после выполнения оптимизации выдаёт в Комокно и в Вычпространство 5 выходных объектов с уникальным идентификатором, не объединённых единой формой представления. При дальнейшей работе с такими объектами в новых сессиях одновременный их вызов возможен только путём индивидуального обращения, что не всегда удобно. Этот же недостаток при копировании результата в текстовый документ принуждает к дополнительному табличному форматированию результата средствами редактора документа. Другим недостатком оптимизатора является отсутствие прямой связи между результатом и условиями его получения, что в значительной степени затрудняет решение серии линпрогзадач в одной сессии Матлаба. Эти недостатки устранены в линпрогоптимизаторе с выдачей итог-таблицы и копировании линпрогуказания результата с именем файла итогтаблицы в действующую папку.

2.4.4.1. Описание универсальных итогтабличных линпрогуказания $LinProgUnivUkz_Tab$ и линпрогоптимизатора $LinProgUnivAvtomatMU_Tab$

На рис. 2.12 и 2.13 представлен линпрогкомплекс (линпрогуказание и линпрогоптимизатор), обеспечивающий выдачу выходных объектов «россыпью» и в единой итогтаблице (рис. 2.14), а также копию линпрогуказания с именем файла итогтаблицы в действующую папку (рис. 2.15).

```

tic
fo=[];Ao=[];bo=[];Aeqo=[];beqo=[];Lbo=[];Ubo=[]; %Очистка значений предыдущего запуска.
NumLPPrmi=0; % Нумератор объектов линпрогмина очистить при необх. новой нумерации. После 1
пуска – закомментировать.
NumLPPrma=0; % Нумератор объектов линпрогмакса очистить при необх. новой нумерации. После 1
пуска – закомментировать.
Lmi_Lma=1; % Поиск линпрогмина. Закоммент. при поиске линпрогмакса.
% Lmi_Lma=-1; % Поиск линпрогмакса. Закоммент. при поиске линпрогмина.
ItogTabLPr=1; % Запуск выдачи итог-тпблицы оптимизации – закомментировать, если итог-таблицы
не нужны, и раскомментировать следующую строку.
% ItogTabLPr=-1; % Стирает в Вычпространстве 1, прекращая выдачу итог-таблиц оптимизатором
% LinProgUniversAvtomatMU_Tab.
% options=optimset('param1',число,'param2','строк-значение',...); % Изменение настроек линпрогфунк-
ции Матлаба linprog(...):раскомментировать строку и заполнить шаблон optimset(...).
ImjaKopirFailLPukaz='LinProgUnivUkz_Tab.m'; % Имя копируемого в дейст. Папку файла линпрогука-
% зания (с именем ф. итогтаблицы).

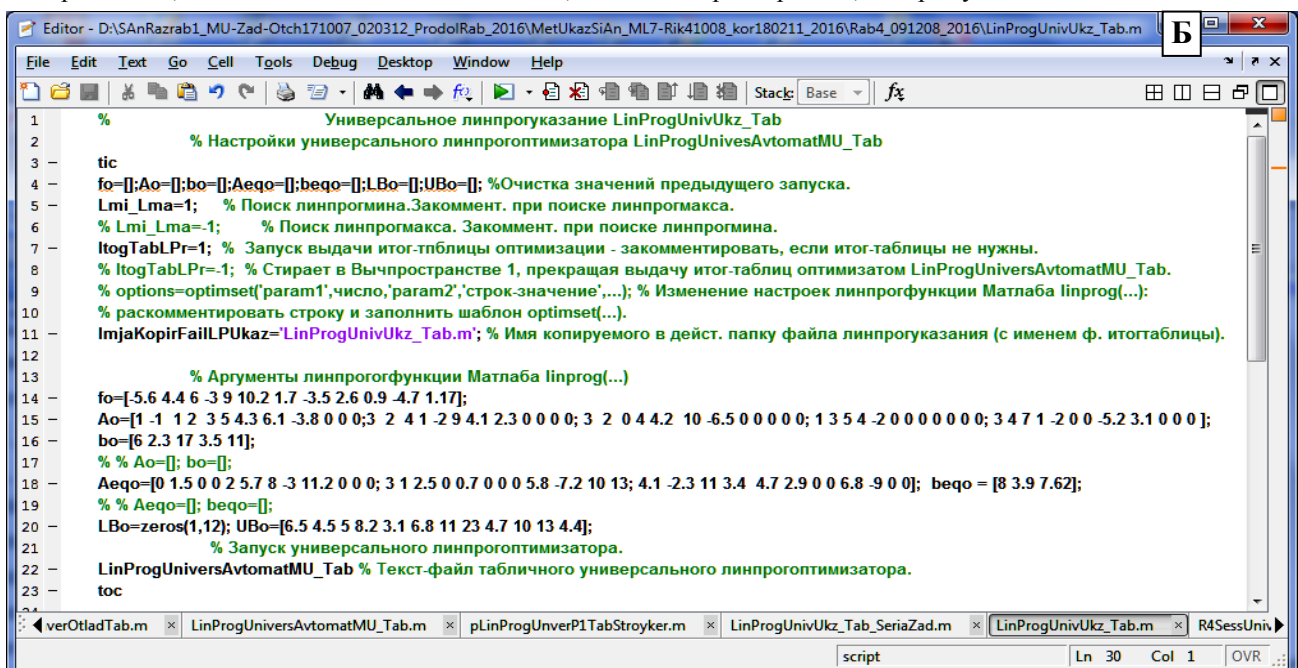
% 2. Аргументы линпрогфункции Матлаба linprog(...)
fo=[-5.6 4.4 6 -3 9 10.2 1.7 -3.5 2.6 0.9 -4.7 1.17];
Ao=[1 -1 1 2 3 5 4.3 6.1 -3.8 0 0 0; 3 2 4 1 -2 9 4.1 2.3 0 0 0 0; 3 2 0 4 4.2 10 -6.5 0 0 0 0 0; 1 3 5 4 -2 0 0 0 0 0 0
0; 3 4 7 1 -2 0 0 -5.2 3.1 0 0 0 ];
bo=[6 2.3 17 3.5 11];
% Ao=[]; bo=[];
Aeqo=[0 1.5 0 0 2 5.7 8 -3 11.2 0 0 0; 3 1 2.5 0 0.7 0 0 0 5.8 -7.2 10 13; 4.1 -2.3 11 3.4 4.7 2.9 0 0 6.8 -9 0 0]; beqo
=[8 3.9 7.62];
% Aeqo=[]; beqo=[];
Lbo=zeros(1,12); Ubo=[6.5 4.5 5 8.2 3.1 6.8 11 23 4.7 10 13 4.4];

% 3. Запуск универсального линпрогоптимизатора с итог-таблицей.
LinProgUniversAvtomatMU_Tab % Текст-файл табличного универсального линпрогоптимизатора.
Toc

```

Рис. 2.12. Линпрогуказание LinProgUnivUkz_Tab (исполняемый код, А) настроено на поиск минимума (выполняемого оптимизатором LinProgUniversAvtomatMU_Tab – рис. 2.13). Обеспечивает выдачу выходных объектов x,y,ef, pr и gv в виде группы в Комокно и в виде итог-таблицы в html-формате (рис. 2.14), а также копирование самого линпрогуказания с именем файла итогтаблицы (рис. 2.15). Продолжение рисунка – на следующей стр.

Это указание создано на базе указания рис. 2.4 с добавлением команд управления выдачей итог-таблицы «ItogTabLPr=1», «ItogTabLPr=-1», 2 строк очистки линейных неравенств и равенств для задач, где это требуется. Кроме того, изменён вид задачи оптимизации, включая её размерность, которая увеличена с 5 до 12.



Продолжение рис. 2.12. . Линпрогуказание LinProgUnivUkz_Tab – снимок с окна Редактора с нумерацией строк (Б). Строка 11 позволяет изменять имя файла линпрогуказания: новое имя вместе с

расширением «.m» следует вставить в кавычки вместо существующего LinProgUnivUkz_Tab.m – пример изменения имени линпрогоуказания – см. рис. 2.19 в разд. 2.6.

Наличие прямой связи между файлом итогтаблицы и файлом условий получения итогтаблицы (в виде одинаковых имён обоих файлов; расширения разные – см. на рис. 2.15 пару файлов в штрих-рамке) позволяет значительно сократить трудоёмкость решения серии линпрогзадач в одной сессии Матлаба. Пример решения серии из 4 задач показан в разд. 2.6. Это происходит в следствие того, что не нужно вручную: копировать

```
% Универсальный линпрогооптимизатор LinProgUnivAvtomatMU_Tab
% С переключательным выбором типа оптимизации(мин/макс), с возможностью выдачи итогтаблицы в виде
% выходными переменными xmi/ xma, ymi/ yma, efmi/ efma, nrmi/ nrma, rvmi/ rvma и копии линпрогуказания с именем файла итог-
% таблицы, выдачей в Комокно значений выполненной линпрогооптимизации: выходных переменных x,y,ef и отладобъектов
% с nr и rv.
Tic; % Запуск таймера программы.
ZagStlb={};Nx_Cel={};Ny_Cell={};Nef_Cell={};Nnr_cell={}; Nrv_cell={};Ny_CellPustCh={}; Nx=[];%Очистка значений
предыдущего запуска.
A=Lmi_Lma;
nro=[]; rvo=[]; Nx_Cell={' '};Ny_Cell={' '};Nef_Cell={' '}; Nnr_cell={' '}; Nrv_cell=[];% Очистка выходных объектов от ре-
зультата предыдущего вычисления.
Ny_CellPustCh={' '};Nef_Cell={' '};
SumDatKor=datestr(clock); SumDatKor(3)='_';SumDatKor(7)='_'; SumDatKor(15)='h';% Дата и время запуска линпр-
гооптимизатора с заменой %Rik
SumDatKor(18)='m';SumDatKor=[SumDatKor 's']; % дефисов на нижн. Подчёркивание и добавление во время букв h
(час), m (мин), s (сек).
[xo,yo,efo,out]= linprog(a*fo,Ao,bo,Aeqo,beqo,Lbo,Ubo); % Вычисление линпрогооптима.

% Вычисление отладочных выходных объектов: векторов огранич.-неравенств и огранич.-равенств.
If isempty(Ao) % Проверка пустого значения матрицы Ao.
    Nro=[];
else
    nro=[Ao*xo, bo'];
end
if isempty(Aeqo) % Проверка пустого значения матрицы Aeqo.
    Rvo=[];
else
    rvo=[Aeqo*xo, beqo'];
end
if Lmi_Lma==1% Формирование объектов линпрогминимума
    Nf=['fmi'];Nx=['xmi'];Ny=['ymi'];Nef=['efmi']; % Создание имён вых.переменным и объектов
    Nnr=['nrmi'];Nrv=['rvmi']; % линпрогминимизации, если она задана.
    Assignin('base', Nf, fo); % Присвоение созданным именам данных текущей линпрогминимизации,
    assignin('base', Nx, xo);assignin('base', Ny, yo);assignin('base', Nef, efo); % с записью в Вычпространстве,
    assignin('base', Nnr, nro);assignin('base', Nrv, rvo); % если линпрогминимизация задана.
    Evalin('base', Nx), evalin('base', Ny), evalin('base', Nef), evalin('base', Nnr), evalin('base', Nrv) % Выдача в Комокно
    текущих
% значений линпрогминимиз. Xmi,ymi,efmi,nrmi и rvmi, если она задана.
If ItogTabLPr==1% Формирование итогтаблицы линпрогминимума
    ImjaFailTabLPrmi=['TabLPrmi_' SumDatKor]; %Rik
    ZagTabLPrmi=char(['Табл.ЛинПрогМин ' blanks(1) SumDatKor]);%Rik
    ZagStlb={Nx Ny Nef [Nnr 'A*xmi'] [Nnr 'b'] [Nrv 'Aeq*xmi'] [Nrv 'beq']}; % Ячк-имена столбцов итог-табл.
    Nx_Cell=num2cell(evalin('base', Nx)); % Преобразование числ-столбца xmi в ячк-столбец.
    Ny_CellPustCh(1:length(evalin('base', Nx))-1,1)={' '}; % Создание пустой части ячк-столбца ymi.
    Ny_Cell=[num2cell(evalin('base', Ny)); Ny_CellPustCh]; %Создание ячк-столбца ymi.
    Nef_Cell=[num2cell(evalin('base', Nef)); Ny_CellPustCh]; % Создание ячк-столбца efmi.
    If isempty(Ao) % Проверка пустого значения матрицы Ao.
        Nnr_cell=[{' '},{ ' '};Ny_CellPustCh,Ny_CellPustCh]; % Формиров. Пустой ячк-матрицы Nnr.
    Elseif length(bo)<length(xo)
        Nnr_cell=[num2cell(evalin('base', Nnr));Ny_CellPustCh(1:length(xo)-length(bo)),...
        Ny_CellPustCh(1:length(xo)-length(bo))]; % Формиров ячк-матрицы Nnr, дополнен. Пустыми строками
        % до размерности xo.
    Else
        Nnr_cell=num2cell(evalin('base', Nnr)); % Формиров ячк-матрицы Nnr, совпадающей по размерности с xo.
    End
    if isempty(Aeqo) % Проверка пустого значения матрицы Aeqo.
        Nrv_cell=[{' '},{ ' '};Ny_CellPustCh,Ny_CellPustCh]; % Формиров. Пустой ячк-матрицы Nrv.
    Elseif length(beqo)<length(xo)
        Nrv_cell=[num2cell(evalin('base', Nrv));Ny_CellPustCh(1:length(xo)-length(beqo)),...
        Ny_CellPustCh(1:length(xo)-length(beqo))]; % Формиров ячк-матрицы Nrv, дополнен. Пустыми строка-
ми
        % до размерности xo.
    Else
        Nrv_cell=num2cell(evalin('base', Nrv)); % Формиров ячк-матрицы Nrv, совпадающей по размерности с xo.
```

```

End
Itog_TabCellLPrmi=[ZagStlb;Nx_Cell,Ny_Cell,Nef_Cell,Nnr_cell,Nrv_cell]; % Формирование ячейковой итог-
табл. запуска.
Copyfile(ImjaKopirFailLPukaz, [ ImjaFailTabLPrmi '.m'],'f') % Копирование ф. LinProgUnivUkz_Tab.m с име-
нем файла итогтаб. Линпрогмина.
Report ItogTabLPrmi.rpt % Команда работает после создания посредством Report Explorer'a и настройки шаб-
лона таблицы TabLPrmi.
End
end
if Lmi_Lma== -1 % Формирование объектов линпрогмаксимума
Nf=['fma']; Nx=['xma']; Ny=['yuma']; Nef=['efma']; % Создание имён вых.переменным и объектов
Nnr=['nrma']; Nrv=['rvma']; % линпрогмаксимизации, если она задана.
Assignin('base', Nf, fo); % Присвоение созданным именам данных текущей
assignin('base', Nx, xo); assignin('base', Ny, -yo); assignin('base', Nef, efo); % с записью в Вычпространство,
assignin('base', Nnr, nro); assignin('base', Nrv, rvo); % если линпрогмаксимизация задана.
Evalin('base', Nx), evalin('base', Ny), evalin('base', Nef), evalin('base', Nnr), evalin('base', Nrv) % Выдача в Комокно
текущих
% значений линпрогмаксимиз. Xma,yuma,efma,nrma и rvma, если она задана.
If ItogTabLPr==1 % Формирование итогтаблицы линпрогмаксимума
ImjaFailTabLPrma=['TabLPrma_' SumDatKor];
ZagTabLPrma=char(['Табл.ЛинПроГМакс:' blanks(1) SumDatKor]);
ZagStlb={Nx Ny Nef [Nnr ' _A*xma'] [Nnr ' _b'] [Nrv ' _Aeq*xma'] [Nrv ' _beq']}; % Ячк-имена столбцов итог-табл.
Nx_Cell=num2cell(evalin('base', Nx)); % Преобразование числ-столбца xma в ячк-столбец.
Ny_CellPustCh(1:length(evalin('base', Ny))-1,1)={' '}; % Создание пустой части ячк-столбца ума.
Ny_Cell=[num2cell(evalin('base', Ny)); Ny_CellPustCh]; %Создание ячк-столбца ума.
Nef_Cell=[num2cell(evalin('base', Nef)); Ny_CellPustCh]; % Создание ячк-столбца efma.
If isempty(Ao) % Проверка пустого значения матрицы Ao.
Nnr_cell=[{ ' },{ ' '};Ny_CellPustCh,Ny_CellPustCh]; % Формиров. Пустой ячк-матрицы Nnr.
Elseif length(bo)<length(xo)
Nnr_cell=[num2cell(evalin('base', Nnr));Ny_CellPustCh(1:length(xo)-length(bo)),...
Ny_CellPustCh(1:length(xo)-length(bo))]; % Формиров ячк-матрицы Nnr, дополнен. Пустыми строками
% до размерности xo.
Else
Nnr_cell=num2cell(evalin('base', Nnr)); % Формиров ячк-матрицы Nnr, совпадающей по размерности с xo.
End
if isempty(Aeqo) % Проверка пустого значения матрицы Aeqo.
Nrv_cell=[{ ' },{ ' '};Ny_CellPustCh,Ny_CellPustCh]; % Формиров. Пустой ячк-матрицы.
Nrv.num2cell(evalin('base', Nrv))
elseif length(beqo)<length(xo)
Nrv_cell=[num2cell(evalin('base', Nrv));Ny_CellPustCh(1:length(xo)-length(beqo)),...
Ny_CellPustCh(1:length(xo)-length(beqo))]; % Формиров ячк-матрицы Nrv, дополнен. Пустыми строка-
ми
% до размерности xo.
Else
Nrv_cell=num2cell(evalin('base', Nrv)); % Формиров ячк-матрицы Nrv, совпадающей по размерности с xo.
End
Itog_TabCellLPrma=[ZagStlb;Nx_Cell,Ny_Cell,Nef_Cell,Nnr_cell,Nrv_cell]; % Формирование ячейковой итог-
табл. запуска.
Copyfile(ImjaKopirFailLPukaz, [ ImjaFailTabLPrma '.m'],'f') % Копирование ф. LinProgUnivUkz_Tab.m с име-
нем файла итогтаб. Линпрогмакса.
Report ItogTabLPrma.rpt % Команда работает после создания посредством Report Explorer'a и настройки
шаблона таблицы TabLPrma.
End
end
toc % Считывание времени оптимизации.

```

Рис. 2.13, А. Универсальный линпрогоптимизатор LinProgUniversAvtomatMU_Tab (программный код) с выдачей выходных объектов в группе в Комокно и в виде выходной таблицы для каждой оптимизации (см. рис. 2.14), а также копирование самого линпрогуказания с именем файла итогтаблицы (см. рис. 2.15). Снимок окна редактора с линпрогоптимизатором LinProgUniversAvtomatMU_Tab с нумерацией строк – на рис. 2.13, Б.

выходные объекты x,y,ef, ng и gv в Уорд-файл и объединять их в таблицу, копировать линпрогуказание с условиями результата, давать имена в соответствии с номером решённой задачи и типом оптимизации Уорд-файлу с таблицей и копии линпрогуказания.

Линпрогоптимизатор LinProgUniversAvtomatMU_Tab построен на базе универсального линпрогоптимизатора LinProgUniversAvtomatMU с добавлением функции создания итогтаблиц выходных объектов и копий линпрогуказаний, создавших эти таблицы.

Формирование итог-таблиц производится, если в табличном линпрогуказании LinProgUnivUkz_Tab (рис. 2.12) раскомментирована строка с командой «ItogTabLPr=1». Таблица для минимума формируется в if-сегменте

программы LinProgUniversAvtomatMU_Tab в строках 33...62 (см. рис. 2.13, Б). Таблица для максимума – в строках 72...101. Таблицы будут выдаваться командами «report ItogTabLPrmi.rpt/ report ItogTabLPrma.rpt», если будут созданы посредством Report Explorer'a и настроены шаблоны соответствующих таблиц TabLPrmi/ TabLPrma (см. разд. 2.4.4.2).

Копии файлов линпрогуказаний, сформировавшие итогтаблицы минимума и максимума, создаются с именами файлов итогтаблиц в строках с номерами соответственно 60 и 99.

В if-сегменте (для линпрогминимума) формируются следующие элементы итог-таблицы: имя html-файла таблицы вида «TabLPrmi_21_Sep_2016 12h40m21s.html» (строки 10, 11, 34 – рис.2.13, Б), заголовок итогтаблицы вида «ТаблЛинПроГМин_ 21_Sep_2016 12h40m21s» (строка 35), ячейковая структура (ячк-структура) столбцов выходных данных таблицы вида рис. 2.15 (строки 36...58), итог-таблица в ячк-формате в виде значения переменной «Itog_TabCellLPrmi» в Вычпространстве (строка 59), копия линпрогуказания LinProgUnivUkz_Tab.m с именем файла итогтаблицы (строка 60) и итог-таблица в html-формате (строка 61), формируемая в действующей папке с именем файла из строки 34 (рис. 2.13, Б).

В Вычпространстве сохраняется: только имена файлов итогтаблиц последнего запуска (для минимума и максимума) с шаблонными именами ImjaFailTabLPrmi/ ImjaFailTabLPrma и только итогтаблицы в ячейковом формате (для минимума и максимума) последнего запуска с шаблонными именами Itog_TabCellLPrmi/ Itog_TabCellLPrma. Все озданные html-итогтаблицы и копии линпрогуказаний с именами файлов итогтаблиц хранятся в Действующей папке – см. рис. 2.15, Б.

При формировании Даты в имя html-таблицы (строки 10) из-за особенности оператора объединения (квадратные скобки) пришлось внести изменения в значение даты, генерируемой Матлабом в команде «datestr(clock)»: дефисы заменить нижним подчёркиванием, а два двоеточия заменить соответственно на «h» и «m» час и мин и добавить ещё сек «s» (строка 11).

При формировании пустых матриц Nnr и Nrv (строки 42 и 51) оказалось, что генерируемое значение {} командами num2cell(evalin('base', Nnr)) и num2cell(evalin('base', Nrv)) воспринимается оператором объединения ([...]) как ошибка. В результате они были заменены двумя значениями {' ',' ' }.

If-сегмент формирования итог-таблицы линпрогмаксимума аналогичен структурно описанному if-сегменту линпрогминимума.

```

Editor - D:\SAnRazrab1_MU-Zad-Otch171007_020312_ProdolRab_2016\MetUkazSiAn_ML7-Rik41008_kor180211_2016\Rab4_091208_2016\LinProgUniversAvtomatMU_Tab.m*
File Edit Text Go Cell Tools Debug Desktop Window Help
% Универсальный линпрогоптимизатор LinProgUniversAvtomatMU_Tab
% С перекрывающимися выбором типа оптимизации(мин/макс), с возможностью выдачи итоговой таблицы оптимизации с
% выходными переменными xmi/ xma, ymi/ yma, efmi/ efma, nrmi/ nrma, rvmi/ rvma и копии линпрогказания с именем файла итоговой таблицы,
% с выдачей в Командно значений выполненной линпрогоптимизации: выходных переменных x,y,ef и отладочных объектов ng и gv.
tic; % Запуск таймера программы.
ZagStlb={};Nx_Cell={};Ny_Cell={};Nef_Cell={};Nnr_cell={};Nrv_cell={};Ny_CellPustCh={};Nx=[];%Очистка значений предыдущего запуска.
a=Lmi_Lma;
nro=[]; rvo=[]; Nx_Cell=[];Ny_Cell=[];Nef_Cell=[]; Nnr_cell=[]; Nrv_cell=[];% Очистка выходных объектов от результата предыдущего вычисления.
Ny_CellPustCh=[];Nef_Cell=[];
SumDatKor=datestr(clock); SumDatKor(3)='-';SumDatKor(7)='-'; SumDatKor(15)='h';% Дата и время запуска линпрогоптимизатора с заменой %Rik
SumDatKor(18)='m';SumDatKor=[SumDatKor '$']; % дефисов на нижн. подчеркивание и добавление во время букв h (час), m (мин), s (сек).
[xo,yo,efo,outf]=linprog(a'fo,Ao,bo,Aeqo,beqo,LBo,UBo); % Вычисление линпрогоптимума.
% Вычисление отладочных выходных объектов: векторов огранич.-неравенств и огранич.-равенств.
if isempty(Ao) % Проверка пустого значения матрицы Ao.
nro=[];
else
nro=[Ao*xo, bo];
end
if isempty(Aeqo) % Проверка пустого значения матрицы Aeqo.
rvo=[];
else
rvo=[Aeqo*xo, beqo];
end
if Lmi_Lma==1% Формирование объектов линпрогминимума
Nf=[fmi];Nx=[xmi];Ny=[ymi];Nef=[efmi]; % Создание имен вых.переменным и объектов
Nnr=[nrmi];Nrv=[rvmi]; % линпрогминимизации, если она задана.
assignin('base', Nf, fo); % Присвоение созданным именам данных текущей линпрогминимизации,
assignin('base', Nx, xo);assignin('base', Ny, yo);assignin('base', Nef, efo); % с записью в Вычпространстве,
assignin('base', Nnr, nro);assignin('base', Nrv, rvo); % если линпрогминимизация задана.
evalin('base', Nx), evalin('base', Ny), evalin('base', Nef), evalin('base', Nnr), evalin('base', Nrv) % Выдача в Командно текущих
значений линпрогминимиз. xmi,ymi,efmi,nrmi и rvmi, если она задана.
if ItogTabLP==1% Формирование итоговой таблицы линпрогминимума
ImjaFailTabLPPrmi=[TabLPPrmi ' SumDatKor']; %Rik
ZagTabLPPrmi=char([TabLPPrmi' blanks(1) SumDatKor']);%Rik
ZagStlb=[Nx Ny Nef [Nnr 'A*xmi'] [Nnr 'b'] [Nrv 'Aeq*xmi'] [Nrv 'beq']]; % Ячк-имена столбцов итог-табл.
Nx_Cell=num2cell(evalin('base', Nx)); % Преобразование числ-столбца xmi в ячк-столбец.
Ny_CellPustCh(1:length(evalin('base', Nx))-1,1)=''; % Создание пустой части ячк-столбца ymi.
Ny_Cell=num2cell(evalin('base', Ny)); Ny_CellPustCh; %Создание ячк-столбца ymi.
Nef_Cell=[num2cell(evalin('base', Nef)); Ny_CellPustCh]; % Создание ячк-столбца efmi.
if isempty(Ao) % Проверка пустого значения матрицы Ao.
Nnr_cell={};Ny_CellPustCh,Ny_CellPustCh; % Формиров. пустой ячк-матрицы Nnr.
elseif length(bo)<length(xo)
Nnr_cell=[num2cell(evalin('base', Nnr));Ny_CellPustCh(1:length(xo)-length(bo))];...
Ny_CellPustCh(1:length(xo)-length(bo)); % Формиров ячк-матрицы Nnr, дополнен. пустыми строками
% до размерности xo.
else
Nnr_cell=num2cell(evalin('base', Nnr)); % Формиров ячк-матрицы Nnr, совпадающей по размерности с xo.
end
if isempty(Aeqo) % Проверка пустого значения матрицы Aeqo.
Nrv_cell={};Ny_CellPustCh,Ny_CellPustCh; % Формиров. пустой ячк-матрицы Nrv.
elseif length(beqo)<length(xo)
Nrv_cell=[num2cell(evalin('base', Nrv));Ny_CellPustCh(1:length(xo)-length(beqo))];...
Ny_CellPustCh(1:length(xo)-length(beqo)); % Формиров ячк-матрицы Nrv, дополнен. пустыми строками
% до размерности xo.
else
Nrv_cell=num2cell(evalin('base', Nrv)); % Формиров ячк-матрицы Nrv, совпадающей по размерности с xo.
end
Itog_TabCellLPPrmi=[ZagStlb,Nx_Cell,Ny_Cell,Nef_Cell,Nnr_cell,Nrv_cell];% Формирование ячк-матрицы итог-табл. запуска.
copyfile(ImjaKopirFailLPukaz, [ImjaFailTabLPPrmi '.m'],'f') % Копирование ф. LinProgUnivUkz_Tab.m с именем файла итогтаб. линпрогмина.
report ItogTabLPPrmi.rpt % Команда работает после создания посредством Report Explorer'a и настройки шаблона таблицы TabLPPrmi.
end
end
if Lmi_Lma==1 % Формирование объектов линпрогмаксимума
Nf=[fma];Nx=[xma];Ny=[yma];Nef=[efma];% Создание имен вых.переменным и объектов
Nnr=[nrma];Nrv=[rvma]; % линпрогмаксимизации, если она задана.
assignin('base', Nf, fo); % Присвоение созданным именам данных текущей
assignin('base', Nx, xo);assignin('base', Ny, yo);assignin('base', Nef, efo);% с записью в Вычпространство,
assignin('base', Nnr, nro);assignin('base', Nrv, rvo); % если линпрогмаксимизация задана.
evalin('base', Nx),evalin('base', Ny),evalin('base', Nef), evalin('base', Nnr), evalin('base', Nrv)% Выдача в Командно текущих
значений линпрогмаксимиз. xma,yma,efma,nrma и rvma, если она задана.
if ItogTabLP==1% Формирование итоговой таблицы линпрогмаксимума
ImjaFailTabLPPrma=[TabLPPrma ' SumDatKor'];
ZagTabLPPrma=char([TabLPPrma' blanks(1) SumDatKor']);
ZagStlb=[Nx Ny Nef [Nnr 'A*xma'] [Nnr 'b'] [Nrv 'Aeq*xma'] [Nrv 'beq']]; % Ячк-имена столбцов итог-табл.
Nx_Cell=num2cell(evalin('base', Nx)); % Преобразование числ-столбца xma в ячк-столбец.
Ny_CellPustCh(1:length(evalin('base', Nx))-1,1)=''; % Создание пустой части ячк-столбца yma.
Ny_Cell=[num2cell(evalin('base', Ny)); Ny_CellPustCh]; %Создание ячк-столбца yma.
Nef_Cell=[num2cell(evalin('base', Nef)); Ny_CellPustCh]; % Создание ячк-столбца efma.
if isempty(Ao) % Проверка пустого значения матрицы Ao.
Nnr_cell={};Ny_CellPustCh,Ny_CellPustCh; % Формиров. пустой ячк-матрицы Nnr.
elseif length(bo)<length(xo)
Nnr_cell=[num2cell(evalin('base', Nnr));Ny_CellPustCh(1:length(xo)-length(bo))];...
Ny_CellPustCh(1:length(xo)-length(bo)); % Формиров ячк-матрицы Nnr, дополнен. пустыми строками
% до размерности xo.
else
Nnr_cell=num2cell(evalin('base', Nnr)); % Формиров ячк-матрицы Nnr, совпадающей по размерности с xo.
end
end

```

```

89 - if isempty(Aeqo) % Проверка пустого значения матрицы Aeqo.
90 -     Nrv_cell={ ' ' };Ny_CellPustCh,Ny_CellPustCh; % Формиров. пустой ячк-матрицы. Nrv.num2cell(evalin('base', Nrv))
91 - elseif length(beqo)<length(xo)
92 -     Nrv_cell=num2cell(evalin('base', Nrv));Ny_CellPustCh(1:length(xo)-length(beqo)),...
93 -     Ny_CellPustCh(1:length(xo)-length(beqo)); % Формиров ячк-матрицы Nrv, дополнен. пустыми строками
94 -     % до размерности xo.
95 - else
96 -     Nrv_cell=num2cell(evalin('base', Nrv)); % Формиров ячк-матрицы Nrv, совпадающей по размерности с xo.
97 - end
98 - Itog_TabCellPrma=[ZagStlb;Nx_Cell,Ny_Cell,Nef_Cell,Nnr_cell,Nrv_cell]; % Формирование ячейковой итог-табл. запуска.
99 - copyfile(lmjaKopirFailLPukaz, [lmjaFailTabLPma '.m'],'f') % Копирование ф. LinProgUnivUkz_Tab.m с именем файла итогтаб. линпрогмакса.
100 - report ItogTabLPma.rpt % Команда работает после создания посредством Report Explorer'a и настройки шаблона таблицы TabLPma.
101 - end
102 - end
103 - toc % Считывание времени оптимизации.
104 -

```

Рис. 2.13, Б. Универсальный линпрогоптимизатор LinProgUniversAvtomatMU_Tab (снимок с окна Редактора Матлаба с номерами строк).

file:///D:/SAnRazrab1_MU-Zad-Otch171007_020312_ProdolRab_2016/MetUkazSiAn_ML7-Rik41008_kor180211_2016/Rab4_091208_2016/TabLPmi_21_Sep_2016 12h40m21s.html

Location: file:///D:/SAnRazrab1_MU-Zad-Otch171007_020312_ProdolRab_2016/MetUkazSiAn_ML7-Rik41008_kor180211_2016/Rab4_091208_2016/TabLPmi_21_Sep_2016 12h40m21s.html

Таблица 1. Табл.ЛинПроГМин_ 21_Sep_2016 12h40m21s

xmi	yumi	efmi	nrmi_A*xmi	nrmi_b	rvmi_Aeq*xmi	rvmi_beg
0.5540	-2.7571	1	-0.3797	6	8.0000	8
6.4202e-014			2.3000	2.3000	3.9000	3.9000
8.3716e-014			5.0156	17	7.6200	7.6200
0.7694			3.5000	3.5000		
0.0657			4.4779	11		
4.4650e-014						
1.7614e-013						
4.8265e-013						
0.7025						
0.2615						
1.6111e-013						
7.9786e-014						

Done

Рис. 2.14. Итогтаблица, выданная после запуска линпрогуказания LinProgUnivUkz_Tab из рис. 2.12, файл которой с именем TabLPmi_21_Sep_2016 12h40m21s.html записан в действующую папку – см. рис. 2.15, 1-я строка в штриховой рамке.

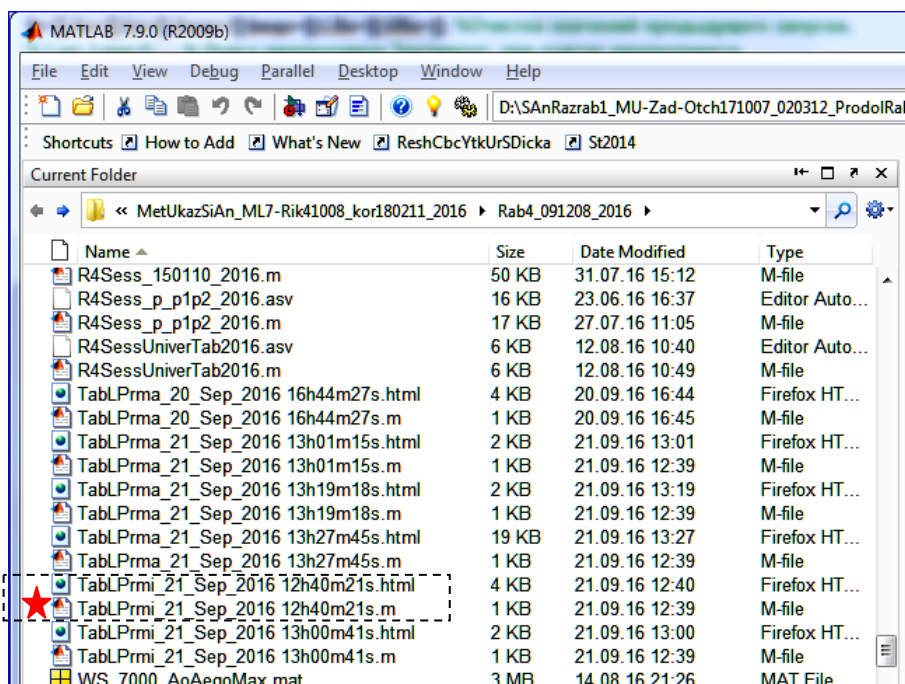


Рис. 2.15. Участок Действующей папки с рядом итогтаблиц и соответствующих им линпрогуказаний. Вышеприведенная итогтаблица рис. 2.14 и соответствующая ей копия линпрогуказания (помечена звёздочкой) выделены штрих-рамкой.

2.4.4.2. Формирование шаблонов итог-таблиц ItogTabLPrmi.rpt/ ItogTabLPrma.rpt/

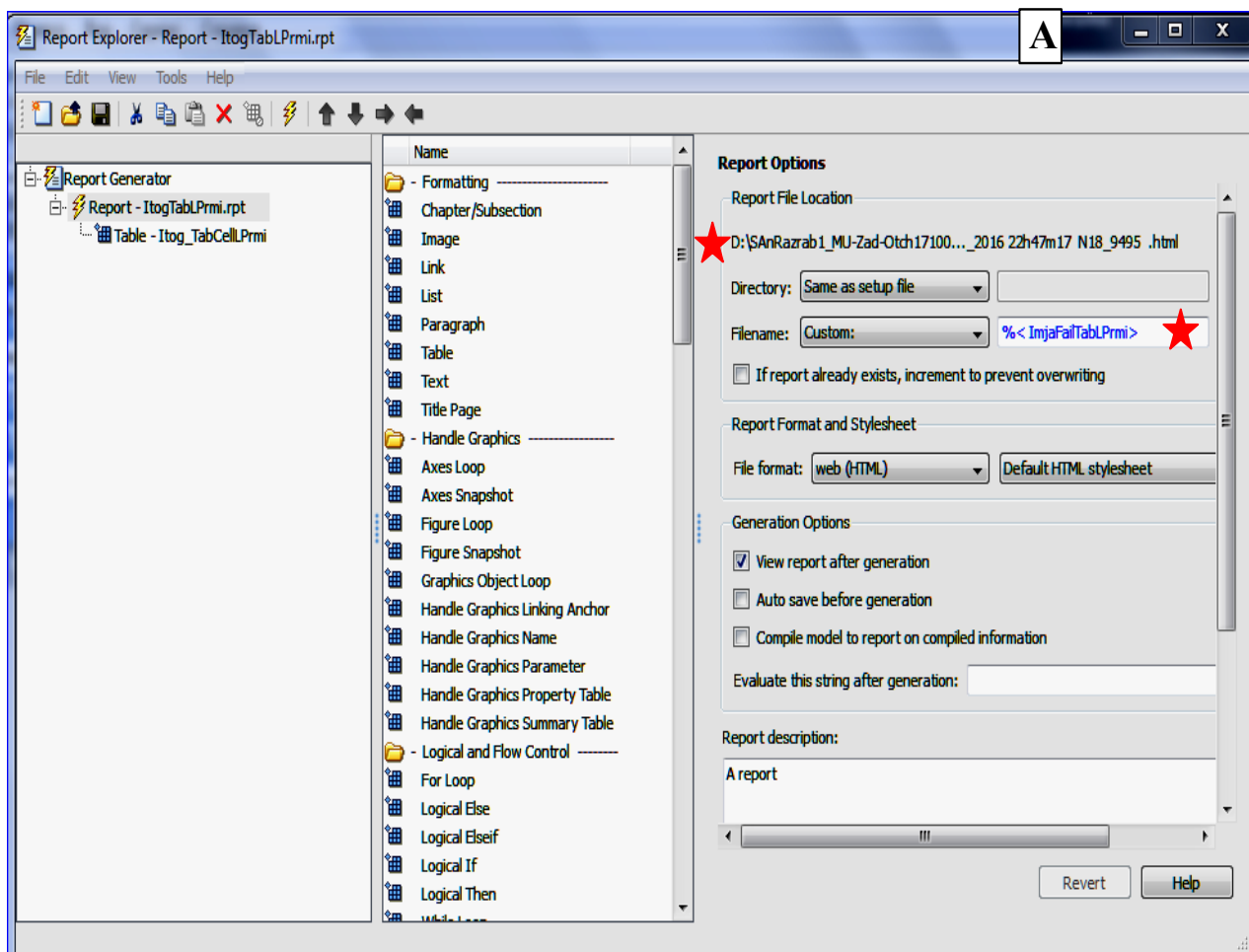
Хотя каждый шаблон формируется под таблицу конкретной задачи, в действительности он является универсальным и воспринимает изменение размерности задачи и другие отличия новых задач по сравнению с исходной. Шаблон таблицы формируется в Генераторе отчётов, который затем в программе LinProgUniversAvtomatMU_Tab (строки 61 и 100) преобразует итог-таблицы, выполненные в ячк-формате (строки 59 и 98) в html-формат: заявленные в MATLAB Report Generator'е другие формы таблицы (pdf, rtf и doc) в Матлабе 2009 не реализуются.

Формирование шаблона итог-таблиц ItogTabLPrmi.rpt и настройка Генератора отчётов производится в следующем порядке.

А. В оптимизаторе LinProgUniversAvtomatMU_Tab закомментируйте строки с командой report: 61 и 100. Проверьте, чтобы линпрогуказание LinProgUnivUkz_Tab было настроено на вычисление минимума с задачей таблицы аналогично рис. 2.12 (с аргумент-объектами Исполнителя или с теми, что приведены на рисунке). Кнопкой  запустите линпрогуказание LinProgUnivUkz_Tab. При удачном вычислении проверьте появление в Вычпространстве переменной Itog_TabCellLPrmi, т.е. итог-таблицы минимума в ячк-формате.

Б. Командами Start (кнопка в левом нижнем углу окна Матлаба)_Matlab_Matlab-Report-Generator_Report-File-List вызывается панель Генератора отчетов. На появившейся панели нажимается кнопка «Crtate and edit a new file», а в списке «Name» дважды щёлкается строка «Table»: панель примет вид, аналогичный рис. 2.16 А, без ввода настроек.

В. В дереве «Report Generator» выделите 1-ю ветвь «Report». Сохраните файл с именем ItogTabLPrmi.rpt. Проконтролируйте, чтобы имя сохранённого файла появилось в заголовке Репорт Эксплорера и в этой ветви.



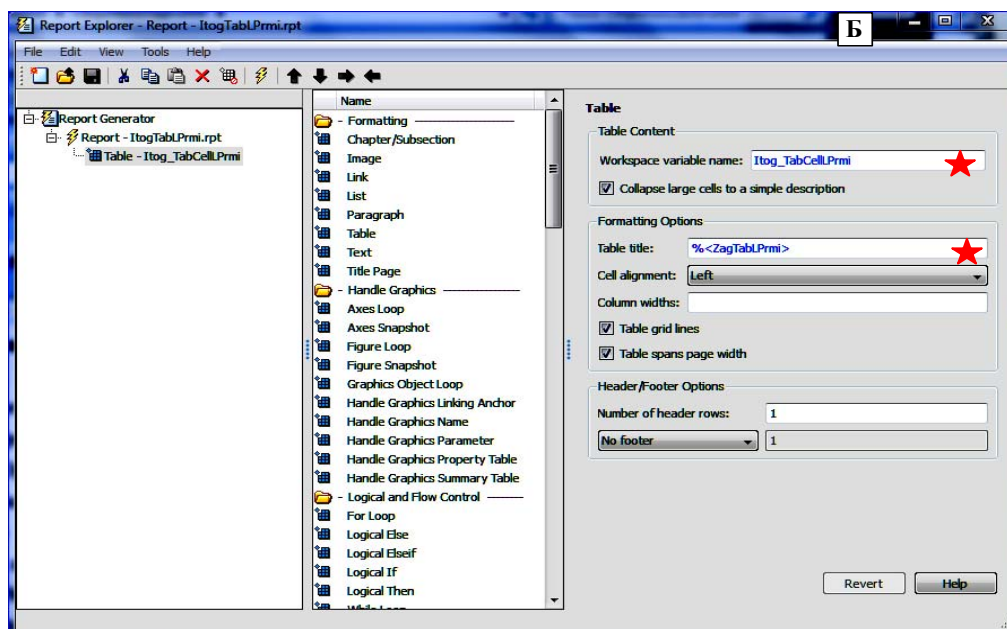


Рис. 2.16. Вид Репорт Эксплорера при выделении 1-й строки в дереве Репорт Генератора (А) после сохранения с заданным именем ItogTabLPrmi.rpt и ввода настройки во вторую строку со звездочкой. В упомянутую строку вставляется команд-имя файлов итогтаблиц линпрогминимума, создаваемое в строке 56 линпрогоптимизатора LinProgUniversAvtomatMU_Tab. Б – Вид Репорт Эксплорера при выделении 2-й строки в дереве Репорт Генератора. Звёздочками помечены строки, в которые нужно вставить значения, приведенные в этих строках: имя переменной итог-таблицы в ячк-формате из строки 59 оптимизатора LinProgUniversAvtomatMU_Tab и имя заголовка итог-таблицы из строки 57 в формате исполняемой команды.

Г. Во 2-ю строку на рис. 2.16,А, отмеченную звёздочкой, вставляется выражение `%< ImjaFailTabLPrmi>`, в угловых скобках которого стоит команд-имя файлов итог-таблиц линпрогминимума, создаваемое в строке 56 линпрогоптимизатора LinProgUniversAvtomatMU_Tab. Это команд-имя будет формировать имена html-файлов итог-таблиц, сохраняемых в действующей папке. Проконтролируйте: это имя должно появиться в 1-й строке раздела «Report Options», отмеченной звёздочкой. В дальнейшем в этом месте будет появляться имя последнего сохранённого файла итог-таблицы линпрогминимума (на рисунке это имя: «TabLPrmi_24_Apr_2016 22h47m17 N18_9495»).

Д. Выделяется 2-я строка в дереве Репорт генератора – см. рис. 2.16, Б. В строки, помеченные звёздочками вставляются указанные в них значения. Сохраните введённые изменения.

Е. Командами меню File_Generate-M-file запустите генерацию М-файла html-шаблона итоговой таблицы ItogTabLPrmi. Проконтролируйте появление в текущей папке файла buildItogTabLPrmi.m и его же появление в окне Редактора-отладчика.

Ж. Нажатием кнопки «Report»  на панели инструментов запускается Генератор отчетов на преобразование итог-таблицы из ячк- в html –формат: в действующей папке появятся файл со структурой имени типа «TabLPrmi_24_Apr_2016 22h47m17 N18_9495». Одновременно появится Окно сообщения о ходе преобразования и html-итог-таблица вида рис. 2.15, А. Настройка шаблона и Генератора отчетов для итог-таблицы линпрогминимума закончена.

З. Для формирования итог-таблицы и настройки Генератора отчётов для линпрогмаксимума следует повторить п. А...Ж, применяя имена и значения из if-сегмента максимума табличного оптимизатора LinProgUniversAvtomatMU_Tab – строки 72...101. Можно поступить иначе: вызвать новый шаблон итогтаблицы сохранить его с именем ItogTabLPrma.rpt, вставить в поля шаблона скопированные имена из шаблона ItogTabLPrmi.rpt с заменой буквы i на a (английское).

И. В завершение настройки следует снять закомментированность строк 61 и 100 в оптимизаторе LinProgUniversAvtomatMU_Tab.

2.4.4.3. Настройки при применении универсального итогтабличного линпрогооптимизатора LinProgUniversAvtomatMU_Tab

Линпрогооптимизатор LinProgUniversAvtomatMU_Tab настраивается всего лишь один раз перед первым запуском программы при формировании шаблонов таблиц ItogTabLPrmi.rpt/ ItogTabLPrma.rpt выходных объектов x , y , ef , ng и gv . В дальнейшем, при вычислении линпрогооптимумов необходимые настройки выполняются на табличном команд-указании LinProgUnivUkz_Tab рис. 2.12.

А. Формирование шаблонов итог-таблиц и настройка Репорт Эксплорера на выдачу итог-таблиц в html-формате производится согласно разд. 2.4.4.2. При этом можно использовать аргументы-условия, приведенные в линпрогуказании рис. 2.12, или заменить их данными Исполнителя.

Б. Аргументы-условия (fo , Ao , bo , $Aeqo$, $beqo$, Lbo , Ubo) линпрогзадачи Исполнителя при её размерности до 50 следует записывать в строки 12...14, 16 и 18 (рис. 2.12, Б), не меняя имён переменных.

В. При размерности задачи до 100 и более эти аргументы целесообразно формировать в отдельном текст-файле с присвоением им тех же имён. Сформированные аргументы выделяются и отправляются в Вычпространство клавишей $f9$ ($fn+f9$ – для ноутбука) или контексткомандой «Evaluate Selection». Перед запуском линпрогуказания следует закомментировать строки с аргументами, сформированными в другом файле: функция `linprog(...)` будет брать значения аргументов из Вычпространства

Г. Если какая либо из пар Ao , bo и $Aeqo$, $beqo$ или обе одновременно не участвуют в оптимизации, строки с числовыми данными пар следует закомментировать, и раскомментировать строки 15 и 17.

Д. Если значения некоторых компонент в векторах нижних (Lbo) и верхних границ (Ubo) ц-области неизвестны, то в Lbo на место этих компонент ставится значение «-inf», а в Ubo – «inf».

Е. Если при выполнении запуска оптимизации итог-таблица не нужна, то перед запуском следует закомментировать строку 7 и раскомментировать строку 8.

Ж. Для вычисления линпрогмаксимума следует закомментировать строку 5 и раскомментировать строку 6.

З. Для изменения величин уставок базовой функции Матлаба `linprog(...)` следует раскомментировать строку 9 и ввести в команду `optimset(...)` необходимые уставки, руководствуясь таблицей уставок из разд. 2.7.2.

2.5. Коррекция задач линпрогооптимизации с использованием выходных объектов ng и gv

При непустых значениях этих объектов согласно командам в строках 33 и 38 линпрогооптимизатора LinProgUniversAvtomatMU_Tab (рис. 2.13, Б)

$$\left. \begin{aligned} nro &= [Ao * x_o, bo'], \\ rvo &= [Aeqo * x_o, beqo'], \end{aligned} \right\} \quad (2.16)$$

где x_o – оптимизационный вектор, вычисленный в результате решения задачи.

При удачном завершении оптимизации (флаг процесса $ef=1$) значения этих объектов могут выглядеть следующим образом (см. рис. 2.14):

efmi_16_039866 =			
1			
nrmi_16_039866 =		rvmi_16_039866 =	
-0.3797	6.0000	8.0000	8.0000
2.3000	2.3000	3.9000	3.9000
5.0156	17.0000	7.6200	7.6200
3.5000	3.5000		
4.4779	11.0000		

В этом случае 1-й объект характеризует степень удовлетворения ограничений-неравенств (правый столбец – вектор bmi_{16_039866}), а 2-й – констатирует удовлетворение ограничений-равенств (правый столбец – вектор $beqmi_{16_039866}$).

Если оптимизация оказалась неудачной при $ef=-2$, то по виду выходных объектов можно обнаружить причины неудачи и устранить их, скорректировав значения: векторов bo / $beqo$ (что наиболее просто) или ко-

эффицентов в «ошибочных» строках матриц A_0 / A_{eq} . Рассмотрим пример. На рис. 2.17, А показано линпрогуказание с настройкой на минимум, запуск которого закончился неудачей (рис. 2.17, Б): звёздочками отмечены строки, в которых не выполняются ограничения-неравенства (1-я строка в объекте $prmi$) и ограничения-равенства (все 3 строки объекта $gvmi$).

```

1 % Универсальное линпрогуказание LinProgUnivUkz_TabA
2 % Настройки универсального линпрогоптимизатора LinProgUnivesAvtomatMU_Tab
3 % NumLPmi=0; % Нумератор объектов линпрогмина очистить при необх. новой нумерации.После 1 пуска - закомментировать.
4 % NumLPma=0; % Нумератор объектов линпрогмакса очистить при необх. новой нумерации.После 1 пуска - закомментировать.
5 - Lmi_Lma=1; % Поиск линпрогмина.Закоммент. при поиске линпрогмакса.
6 % Lmi_Lma=-1; % Поиск линпрогмакса. Закоммент. при поиске линпрогмина.
7 - ItogTabLP=1; % Запуск выдачи итог-тблицы оптимизации - закомментировать, если итог-таблицы не нужны.
8 % ItogTabLP=-1; % Стирает в Вычпространстве 1, прекращая выдачу итог-таблиц оптимизатором LinProgUnivesAvtomatMU_Tab.
9 % options=optimset('param1',число,'param2','строка-значение',...); % Изменение настроек линпрогфункции Матлаба linprog(...):
10 % раскомментировать строку и заполнить шаблон optimset(...)
11 % Аргументы линпрогфункции Матлаба linprog(...)
12 - fo = [-5.6 4.4 6 -3 9];
13 - Ao = [1 -1 1 2 3; 3 2 4 1 -2; 3 2 0 4 3; 1 3 5 4 -2; 3 4 7 1 -2];bo = [4.5 2.3 17 6.5 11];
14 - Aeqo = [0 1.5 0 0 2; 3 1 2.5 0 0.7; 4.1 -2.3 11 3.4 4.7]; beqo = [7 2 3];
15 % % Ao=[]; bo=[];
16 % % Aeqo=[]; beqo=[];
17 - LBo = zeros(1,5);UBo = [6.5 4.5 5 8.2 3.1];
18 % Запуск универсального линпрогоптимизатора.
19 - LinProgUnivesAvtomatMU_Tab % Текст-файл табличного универсального линпрогоптимизатора.
20

```

Рис. 2.17. Линпрогуказание с неуспешным запуском по поиску линпрогминимума (А) и результат неуспешного запуска (Б). Продолжение рисунка – на следующей стр.

```

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

Exiting: One or more of the residuals, duality gap, or total relative error
has grown 100000 times greater than its minimum value so far:
the primal appears to be infeasible (and the dual unbounded).
(The dual residual < TolFun=1.00e-008.)

xmi_1_51418 =

    0.0000
    1.8248
    0.0000
    0.0000
    2.5091

ymi_1_51418_04_May_2016 =

    30.6111

efmi_1_51418 =

    -2

nrmi_1_51418 =
★ 5.7026  4.5000
  -1.3685  2.3000
  11.1769 17.0000
   0.4563  6.5000
   2.2811 11.0000

rvmi_1_51418 =
★ 7.7554  7.0000
★ 3.5812  2.0000
★ 7.5961  3.0000

```

Продолжение рис. 2.17. Результат неуспешного запуска линпрогуказания рис. 2.17, А. вёздочками отмечены строки, в которых не выполнены ограничения-неравенства (объект $prmi$) и ограничения-равенства (объект $gvmi$).

Откорректируем 1-ю компоненту вектора bo и все 3 компоненты вектолла $beqo$. Для компоненты $bo(1)$ возьмём значение, превышающее значение $prmi(1,1)$ на 0,01, т. Е. $bo(1)=5.71$. Для вектора $beqo=[7.75 \ 3.58 \ 7.6]$.

Результат запуска линпрогуказания рис. 2.17,А с откорректированными векторами bo и $beqo$ представлен на рис. 2.18.

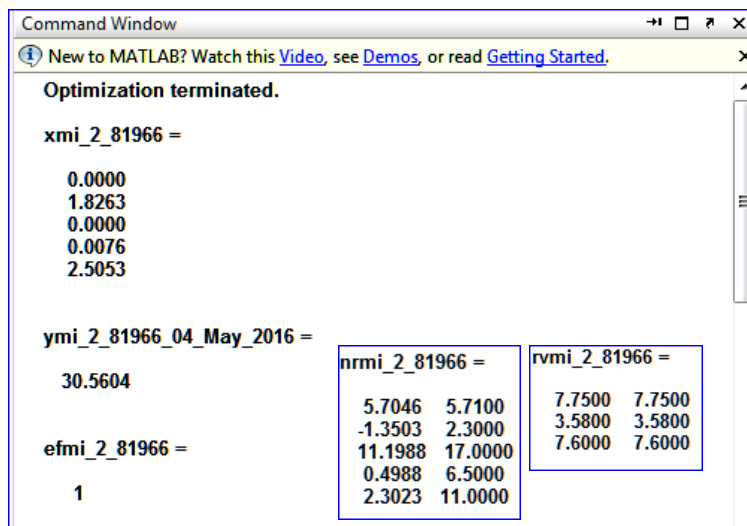


Рис. 2.18. Результат успешной линпрогминимизации по указанию рис. 2.17,А, в котором в качестве откорректированных векторов bo и $beqo$ использовались 2-ые столбцы объектов соответственно $nrmi$ и $rvmi$.

Линпрогминимизация оказалась успешной после 1-й итерации коррекции. При большой размерности задачи для успешной линпрогоптимизации возможно потребуется не одна, а несколько итераций коррекции.

2.6. Решение серии линпрогзадач в одной сессии Матлаба

Наличие в линпрогкомплексе «LinProgUnivUkz_Tab – LinProgUniversAvtomatMU_Tab» процесса создания в действующей папке в результате оптимизации пары файлов итогтаблицы и копии линпрогуказания со всеми условиями оптимизации позволяет реализовать следующую процедуру решения серии линпрогзадач в одной сессии Матлаба. Для размерностей задач в несколько десятков условия каждой задачи записываются в одно и тоже линпрогуказание LinProgUnivUkz_Tab с одними и теми же шаблонными именами переменных в отведенную для каждой задачи нумерованную рубрику. При этом, условия ранее решённых задач должны быть закомментированы. Как только задача решена (с учётом возможной коррекции условий согласно разд.2.5), ее условия комментируются, и начинается ввод условий следующей задачи. На рис. 2.19 показано линпрогуказание LinProgUnivUkz_Tab_SeriaZad с набором из 4 линпрогзадач, полученное из линпрогуказания

```

Editor - D:\SAnRazrab1_MU-Zad-Otch171007_020312_ProdolRab_2016\MetUkazSiAn_ML7-Rik41008_kor180211_2016\Rab4_091208_2016\LinProgUnivUkz_Tab_S...
File Edit Text Go Cell Tools Debug Desktop Window Help
tic
fo=[];Ao=[];bo=[];Aeqo=[];beqo=[];LBo=[];UBo=[]; %Очистка значений предыдущего запуска.
Lmi_Lma=1; % Поиск линпрогмина.Закоммент. при поиске линпрогмакса.
Lmi_Lma=-1; % Поиск линпрогмакса. Закоммент. при поиске линпрогмина.
ItogTabLPr=1; % Запуск выдачи итог-тпблицы оптимизации - закомментировать, если итог-таблицы не нужны.
% ItogTabLPr=-1; % Стирает в Вычпространстве 1, прекращая выдачу итог-таблиц оптимизатором LinProgUniversAvtomatMU_Tab.
% options=optimset('param1',число,'param2','строк-значение',...); % Изменение настроек линпрогфункции Матлаба linprog(...):
% раскомментировать строку и заполнить шаблон optimset(...).
ImjaKopirFailLPukaz='LinProgUnivUkz_Tab_SeriaZad.m'
% Аргументы линпрогфункции Матлаба linprog(...)
% Zad.1
fo=[-5.6 4.4 6 -3.9 10.2 1.7 -3.5 2.6 0.9 -4.7 1.17];
Ao=[1 -1 1 2 3 5 4.3 6.1 -3.8 0 0 0; 3 2 4 1 -2.9 4.1 2.3 0 0 0 0; 3 2 0 4 4.2 10 -6.5 0 0 0 0 0; 1 3 5 4 -2 0 0 0 0 0 0 0; 3 4 7 1 -2 0 0 -5.2 3.1 0 0 0];
bo=[6 2.3 17 3.5 11];
% Ao=[]; bo=[];
Aeqo=[0 1.5 0 0 2 5.7 8 -3 11.2 0 0 0; 3 1 2.5 0 0.7 0 0 0 5.8 -7.2 10 13; 4.1 -2.3 11 3.4 4.7 2.9 0 0 6.8 -9 0 0]; beqo = [8 3.9 7.62];
% Aeqo=[]; beqo=[];
LBo=zeros(1,12); UBo=[6.5 4.5 5 8.2 3.1 6.8 11 23 4.7 10 13 4.4];
% Zad.2
fo = [-5.4 -6 -3.9];
Ao = [1 -1 1 2 3; 3 2 4 1 -2; 3 2 0 4 3; 1 3 5 4 -2; 3 4 7 1 -2]; bo = [20 42 30 15 23];
Aeqo=[]; beqo=[];
LBo = zeros(1,5); UBo = [6.5 4.5 5 8.2 3.1];
% Zad.3
fo = [-5.3 2.2 -6 -3.9];
Ao = [1 -1 1 2 3; 3 2 4 1 -2; 3 2 0 4 3; 1 3 5 4 -2; 3 4 7 1 -2]; bo = [20 42 30 15 23];
  
```

```

29 % Aeql = [1.7 1.5 0 0 2; 3 1 4.2 0.8 0]; beql = [5.3 2.9];
30 % LBo=zeros(1,5); UBo=ones(1,5)*10;
31 % Zad.4
32 - n =70;n1=10;n2=5;Ar=3.*rand(n1,n2);A1o=zeros(n1,n-n2);Apr=catt(2,Ar,A1o);
33 - A2o=zeros(n-n1,n);Ao=catt(1,Apr,A2o);
34 - bo=[14.*rand(n1,1);zeros(n-n1,1)]; % Первые n1 компонент числа, равномерно
35 % распределенные в диапазоне 0...14; а остальные 60 - нули.
36 - fo=(-1).^fix(10.*rand(1,n)).*6.*rand(1,n); % равномерное распределение компонент
37 % в диапазоне -2...0 с умножением некоторых на -1, когда
38 % fix(10.*rand(1,n))=нечетному числу
39 - n3=30;n4=50;Ar1=4.*rand(n3,n4);A3o=zeros(n3,n-n4);Apr1=catt(2,Ar1,A3o); % Формирование матрицы Aeql.
40 - A4o=zeros(n-n3,n);Aeqo=catt(1,Apr1,A4o); % Формирование матрицы Aeql.
41 - beql=[10.*rand(n3,1);zeros(n-n3,1)]; % Первые n3 компонент числа, равномерно
42 % распределенные в диапазоне 0...10; а остальные 40 - нули.
43 - LBo=-2.*rand(1,n); % равномерное распределение компонент в диапазоне -2...0
44 - UBo=5.*rand(1,n); % равномерное распределение компонент в диапазоне 0...5
45
46 % Запуск универсального линпрогоптимизатора.
47 - LinProgUniversAvtomatMU_Tab % Текст-файл табличного универсального линпрогоптимизатора.
48 - toc
49

```

Рис. 2.19. Линпрогуказание LinProgUnivUkz_Tab_SeriaZad.m с серией решённых 4 линпрогзадач. На рис. 2.20 показаны в действующей папке пары файлов итогтаблицы и линпрогуказания для каждого задания.

LinProgUnivUkz_Tab путём изменения его имени, вставки условий задач и вставки изменённого имени в команду строки 11. После решения каждой из задач в действующей папке возникает пара файлов (итогтаблицы и копии линпрогуказания с именем файла итогтаблицы) – все 4 пары показаны на снимке фрагмента действующей папки на рис. 2.20. Фрагмент итогтаблицы задания 4 показан на рис. 2.21.

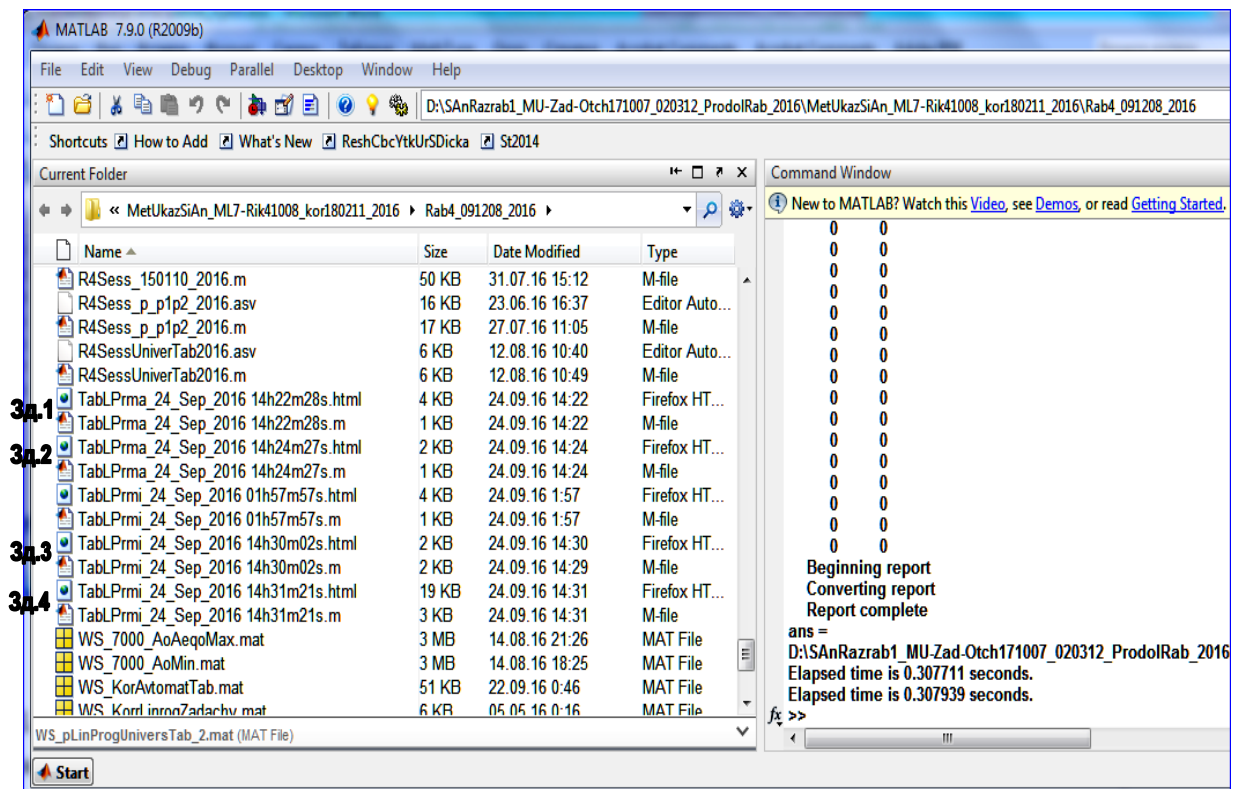


Рис. 2.20. Фрагмент действующей папки с файлами итогтаблицы и копии линпрогуказания для каждой задачи из рис. 2.19.

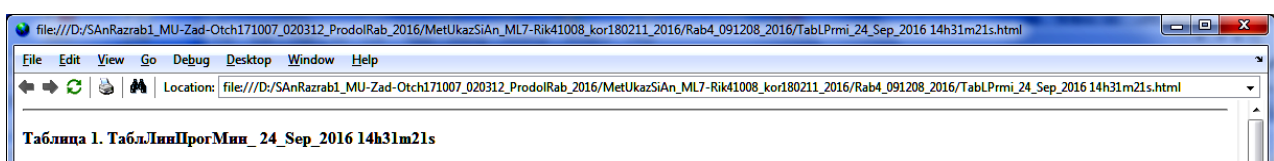


Рис. 2.21. Фрагмент итогтаблицы задачи 4 из линпрогуказания рис. 2.19: показаны 20 строк из 70.

2.7. Правила и этапы решения непараметрических задач линейного программирования посредством линпрогооптимизаторов с функцией linprog

2.7.1. Правила и этапы

1. Подготовка вычислительной среды для решения задач линпрогооптимизации

А) Имена папок, подпапок, файлов, переменных должны выполняться английским шрифтом. Наличие русских букв в директории файла приводит к неузнаванию его Матлабом и к отказу открытия файла.

Б) Выполнение инженерных или учебных расчётов с применением функции linprog целесообразно проводить в отдельной папке, в которой будут храниться файлы, обеспечивающие вычислительную среду расчётов и вывода их результатов. Пусть работа по линпрогооптимизации будет иметь 4-й номер. Предполагаем также, что выполняется учебное задание разд. 4 для одного из вариантов исходных данных разд. 5.

Создайте папку «Rab4Familija(vasha)», в которую следует помещать все файлы, связанные с выполнением Задания 4 по линпрогооптимизации. Создайте в папке Rab4 сессионный файл (сессфайл) «R4SessFamilija(vasha)N», мат-файл «WS_R4Familija(vasha)N.mat».

Для этого откройте Матлаб. В поле «Current Directory» вставьте полное имя папки Rab4 (путь к ней), используя кнопку поиска папки . Создайте объекты среды для выполнения работы 4.

Файл «WS_R4Familija(vasha)N.mat» (файл Вычпространства работы 4): после открытия папки Rab4 откройте меню «Desktop» и поставьте галочку в строке «Workspace». В открывшемся окне Вычпространства (Workspace) командами меню «File_Save Workspace As» сохраните пустой файл с указанным именем. В нём сохраняются имена и значения всех вычислительных объектов расчёта, за исключением файлов.

Сессионный файл «R4SessFamilija(vasha)N» (N – номер варианта задания) путем открытия нового м-файла в окне Редактора (Editor) и сохранения его в папке Rab4 командами File-Save_As. В этом файле набираются и сохраняются данные, командования, программы, результаты вычислений (копируемые из Командного окна), комментарии и другие рабочие материалы.

2. Математическое описание задач линпрогооптимизации

В) Математическое описание линпрогзадачи (целевую функцию, ограничения-неравенства и ограничения-равенства, ц-область изменения оптимизирующего вектора x) представляется в виде выражений типа (2.8)...(2.11) с отражением в них специфики задачи. Такая запись помогает уяснить сущность задачи оптимизации и фиксирует её трактовку Исполнителем. Дело в том, что для успешного решения конкретной задачи может потребоваться изменить неравенства на равенства или часть неравенств включить в число равенств (и наоборот), входя в кажущееся противоречие с исходной постановкой линпрогзадачи. Фиксация в тексте этапов таких манипуляций с данными позволяет избежать случайных ошибок и ошибок неправильного распоряжения данными.

Выражения записываются в сессфайле в виде комментариев (т.е. в строках с начальным знаком проценты (%)) в рубрике с заголовком типа «Постановка задачи/задач Линпрогооптимизации». Если задач несколько, то их описания целесообразно пронумеровать, чтобы присваивать каждый номер всем переменным, задействованным в решении задачи данного номера.

Для учебного примера разд. 4 эту запись производить не нужно.

3. Формирование линпрогуказания для решения задач

Г) При формировании Линпрогуказания целесообразно использовать его шаблоны из разд. 2.4. Для 1...2 задач – простые шаблоны рис. 2.3. Для нескольких задач – шаблон универсального линпрогуказания LinProgUnivUkz совместно с универсальным линпрогооптимизатором LinProgUniversAvtomatMU из разд. 2.4.2. Для множества последовательно решаемых задач – шаблон универсального табличного линпрогуказания LinProgUnivUkz_Tab совместно с универсальным табличным линпрогооптимизатором LinProgUniversAvtomatMU_Tab из разд. 2.4.4. Все виды шаблонов и их линпрогооптимизаторы можно копировать из электронной версии Пособия, вставлять в отдельные текстовые М-файлы и использовать как рабочие инструменты линпрогооптимизации. Ниже приведены дополнительные пояснения по выбору вариантов.

Рекомендуемый вариант линпрогуказания для решения всех упомянутых заданий – последний. Он обеспечивает выходным данным запущенной задачи (оптимвектору, значению ц-функции, флагу оптимизации (ef) и выходным объектам (pr – неравенствам и tv – равенствам) присвоение уникальных (практически не

повторяющихся) идентификаторов и формирование из этих данных таблицы в html-формате, в заголовке которой указываются все аргументы-условия, задействованные в задаче.

3.1. Простой шаблон линпрогуказания рис. 2.3

Д) В сессфайл выполняемой Работы 4 набирается линпрогуказание в нумерованных строках (или копируются эти строки из электронной версии Монографии) типа рис. 2.3, А (для поиска минимума) или рис. 2.3, Б (для максимума) с коррекцией числовых данных по виду выражений оптимизируемой функции $f(x)$ типа (2.8), ограничений-неравенств (2.9), ограничений-равенств (2.10) и границ α -области (2.11). Назначение объектов. Векторы: f (коэффициентов целевой функции), b (правых частей ограничений-неравенств), beq (правых частей ограничений-равенств), lb (нижних пределов α -области для компонентов вектора x) и ub . Матрицы A , Aeq (см. рис. 2.3) составлены из коэффициентов левых частей неравенств и равенств типа соответственно (2.9) и (2.10). Каждый объект оканчивается «точкой с запятой» (;) для предотвращения его выдачи в Командное окно. Объекты в последних 3 строках обоих линпрогуказаний должны выводиться в Комокно для копирования в сессфайл, и точка с запятой в этих строках не ставится.

Е) Векторы нижних (lb) и верхних (ub) границ могут вводиться в виде выражений соответственно $[nz1\ nz2\ \dots\ nzn]$ и $[vz1\ vz2\ \dots\ vzn]$ (n – размерность задачи) или с использованием операторов $zeros(1,n)$ (в случае нулевых значений нижних границ) или $vz*ones(1,n)$ (в случае равенства всех верхних границ величине vz).

Если какая-либо система ограничений ($A\&b$ или $Aeq\&beq$) не задействуется в поиске оптимума, то её переменные приравниваются знаку $[]$ (например, $A=[]$, $b=[]$).

Если значения некоторых компонент в векторах нижних (lb) и верхних границ (ub) α -области неизвестны, то в lb на место этих компонент ставится значение « $-\inf$ », а в ub – « $\inf$ ».

Ж) Если шаблон применяется для решения нескольких задач оптимизации с одним и тем же Вычпро-странством, то для предотвращения стирания текущим результатом предыдущего следует нумеровать все переменные линпрогуказания (как показано на рис. 2.3).

З) При поиске максимума перед вектором f в аргументном списке оптимизатора ставится минус – рис. 2.3 Б.

И) Для задач минимизации искомые вектор и функцию обозначают как xm (xmi) и um (umi), а для максимизации – xma и uma , причём в оптимизатор вместо uma вставляют ur (u промежуточное) и к оптимизатору добавляют: присваивание $uma = -ur$. В оба вида оптимизатора добавляются также объекты: $[A*хорт,b']$, $[Aeq*хорт,beq']$ (по отдельности или оба вместе – в зависимости от вида оптимизатора см. рис. 2.2 В, Г и рис. 2.3, Б), которые используются в исследовательских целях и для коррекции задачи при неуспешном решении – см. разд. 2.5.

3.2. Шаблон универсального линпрогуказания LinProgUnivUkz (рис. 2.4)

К) Работает совместно с линпрогооптимизатором LinProgUniversAvtomatMU (рис. 2.5). Это линпрогуказание обладает параметрическим выбором типа оптимизации (мин/макс), с возможностью раздельной (для мин/макс) нумерации и уникальной идентификации всех объектов и выходных переменных и с выдачей в Комокно значений текущей линпрогооптимизации: нумерованных и идентифицированных выходных переменных x,u,ef и выходных объектов pg и gv .

Описание оптимизатора – в разд. 2.4.2. При решении нескольких задач оптимизации не нужно заботиться о нумерации объектов задачи: линпрогооптимизатор LinProgUniversAvtomatMU обеспечивает это автоматически.

Л) Если есть возможность, тексты линпрогуказания LinProgUnivUkz и линпрогооптимизатора LinProgUniversAvtomatMU следует скопировать из электронной версии Пособия.

М) Линпрогуказание LinProgUnivUkz и линпрогооптимизатор LinProgUniversAvtomatMU следует помещать в отдельных текст-файлах с приведенными именами (с возможным добавлением к ним фамилии Исполнителя на английском).

Если решаемые задачи небольшой размерности, то их условия ($f_0, A_0, b_0, Aeq_0, beq_0, LBo, UBo$) можно записать подряд, не меняя имён переменных, в одном линпрогуказании, и закомментировать все, кроме принадлежащих запускаемой задаче. Закомментировав условия решённой задачи и раскомментировав условия запускаемой переходят к решению следующей задачи. Процесс повторяется до решения всех задач.

3.3. Шаблон универсального табличного линпрогуказания LinProgUnivUkz_Tab (рис. 2.12)

Н) Работает совместно с линпрогооптимизатором LinProgUniversAvtomatMU_Tab (рис. 2.13, А). Это линпрогуказание, как и предыдущее, обладает параметрическим выбором типа оптимизации (мин/макс), с возможностью раздельной (для мин/макс) нумерации и уникальной идентификации всех объектов и выходных переменных, а также параметрическим управлением выдачи (выдавать/ не выдавать) итогтаблицы оптимизации. Оно выдаёт в Комокно результаты выполненной линпрогооптимизации: нумерованных и уникально идентифицированных выходных переменных x, y, ef и выходных объектов pg и gv . Выдаваемая линпрогуказанием в действующую папку итогтаблица в имени переменной её файла имеет: тип оптимизации (мин/ макс), дату и время запуска оптимизации, порядковый номер и уникальный идентификатор, присвоенный всем объектам этой оптимизации – см. рис. 2.15, Б. В заголовке итогтаблицы помещён перечень имён всех аргументов-условий (для функции $linprog(\dots)$), задействованных в данной оптимизации – рис. 2.15, А. При последующем анализе результатов решения ряда оптимизационных задач перечень имён может быть скопирован в Комокно или в Редактор, и все эти условия (по частям или целиком) могут быть вызваны в Комокно для исследования.

О) Если есть возможность, тексты линпрогуказания LinProgUnivUkz_Tab и линпрогооптимизатора LinProgUniversAvtomatMU_Tab следует скопировать из электронной версии Монографии.

П) Для того, чтобы линпрогуказание выдавало итогтаблицу, необходимо настроить её шаблоны (для мин/ макс). Методика настройки – в разд. 2.4.4.2 «Формирование шаблонов итог-таблиц ItogTabLPrmi.rpt/ ItogTabLPrma.rpt/».

Р) Решение задач большой размерности с громоздкими для ручного ввода аргументами-условиями. Эти условия следует набирать в отдельном текст-файле с присвоением аргументам шаблонных имён: $f_0, A_0, b_0, A_{eq}, b_{eq}, LBo, UBo$. В этом файле не должно быть других объектов. Файл запускается кнопкой «Run»  перед запуском линпрогуказания.

4. Запуск линпрогуказания

С) Если текст линпрогуказания последних 2 типов (и особенно, линпрогооптимизаторов) набирался вручную (не был скопирован с электронной версии Монографии), то его следует проверить на наличие в нем жёлтых меток, посредством которых Матлаб сообщает совет или замечание об ошибке для данного места текста (при подводе к метке мышки). Следует проверить все метки на замечание об ошибке, и устранить ошибки. кроме того, полезно проверить правильность работы линпрогуказания, подставив в него данные из пособия, и получить успешное решение ($efmi/ efma=1$).

Т) Случаай простого линпрогуказания (пп. Д)...И)). Оно выделяется и после нажатия клавиши f_9 ($f_n + f_9$ – в ноутбуке) Матлаб создаст в Вычпространстве (в окне Workspace) вычислительные объекты согласно их определениям из указания и выдаст в Командное окно искомое решение (x, y, ef, pg и gv) или сообщение об ошибке. При наличии ошибок в линпрогуказании. Ошибки следует найти и устранить.

У) Случаай универсальных линпрогуказаний (пп.К)...Р)). Каждое указание запускается кнопкой «Run» .

Ф) После правильного вычисления оптимизации следует перейти в окно вычобъектов и сохранить его содержание в мат-файле с именем «R?FamilijaPolzovatelja.mat» (R? – номер выполняемой работы).

Х) Команда вызова управляющих уставок и их номинальные значения (в графе 3 табл. 2.1):

$$\text{options}=\text{optimset}('linprog') \quad (2.17)$$

2.7.2. Уставки оптимизатора linprog и значения переменной exitflag (ef)

Таблица 2.1

Имя параметра	Описание	Значения уставок, в скобках { } – по умолчанию	Примечания
1. LargeScale	Применяется крупно-размерностный (КР) алгоритм поиска оптимума.	{ 'on' } 'off'	При значении 'off' работает средне-размерностный ПКвП-алгоритм, (SQP-алгоритм)
2. Diagnostics	Выводит диагностическую информацию о ц-функции.	{ 'off' } 'on'	

Имя параметра	Описание	Значения уставок, в скобках { } – по умолчанию	Примечания
3.Display	Значения: 'off' - нет выходных данных; 'iter'- выходные данные по каждой итерации (только для КР- и Симплекс-алгоритма в Матлабе R2009b) ; 'final' - данные по результатам вычислений; 'notify' выдает информацию, если решение не найдено.	{ 'final' } 'off' 'iter' 'notify'	
4.MaxIter	Максимально-допустимое число итераций.	{85} Положительное целое число	85 – для КР-алгоритма; 10*(длина оптимвектора) – для Симплекс-алгоритма; 10*max(длина оптимвектора, фактич. число строк в матрице A+ суммарное число нижних и верхних границ в векторах lb и ub, т.е. без +- inf) – для ПКВП-алгоритма
5. Simplex	Симплекс-алгоритм	{ 'off' } 'on'	Для включения Симплекс алгоритма: options = optimset('LargeScale','off', 'Simplex', 'on')
6.TolFun	Погрешность окончания по приращению величины целевой функции.	{10 ⁻⁸ } Положительное дробное число	10 ⁻⁸ – для КР-алгоритма; 10 ⁻⁶ – для Симплекс-алгоритма; [] – не установлено, для ПКВП-алгоритма.

Ц) Значения выходной переменной exitflag (efm, efma – при сокращении её имени для поиска минимума и максимума соответственно).

- 1 – минимизирующее решение вычислено.
- 0 – число итераций превысило уставку MaxIter.
- 2 – решение не найдено.
- 3 – не введены границы для вектора x.
- 4 – в расчетах появилось значение NaN (Not a Number, не число).
- 5 – прямая и двойственная задачи являются недопустимыми.
- 7 – поисковый шаг слишком мал, дальнейший поиск оптимума невозможен.

3. Зависимость линпрогооптимума от параметра p1. Однопараметрические линпрогооптимизаторы: с ручной настройкой и универсальный.

В оптимизаторе linprog нет варианта, обеспечивающего поиск оптимума в зависимости от параметров. Параметры могут входить в состав оптимизатора как элементы его аргументов, и оптимизатор правильно воспринимает параметр только как скаляр, что позволяет вручную исследовать зависимость оптимума от величины параметра. Для ввода параметра как вектора и автоматического расчета зависимости необходимо использовать цикл for. Линпрогооптимизацию при наличии параметра p будем именовать p-линпрогооптимизацией.

Будем рассматривать 2 вида p-линпрогооптимизаторов: с ручной настройкой (ручной) и универсальный. Ручной оптимизатор может быть двух типов: p-линпрогоминимизатор и p-линпрогомаксимизатор. Имена переменных, сохраняемых в Вычпространстве, в каждом из двух p-оптимизаторов свои для предотвращения замещения одних другими при запусках в одной Матлаб-сессии разных p-оптимизаторов.

В универсальном p-оптимизаторе имена переменных носят шаблонный вид, не отражающий тип оптимизации. В Вычпространство эти переменные копируются напрямую, а выходные переменные функции linprog(...), параметр и отладочные (выходные) объекты rpl/ rpv копируются вторично с именами, отражающим тип оптимизации. В этом оптимизаторе одни и те же условия можно использовать для вычисления p-минимума и p-максимума путём переключения одной командой-переключателем. При производстве ряда (или массовых) p-линпрогооптимизаций преимущество универсального перед ручным в отношении трудоёмкости многократно возрастает.

3.1. Ручной однопараметрический оптимизатор pLinProgOptRuch.

Пример использования цикла for для исследования зависимости минимума fcp1(x) (3.1) от параметра P1 при наличии ограничений неравенств (3.2) и равенств (3.3) представлен на рис. 3.1 – оптимизатор выделен заливкой, ниже следует выданный Матлабом результат p-линпрогооптимизации.

$$f_{cp1} = -5x_1 + 4P_1x_2 + 6,3x_3 - 3x_4 + 9x_5, \quad P_1 = 0.5:1.4:7.5. \quad (3.1)$$

При следующих ограничениях.

$$\left. \begin{aligned} x_1 - 1 \cdot P_1(k)x_2 + x_3 + 2x_4 + 3x_5 &\leq 4.5, \\ 3x_1 + 2x_2 + 4x_3 + x_4 - 2x_5 &\leq 2.3, \\ 3x_1 + 2x_2 + 4x_4 + 3x_5 &\leq 17, \\ x_1 + 3x_2 + 5x_3 + 4x_4 - 2x_5 &\leq 6.5, \\ 3x_1 + 4x_2 + 7x_3 + x_4 - 2x_5 &\leq 11. \end{aligned} \right\} A * x \leq b, \quad x = [x_1 \ x_2 \ x_3 \ x_4 \ x_5]. \quad (3.2)$$

$$\left. \begin{aligned} 1.5x_2 + 2x_5 &= 7.33, \\ 3x_1 + x_2 &= 2.38. \end{aligned} \right\} A_{eq} * x = b_{eq}, \quad (3.3)$$

pLinProgOptRuch вычисляет зависимость оптимума линейной ц-функции от параметра P1. На рис. 3.1 представлен вариант программы, вычисляющий зависимость минимума линейной цф от P1. Для того, чтобы программа рассчитывала зависимость максимума цф (МаксЦФ(P1)), необходимо выполнить в ней следующие изменения.

- В строке 10: заменить f на -f, а также - xm, um, efм соответственно на xma, up, efma.
- Строку 11 записать в виде: xma1(:,k)=xma; ymi1(k)=-up; efma1(k)=efma;
- В элементах строк 13 и 15 букву m следует заменить сочетанием ma.

Если этот линпрогооптимизатор использовать для решения нескольких задач с сохранением результатов и условий-аргументов в одной действующей папке и в одном Вычпространстве (workspace), то все переменные нужно нумеровать отдельно для минимума и максимума. Чтобы не перепутать нумерацию между типами оптимизации и внутри одного типа целесообразно ввести два отдельных оптимизатора: для минимума (рис. 3.1) и для максимума (рис. 3. 2).

```
% Однопараметрический линпрогоминимизатор с ручными настройками pLinProgMinRuch.m
P1=0.2:1.75:5.45;
pxmi1=[];pymi1=[];pefmi1=[];pnrmi1=[];prvmi1=[];% Очистка расчёта предыдущего запуска.
pLmi_pLma=1; % Программный идентификатор запуска линпрогоминимизатора pLinProgMinRuch.
```

```

tic
for k=1:length(P1);
    pf = [-5 4.*P1(k) 6.3 -3 9];
    pA = [1 -1.*P1(k) 1 2 3; 3 2 4 1 -2; 3 2 0 4 3;...
          1 3 5 4 -2; 3 4 7 1 -2 ];
    pb = [4.5 2.3 17 6.5 11];
%    pA=[]; pb=[]; % Раскомментировать, если A, b не используются - их след. закомментировать.
    pAeq = [0 1.5 0 0 2; 3 1 0 0 0];
    pbeq = [7.33 2.38];
%    pAeq=[]; pbeq=[]; % Раскомментировать, если Aeq, beq не используются - их след. закомментиро-
%    вать.
    plb = zeros(1,5); pub = [6.5 6.5 6.5 6.5 6.5];
    [xo,yo,efo] = linprog(pf,pA,pb,pAeq,pbeq,plb,pub); % оптимизатор linprog
    pxmi1(:,k)=xo; pyim1(k)=yo; pefmi1(k)=efo; % Формирование цикловых элементов массивов pxmi1,
    pyim1, pefmi1.
    % Формирование цикловых элементов отладочных объектов pnrmi1 и prvmi1.
    if isempty(pA) % Проверка пустого значения матрицы pAo.
        pnrmi1=[];
    else
        pnrmi1(:,k)=[pA*xo, pb'];
    end
    if isempty(pAeq) % Проверка пустого значения матрицы Aeq.
        prvmi1=[];
    else
        prvmi1(:,k)=[pAeq*xo, pbeq'];
    end
end
end
P1,pxmi1,pyim1,pefmi1 % Строка вывода массивов результата.
%    pnrmi1,prvmi1 % Учебная строка - закомментировать при работе линпрогоптимизатора.
%    pTabLinProgRuchOtladHtm % Построитель отладочной html-таблицы с объектами pnrmi1, prvmi1.
%    pLinProgRuchP1OptTabFormer % Универс. формирователь html-итогтаблиц линпрогоптимизат. и
%    копии p-линпрогминимизатора с именем файла итогтаблицы
Tvich= toc

```

Optimization terminated.	pnrmi1(:,1) =	pnrmi1(:,4) =	prvmi1(:,1) =
P1 =	4.5000 4.5000	4.5000 4.5000	7.3300 7.3300
0.5000 1.9000 3.3000 4.7000 6.1000 7.5000	1.0250 2.3000	-2.4483 2.3000	2.3800 2.3800
pxmi1 =	10.5000 17.0000	12.1241 17.0000	prvmi1(:,2) =
0.0000 0.0925 0.2164 0.4598 0.5136 0.5524	3.4800 6.5000	-2.3671 6.5000	7.3300 7.3300
2.3800 2.1025 1.7307 1.0007 0.8393 0.7228	5.7850 11.0000	-0.4469 11.0000	2.3800 2.3800
0.0000 0.0000 0.0000 0.0000 0.0000 0.0000	pnrmi1(:,2) =	pnrmi1(:,5) =	prvmi1(:,3) =
0.0250 1.0690 1.4471 0.0000 0.0000 0.0000	4.5000 4.5000	4.5000 4.5000	7.3300 7.3300
1.8800 2.0881 2.3669 2.9145 3.0355 3.1229	1.3754 2.3000	-2.8516 2.3000	2.3800 2.3800
pyim1 =	15.0229 17.0000	12.3258 17.0000	prvmi1(:,4) =
21.6050 31.1027 38.7249 42.7444 45.2317 47.0283	6.5000 6.5000	-3.0394 6.5000	7.3300 7.3300
pefmi1 =	5.5804 11.0000	-1.1729 11.0000	2.3800 2.3800
1 1 1 1 1 1	pnrmi1(:,3) =	pnrmi1(:,6) =	prvmi1(:,5) =
	4.5000 4.5000	4.5000 4.5000	7.3300 7.3300
	0.8240 2.3000	-3.1430 2.3000	2.3800 2.3800
	17.0000 17.0000	12.4715 17.0000	prvmi1(:,6) =
	6.4632 6.5000	-3.5250 6.5000	7.3300 7.3300
	4.2855 11.0000	-1.6974 11.0000	2.3800 2.3800
			Tvich =
			0.0555

Рис. 3.1. Линпрогминимизатор pLinProgMinRuch.m (закрашенные строки) и его результат при включённой учебной строке: вычисление зависимости минимума ц-функции (3.1, в программе - `pyim1`) от параметра P1. Tvich=0.06 сек. График зависимости – на рис. 3.4.

```

%    Однопараметрический линпрогмаксимизатор с ручными настройками
%    pLinProgMaxRuch.m
tic
pxma2=[]; pyma2=[]; pefma2=[]; pnrma2=[]; prvma2=[]; % Очистка расчёта предыдущего запуска.
pLmi_pLma=-1; % Программный идентификатор запуска линпрогмаксимизатора pLinProgMaxRuch.
P1ma = 0.2:1:6.2;
for k=1:length(P1ma);
    pf2 = [6.7 -2.1 -0.3*p1a(k) 2 6.9 4.2 3.9];
    pA2 = [ 2.3 0.1 5.5 11.1 2.4 8.2 2.8;
           1.8 -0.1 11.2 -0.6 2.6 7.6 4.9;
           1.8*p1a(k) -1.5 10.8 2.4 6.6 5.7 5.4;
           2.6 -1.1 7.9 4.9 11.2 1.1 3;
           2.8 1 4.5 7.4 3.6 2.5 3.8];
    pb2 = [1.6 3.7 4.3 9.7 1];

```

```

% pA2=[]; pb2=[]; % Раскомментировать, если pA2, pb2 не используются - их след. закомментиро-
вать.
pAeq2 = [ 6.3      6 2.8 -5.4 5 5.9 1.5;
          3.7 1.7 1.6 -6.8 6.4 -0.3*p1a(k) 0.8;
          2 3.1 -2.5 -5.2 0.8 4.5 7 ];
pbeq2 = [4.2 3.5 9.2];
% pAeq2=[]; pbeq2=[]; % Раскомментировать, если pAeq2, pbeq2 не используются - их след. за-
комментировать.
plb2=zeros(1,7);pub2 = [5.6; 3.5; 8.2; 4.8; 6.9; 3.2; 4.4];
[xo, yo,efo] = linprog(-pf2,pA2,pb2,pAeq2,pbeq2,plb2,pub2); % Оптимизатор linprog
pxma2(:,k)=xo; pyma2(k)=yo; pefma2(k)=efo; % Цикловые элементы массивов xma2, yma2, efma2.
% Формирование цикловых элементов объектов rprma2 и prvma2.
if isempty(pA2) % Проверка пустого значения матрицы pA2.
    pnrma2=[];
else
    pnrma2(:,k)=[pA2*xo, pb2'];
end
if isempty(pAeq2) % Проверка пустого значения матрицы Aeq2.
    prvma2=[];
else
    prvma2(:,k)=[pAeq2*xo, pbeq2'];
end
end
P1ma,pxma2, pyma2,pefma2
pnrma2, prvma2 % Учебная строка - закомментировать при работе линпрогоптимизатора.
% pTabLinProgRuchOtladHtm % Построитель отладочной html-таблицы с объектами rprma2, prvma2.
% pLinProgRuchP1OptTabFormer % Универс. формирователь html-итогаблиц линпрогоптимизат. и
копии р-линпрогминимизатора с именем файла итогтаблицы.
Tvich_ma= toc

```

Exiting: One or more of the residuals, duality gap, or total relative error has stalled:

the primal appears to be infeasible (and the dual unbounded).
(The dual residual < TolFun=1.00e-008.)

P1ma =

0.2000	1.2000	2.2000	3.2000	4.2000	5.2000	6.2000
--------	--------	--------	--------	--------	--------	--------

pxma2 =

0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.8670	0.8711	0.4700	0.8793	0.8833	0.8871	0.4738
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.4212	0.4078	0.4675	0.3817	0.3690	0.3566	0.4573
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
1.1863	1.1831	0.0006	1.1768	1.1738	1.1707	0.0000

pyma2 =

5.7122	5.5983	2.2412	5.3767	5.2691	5.1633	2.1601
--------	--------	--------	--------	--------	--------	--------

pefma2 =

-2	-2	-2	-2	-2	-2
----	----	----	----	----	----

rvma2(:,1) =

9.0871	4.2000
5.1185	3.5000
11.3287	9.2000

rvma2(:,2) =

9.0404	4.2000
5.0372	3.5000
11.3084	9.2000

rvma2(:,3) =

5.1584	4.2000
3.7913	3.5000
1.8353	9.2000

rvma2(:,4) =

8.9494	4.2000
4.8791	3.5000
11.2686	9.2000

rvma2(:,5) =

8.9053	4.2000
4.8023	3.5000
11.2496	9.2000

rvma2(:,6) =

8.8619	4.2000
4.7269	3.5000
11.2306	9.2000

rvma2(:,7) =

5.1290	4.2000
3.7319	3.5000
1.8346	9.2000

Tvich_ma =

0.0712

Б

Рис. 3.2. Линпрогминимизатор pLinProgMaxRuch.m (закрашенные строки, А) и его результат при включённой учебной строке: вычисление зависимости максимума ц-функции (3.1, в программе – pyma2) от параметра P1 – на рис. 3.2, Б. Tvich_ma=0.071 сек. Правильное решение не найдено ни при одном значении параметра P1, о чём свидетельствует значение флага оптимизации pefma2=-2.

Содержание и смысл команд ряда строк линпрогоптимизатора pLinProgMinRuch

Строка «Очистка расчёта...» необходима для правильной работы оптимизатора при переходе к параметру с меньшей, чем в предыдущем запуске размерностью – см. раздел 3.2.

Строка «Программный идентификатор ...» необходима для формирования отладочной и итогтаблицы

соответственно в файлах `pTabLinProgRuchOtladHtm` и `pLinProgRuchP1OptTabFormer`. В зависимости от значения идентификатора (1/-1) формируется имя файла и заголовок таблицы (для мин/ для макс).

Строку с командами `A2=[]` и `b2=[]` следует раскомментировать в задачах с отсутствием ограничений-неравенств. Одновременно следует закомментировать строки с числовыми значениями матрицы `A2` и вектора `b2`.

Строку с командами `Aeq2=[]` и `beq2=[]` следует раскомментировать в задачах с отсутствием ограничений равенств. Одновременно следует закомментировать строки с числовыми значениями матрицы `Aeq2` и вектора `beq2`.

Строка «Цикловые элементы выходных объектов...» необходима для формирования отладочной таблицы линпрогзапуска – файла `pTabLinProgOtladHtm` (см. разд. 3.4).

Строка «Учебная строка – закомментировать...» используется для пояснения структуры выходных объектов `pgm/ pgma` и `gvm/ gvma`. При использовании линпрогооптимизатора её следует закомментировать (чтобы не загружала Комокно лишними данными) или удалить. В объекте `pnpm` (рис. 3.2) для каждого значения параметра `P1` генерируется два столбца: левый – произведение `pA2*хо` (численные значения правых частей неравенств (3.2), правый – вектор `pb2'` (значения свободных членов из (3.2)). Структура объекта `prvm` аналогична частям ограничений-равенств в (3.3).

Если все элементы вектора `efmi` положительны (например, как на рис. 3.3, Б), то выходные объекты показывают степень удовлетворения неравенств и равенств. Если же некоторые (или все) элементы вектора `efm` неположительны (как на рис. 3.2,Б), то это означает, что линпрогооптимизатор не нашёл решение, а выходные объекты указывают строки, в которых не выполняются неравенства (3.2) и равенства (3.3). Для рис. 3.2,Б все строки равенства (3.3) не выполняются, а для ограничений-неравенств не выполняются большинство строчек, но не все.

Строка «Построитель отладочной html-таблицы...» раскомментируется при наличии в результате значений флага `ref`, не равных 1. Она формирует отладочную таблицу, включающую значения параметра `P1`, ц-функции, флага оптимизации и выходных объектов `pnpm/ pngma` и `prvm/ prvma` – см. рис. 3.2, А. Отладка заключается в изменении значений компонент векторов `pb2` и `pbeq2` на обеспечивающие замену неположительных флагов оптимизации `ref` на положительные, равные 1 – см. разд. 3.4.

Строка «Универсальный построитель ...» раскомментируется после завершения отладки задачи. При этом строка отладки должна быть закомментирована, чтобы не выдавать лишний объект оптимизации.

3.2. Необходимость вставки строки «Очистка предыдущего результата» в однопараметрические линпрогооптимизаторы

Эта строка, подобна строке 2 оптимизатора `pLinProgOptRuch.m` рис. 3.1. Предназначена такая строка для нейтрализации следующего недостатка параметрических программ Матлаба.

При однотипных параметрических вычислениях без изменения имён выходных переменных, когда размерность задачи изменяется от исходной величины к меньшей, Матлаб выдаёт результаты не с заменой всего результата предыдущего расчёта (как следовало бы сделать), а путём замены только части предыдущего результата, соответствующего по размерности новому результату. Остаток предыдущего результата остаётся нетронутым. Этот остаток делает новый результат ошибочным. Например, если произведены вычисления для 6 значений параметра, а потом для 4 с изменениями условий вычислений, Матлаб выдаст результат для 4 значений параметра в таком виде, как будто вычисления производились для 6 значений. Причём, первые 4 значений результата будут соответствовать новым вычислениям, а 2 оставшихся – предыдущим: см. рис. 3.2 и 3.3,Б.

Однопараметрический линпрогооптимизатор с ручными настройками `pLinProgMaxRuch.m`

```
tic
% pxma2=[];pyma2=[];pefma2=[];pnрма2=[];prvma2=[];% Очистка расчёта предыдущего запуска.
pLmi_pLma=-1; % Программный идентификатор запуска линпрогмаксимизатора pLinProgMaxRuch.
P1ma = 0.2:1:4.2;
for k=1:length(P1ma);
    pf2 = [6.7 -2.1 -0.3*p1a(k) 2 6.9 4.2 3.9];
    pA2= [ 2.3 0.1 5.5 11.1 2.4 8.2 2.8;
          1.8 -0.1 11.2 -0.6 2.6 7.6 4.9;
          1.8*p1a(k) -1.5 10.8 2.4 6.6 5.7 5.4;
          2.6 -1.1 7.9 4.9 11.2 1.1 3;
```

А

```

2.8 1 4.5 7.4 3.6 2.5 3.8];
pb2 = [ 4.8 7.3 9 9.7 6.9]; %; [ 4.8 7.3 9 9.7 6.9]; %; [ 4.8 7.3 8.8 9.7 6.9]; %; [ 4.6 7 8.1 9.7 7.1];
% [ 4.5 6.93 8.03 9.7 7.1]; %; [ 4.42 6.83 7.89 9.7 6.9]; %; [ 1.6 3.7 4.3 9.7 1];
% pA2=[]; pb2=[]; % Раскомментировать, если A2, b2 не используются - их след. закомментиро-
вать.
pAeq2 = [ 6.3 6 2.8 -5.4 5 5.9 1.5;
3.7 1.7 1.6 -6.8 6.4 -0.3*p1a(k) 0.8;
2 3.1 -2.5 -5.2 0.8 4.5 7 ];
pbeq2 =[4.2 3.5 9.2]; % [9.22 5.24 11.449]; % [9.1 5.2 11.4]; %; [4.2 3.5 9.2];
% pAeq2=[]; pbeq2=[]; % Раскомментировать, если Aeq2, beq2 не используются - их след. заком-
ментировать.
plb2=zeros(1,7);pub2 = [5.6; 3.5; 8.2; 4.8; 6.9; 3.2; 4.4];
[xo, yo,efo] = linprog(-pf2,pA2,pb2,pAeq2,pbeq2,plb2,pub2); % Оптимизатор linprog
pxma2(:,k)=xo; pyma2(k)=yo; pefma2(k)=efo; % Цикловые элементы массивов xma2, yma2, efma2.
% Формирование цикловых элементов объектов pnrma2 и prvma2.
if isempty(pA2) % Проверка пустого значения матрицы pA2.
    pnrma2=[];
else
    pnrma2(:,k)=[pA2*xo, pb2'];
end
if isempty(pAeq2) % Проверка пустого значения матрицы Aeq2.
    prvma2=[];
else
    prvma2(:,k)=[pAeq2*xo, pbeq2'];
end
end
P1ma,pxma2, pyma2,pefma2
% pnrma2, prvma2 % Учебная строка - закомментировать при работе линпрогооптимизатора.
pTabLinProgRuchOtladHtm % Построитель отладочной html-таблицы с объектами pnrma2, prvma2.
% pLinProgRuchP1OptTabFormer % Универс. формирователь html-итогаблиц линпрогооптимизат. и
копии p-линпрогминимизатора с именем файла итогаблицы.
Tvich_ma= toc

```

Optimization terminated.										Б
P1ma =	0.2000	1.2000	2.2000	3.2000	4.2000					
pxma2 =	0.1094	0.0052	0.0020	0.0013	0.0000	0.0000	0.0000			
	0.0000	0.0785	0.0808	0.0814	0.0815	0.8871	0.4738			
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000			
	0.0057	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000			
	0.3335	0.3686	0.3698	0.3701	0.3709	0.3566	0.4573			
	0.0000	0.0000	0.0000	0.0000	0.0005	0.0000	0.0000			
	1.2492	1.2359	1.2357	1.2356	1.2354	1.1707	0.0000			
pyma2 =	7.9179	7.2330	7.2146	7.2100	7.2084	5.1633	2.1601			
pefma2 =	1	1	1	1	1	-2	-2			

Рис. 3.3. Зависимость линпрогооптимума от параметра P1 с 5 значениями (на 2 меньше, чем в предыдущем – см. рис. 3.2), с изменением значений P1 и с закомментированной строкой 3. А – оптимизатор, Б – результат запуска оптимизатора. «Лишние» данные, совпадающие с данными рис. 3.2,Б, помечены розовым цветом.

Описываемая ситуация иллюстрируется рис. 3.2, Б и 3.3, Б. На рис. 3.3, Б Матлаб выдал результат как бы для 7 значений параметра P1 по всем выходным переменным rxma2, ruma2, refma2 несмотря на то, что в расчёте участвовало только 5 значений параметра. Из сравнения таких переменных, как rxma2, ruma2 и refma2 рис. 3.2 и 3.3, видно, что их первые 5 значения не совпадают, а последние 2 – совпадают. Таким образом, Матлаб к 5 значениям нового расчёта для каждой выходной переменной присоединил 2 значения прежнего расчёта, что сделало весь результат нового расчёта ошибочным. Снятие закомментированности 2-й строки в оптимизаторе рис. 3.3, А и повторение расчёта устранил это ошибочное действие Матлаба.

3.3. Построение зависимости $F_{\min}(P1)$

1) Исходный график строится командой из сессфайла выполняемой работы:

`plot(P1,румi1,'o-k'),` (3.4)

где $P1$ и $румi1$ – из Вычпространства (Workspace) заполненного линпрогоптимизатором рис. 3.1. Объект в апострофах - стилизатор кривой. Значения символов, по порядку: маркер точки, вид кривой (сплошная) и цвет (чёрный). Варианты выбора значений – в табл. 2.2 [18].

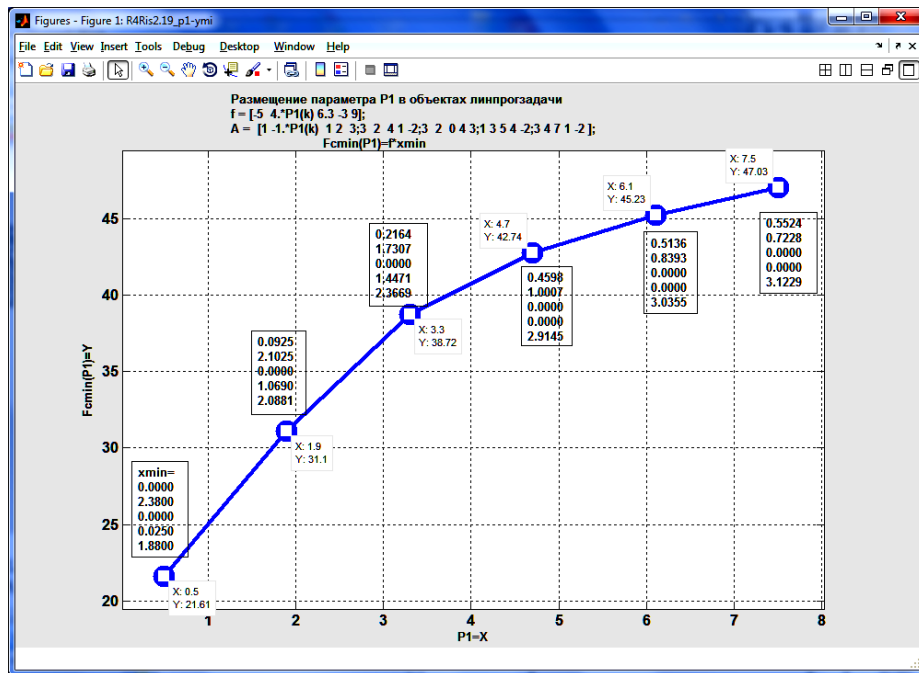


Рис. 3.4. Зависимость минимума f_c (3.1) от $P1$, вычисленная оптимизатором `pLinProgMinRuch.m` рис. 3.1. В рамках – 5-компонентные векторы $rxmin$. В точках кривой: $X = P1$, $Y = f_{\min}(P1)$. Параметр $P1$ входит в ц-функцию f и в левую часть ограничений-неравенств.

Таблица 3.1

Значения стилизатора кривой в команде `plot`

Тип линии		Строительная точка кривой			Цвет линии	
Обозначение в стилизаторе	Описание	Маркер	Обозначение в стилизаторе	Описание	Обозначение в стилизаторе	Цвет
-	Сплошная (по умолчанию)	.	.	Точка	Y	Желтый
--	Штриховая	o	o	Окружность	M	Фиолетовый
:	Точечная	x	x	Крест	C	Голубой
-.	Штрих-пунктирная	+	+	Плюс	R	Красный
		*	*	Звездочка	G	Зеленый
		□	S	Квадрат	B	Синий
		◇	D	Ромб	W	Белый
		▽	V	Треугольник (вниз)	K	Черный
		△	A	Треугольник (вверх)		
		◁	<	Треугольник (влево)		
		▷	>	Треугольник (вправо)		
		☆	P	Пентаграмма		
		☆☆	H	Гексаграмма		

С помощью стилизатора кривой можно изменять тип линии графика, представлять строительные точки различными маркерами (точка, окружность, крест, треугольник с разной ориентацией вершины и т. д.) и менять

цвет линии.

2) Данные в строительных точках графика создаются при включённой кнопке «Data Cursor»  (на панели «Figure Toolbar» – если панель отсутствует, то её нужно вызвать через раздел меню «View») и нажатой клавише «Alt» щелчком мышки в каждой точке. Текст может перемещаться относительно точки дискретно вверх-вниз, влево-вправо.

3) Значения минимизирующего вектора x_{min} в каждой точке вставляются следующим образом. Нажатием кнопки «Т» (текст) на Панели редактирования графика (если Панель отсутствует, её вызывают через раздел меню «View») вставляется шаблон текстового блока с ориентировочными размерами под столбец данных (размеры устанавливаются маркерами шаблона). Двойным щелчком в Вычпространстве на переменной x_{mi} вызывается редактор данных этого массива, столбец, соответствующий заданной точке кривой копируется, а затем вставляется в подготовленный шаблон. Построчным редактированием числа, меньшие 10^{-3} , заменяются на 0, а остальные округляются до 3-го знака после запятой. Маркерами границы шаблона подстраивают под полученную длину чисел столбца.

4) График сохраняется с именем «R4_p1-ymiFamilijaVasha». Это имя должно присутствовать в заголовке фигурокна. Имя копируется из имени файла. В фигурокне командой «Edit_Figure Properties» вызывается Редактор, и в окне «Figure Name» вставляется имя из буфера.

5) Стилизаторы фигуры. Оцифровка осей и их надписи: 12 пт, жирный. Кривая: 3 пт, синяя. Строительные точки: маркер – кружок, заливка и контур синие, размер 12 пт. Шрифт данных x_{min} : 10 пт, жирный.

3.4. Коррекция задачи однопараметрической линейного программирования с использованием выходных объектов `prgm/ prma` в отладочной таблице. Программа `pTabLinProgRuchOtladHtm` построения таблицы

По смыслу этот раздел соответствует разд. 2.5. Разница заключается в том, в разд. 2.5 рассматривался случай однотоочечной коррекции линпрогзадачи, а в настоящем разделе – многотоочечной, когда следует рассматривать для несостоявшегося решения хорт несоответствие левых частей ограничений неравенств и равенств правым в числе точек, равному удвоенному значению точек параметра $P1$. Такая ситуация возникает тогда, когда линпрогоптимум не найден при всех значениях $P1$ – см. рис. 3.2. При каждом из 7 значений параметра флаг решения `efma` равен -2, что свидетельствует об отсутствии решения – см. п.Ц) в разд. 2.6.2.

Для того, чтобы сделать анализ соответствия правых и левых частей в строках выходных объектов более удобным (с использованием инструмента «выделение строк») столбцовое представление выходных объектов следует заменить на горизонтальное, что выполняется программой построения отладочной html-таблицы `pTabLinProgOtladHtm` - см. рис. 3.6, на котором в строках выходного объекта `prma2=[pA2*pxma2 pb2]` выделена единственная строка, в которой соблюдаются заданные ограничения-неравенства при всех значениях параметра $P1ma$.

3.4.1. Пример коррекции условий задачи однопараметрической линпрогмаксимизации

На рис. 3.2, А показана однопараметрическая задача, решаемая линпрогмаксимизатором `pLinProgMaxRuch.m`, а на рис. 3.2, Б – результат решения. Из результата видно, что правильное решение не найдено ни при одном значении параметра $P1ma$: флаг процесса максимизации в каждой точке равен -2. Чтобы флаг процесса сделать равным 1, необходимо изменить условия максимизации: коэффициенты ограничений-неравенств и ограничений-равенств или их свободные члены или и то и другое вместе. Наиболее простой путь – изменение свободных членов, поскольку отладочные объекты `prma2` и `prvm2` позволяют видеть: для каких компонент векторов свободных членов `pb2` и `rbeq2` не соблюдаются соответствующие ограничения `pA2*pxma2` и `pAeq2*pxma2` (`pxma2` – вектор-столбец). Из этого следуют простые правила коррекции условий линпрогзадачи.

3.4.1.1. Правила коррекции условий однопараметрической линпрогзадачи

1) В линпрогоптимизаторе pLinProgMinRuch.m или pLinProgMaxRuch.m, предварительные пуски которых показали отсутствие правильного решения для некоторых (или для всех) значений параметра P1, раскомментировать строку с построителем отладочной таблицы pTabLinProgRuchOtladHtm. Запустить линпрогоптимизатор с получением отладочной таблицы типа «Таблица 1. pTabOtlad_linprogmax 15_Jun_2016 23h55m45s» рис. 3.5.

2) Для коррекции компонент свободного члена pb ограничений-неравенств в отладочной таблице анализируются значения столбцов pA2*pxma2(P1ma) в каждой строке, для которых refm ≤ 0 . Из всех таких значений pA2*pxma2(P1ma) в рассматриваемой строке выбирается наибольшее, увеличивается на 1 единицу в 1-м или 2-м разряде после запятой, и присваивается полученное значение находящемуся в этой строке компоненту вектора pb.

Например, в 1-й строке столбцов pA2*pxma2(P1ma) табл. pTabOtlad_linprogmax 15_Jun_2016 23h55m45s наибольшим значением будет 4.4192. Округляем его до 4.42 и присваиваем это значение 1-му компоненту вектора pb2 вместо 1,6.

Повторяем подобную процедуру замены для других строк столбцов pA2*pxma2(P1ma), за исключением 4-й, в которой ограничения-неравенства соблюдаются. Величина 4-го компонента 9.7 остаётся без изменения. Таким образом, получается откорректированный вектор pb. Вектор pbeq не меняем.

3) Новый вектор pb вставляем в линпрогоптимизатор и запускаем его. Если решение найдено для каждого значения параметра P1 (refm=1), коррекция закончена: в качестве вектора pb принимается его значение из п.2. Если решение не найдено, то повторяется п.2) с запуском линпрогоптимизатора. Описанную итерацию следует повторить несколько раз, пока решение не будет найдено (см. пример – ниже).

4) Компоненты вектора pbeq изменять не следует, несмотря на обнаруживаемые для них нарушения ограничений-равенств: линпрогоптимизатор сам исправит эти нарушения в описанном выше итерационном процессе.

5) После того, как будет найдено решение линпрогзадачи, следует вычислить для всех значений параметра разность ц-функций решения и исходной (pym_resn – pym_ishod) и оценить её максимальную величину – см. табл. 2.3.

Таблица 3.2
Таблица 1. pTabOtlad_linprogmax 15_Jun_2016 23h55m45s

P1ma	0.2000	0.2000	1.2000	1.2000	2.2000	2.2000	3.2000	3.2000	4.2000	4.2000	5.2000	5.2000	6.2000	6.2000
pyma2	5.7122	5.7122	5.5983	5.5983	2.2412	2.2412	5.3767	5.3767	5.2691	5.2691	5.1633	5.1633	2.1601	2.1601
pefma2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2
pA2*pxma2 pb2	pA2*pxma2(P1ma)	pb2	pA2*pxma2(P1ma)	pb2	pA2*pxma2(P1ma)	pb2	pA2*pxma2(P1ma)	pb2	pA2*pxma2(P1ma)	pb2	pA2*pxma2(P1ma)	pb2	pA2*pxma2(P1ma)	pb2
	4.4192	1.6000	4.3784	1.6000	1.1709	1.6000	4.2990	1.6000	4.2605	1.6000	4.2226	1.6000	1.1448	1.6000
	6.8214	3.7000	6.7702	3.7000	1.1716	3.7000	6.6708	3.7000	6.6226	3.7000	6.5750	3.7000	1.1415	3.7000
	7.8855	4.3000	7.7732	4.3000	2.3839	4.3000	7.5550	4.3000	7.4490	4.3000	7.3448	4.3000	2.3072	4.3000
	7.3226	9.7000	7.1581	9.7000	4.7207	9.7000	6.8382	9.7000	6.6827	9.7000	6.5301	9.7000	4.6000	9.7000
	6.8912	1	6.8349	1	2.1553	1	6.7252	1	6.6720	1	6.6197	1	2.1199	1
pAeq2*pxma2 pbeq2	pAeq2*pxma2(P1ma)	pbeq2	pAeq2*pxma2(P1ma)	pbeq2	pAeq2*pxma2(P1ma)	pbeq2	pAeq2*pxma2(P1ma)	pbeq2	pAeq2*pxma2(P1ma)	pbeq2	pAeq2*pxma2(P1ma)	pbeq2	pAeq2*pxma2(P1ma)	pbeq2
	9.0871	4.2000	9.0404	4.2000	5.1584	4.2000	8.9494	4.2000	8.9053	4.2000	8.8619	4.2000	5.1290	4.2000
	5.1185	3.5000	5.0372	3.5000	3.7913	3.5000	4.8791	3.5000	4.8023	3.5000	4.7269	3.5000	3.7319	3.5000
	11.3287	9.2000	11.3084	9.2000	1.8353	9.2000	11.2686	9.2000	11.2496	9.2000	11.2306	9.2000	1.8346	9.2000

Рис. 3.5. Отладочная html–таблица (табл. 3.2), полученная запуском линпрогмаксимизатора рис. 3.2 при раскомментировании строки с именем файла pTabLinProgOtladHtm. Серым цветом выделена единственная строка, в которой соблюдаются частные ограничения-неравенства. Правильный линпрогмаксимум не найден при заданных значениях параметра P1ma. Значения параметра P1ma повторяются в каждой точке расчёта таблицы дважды для того, чтобы разместить отдельными столбцами правые части ограничений и соответствующие им векторы свободных членов.

3.4.1.2. Процедура коррекции условий задачи линпрогмаксимизации оптимизатора рис. 3.2.

А) Запускаем линпрогмаксимизатор рис. 3.2 с раскомментированной строкой с именем файла pTabLinProgOtladHtm. Матлаб выдает отладочную таблицу, показанную на рис. 3.5.

Б) Формируем новый вектор свободного члена pb2 согласно п.2) Правил коррекции... Вектор получился в следующем виде: **pb2** = [4.42 6.83 7.89 9.7 6.9]. Вставляем его в оптимизатор рис. 3.2 и запускаем решение. Матлаб выдает табл. pTabOtlad_linprogmax 22_Jun_2016 17h41m22s, из которой видно, что решение не найдено ни при одном значении параметра P1ma. Однако в последней строке ограничения-неравенства стали соблюдаться (в дополнение к соблюдавшимся неравенствам предпоследней строки) – обе строки выделены серым цветом.

Таблица 3.3
Таблица 1. pTabOtlad_linprogmax 22_Jun_2016 17h41m22s

P1ma	0.2000	0.2000	1.2000	1.2000	2.2000	2.2000	3.2000	3.2000	4.2000	4.2000	5.2000	5.2000	6.2000	6.2000
pyma2	8.4280	8.4280	6.9903	6.9903	6.9868	6.9868	6.9836	6.9836	6.9812	6.9812	6.9782	6.9782	6.9758	6.9758
pefma2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2
pA2*px ma2 pb2	pA2*px ma2(P1 ma)	pb2	pA2*px ma2(P1 ma)	pb2	pA2*px ma2(P1 ma)	pb2	pA2*px ma2(P1 ma)	pb2	pA2*px ma2(P1 ma)	pb2	pA2*px ma2(P1 ma)	pb2	pA2*px ma2(P1 ma)	pb2
	4.7298	4.4200	4.3598	4.4200	4.3546	4.4200	4.3500	4.4200	4.3465	4.4200	4.3422	4.4200	4.3388	4.4200
	7.2944	6.8300	6.9775	6.8300	6.9728	6.8300	6.9685	6.8300	6.9651	6.8300	6.9614	6.8300	6.9583	6.8300
	8.7131	7.8900	8.8325	7.8900	8.8266	7.8900	8.8214	7.8900	8.8172	7.8900	8.8125	7.8900	8.8087	7.8900
	7.5488	9.7000	7.6859	9.7000	7.6696	9.7000	7.6550	9.7000	7.6426	9.7000	7.6303	9.7000	7.6197	9.7000
	6.4399	6.9000	6.2069	6.9000	6.1952	6.9000	6.1848	6.9000	6.1759	6.9000	6.1671	6.9000	6.1596	6.9000
pAeq2* pxma2 pbeq2	pAeq2* pxma2(P lma)	pbeq2	pAeq2* pxma2(P lma)	pbeq2	pAeq2* pxma2(P lma)	pbeq2	pAeq2* pxma2(P lma)	pbeq2	pAeq2* pxma2(P lma)	pbeq2	pAeq2* pxma2(P lma)	pbeq2	pAeq2* pxma2(P lma)	pbeq2
	4.7918	4.2000	4.8400	4.2000	4.8027	4.2000	4.7694	4.2000	4.7399	4.2000	4.7128	4.2000	4.6885	4.2000
	3.6960	3.5000	3.7033	3.5000	3.6832	3.5000	3.6653	3.5000	3.6494	3.5000	3.6350	3.5000	3.6219	3.5000
	9.4578	9.2000	9.4788	9.2000	9.4625	9.2000	9.4480	9.2000	9.4352	9.2000	9.4234	9.2000	9.4128	9.2000

В) Формируем новый вектор pb2 в виде: **pb2** = [4.8 7.3 8.9 9.7 6.9]. Исходные данные для значений компонентов pb2 отмечены зелёным цветом. Компоненты из серых строк берём без изменений. После запуска оптимизатора pLinProgMaxRuch.m с вставленным новым вектором pb2 получим следующую отладтаблицу.

Таблица 3.4
Таблица 1. pTabOtlad_linprogmax 23_Jun_2016 15h35m01s

P1ma	0.2000	0.2000	1.2000	1.2000	2.2000	2.2000	3.2000	3.2000	4.2000	4.2000	5.2000	5.2000	6.2000	6.2000
pyma2	7.8694	7.8694	7.1844	7.1844	7.1841	7.1841	7.1839	7.1839	7.1836	7.1836	7.1834	7.1834	7.1832	7.1832
pefma2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2
pA2*px ma2 pb2	pA2*px ma2(P1 ma)	pb2	pA2*px ma2(P1 ma)	pb2	pA2*px ma2(P1 ma)	pb2	pA2*px ma2(P1 ma)	pb2	pA2*px ma2(P1 ma)	pb2	pA2*px ma2(P1 ma)	pb2	pA2*px ma2(P1 ma)	pb2
	4.5316	4.8000	4.3572	4.8000	4.3568	4.8000	4.3564	4.8000	4.3560	4.8000	4.3558	4.8000	4.3554	4.8000
	7.1557	7.3000	7.0064	7.3000	7.0060	7.3000	7.0057	7.3000	7.0053	7.3000	7.0051	7.3000	7.0048	7.3000
	8.9205	8.9000	8.9802	8.9000	8.9797	8.9000	8.9793	8.9000	8.9788	8.9000	8.9785	8.9000	8.9781	8.9000
	7.6868	9.7000	7.7595	9.7000	7.7581	9.7000	7.7568	9.7000	7.7557	9.7000	7.7548	9.7000	7.7538	9.7000
	6.2254	6.9000	6.1194	6.9000	6.1184	6.9000	6.1175	6.9000	6.1167	6.9000	6.1160	6.9000	6.1153	6.9000
pAeq2* pxma2 pbeq2	pAeq2* pxma2(P lma)	pbeq2	pAeq2* pxma2(P lma)	pbeq2	pAeq2* pxma2(P lma)	pbeq2	pAeq2* pxma2(P lma)	pbeq2	pAeq2* pxma2(P lma)	pbeq2	pAeq2* pxma2(P lma)	pbeq2	pAeq2* pxma2(P lma)	pbeq2
	4.2148	4.2000	4.2545	4.2000	4.2513	4.2000	4.2485	4.2000	4.2459	4.2000	4.2437	4.2000	4.2416	4.2000
	3.5049	3.5000	3.5173	3.5000	3.5156	3.5000	3.5141	3.5000	3.5127	3.5000	3.5115	3.5000	3.5104	3.5000
	9.2064	9.2000	9.2237	9.2000	9.2223	9.2000	9.2211	9.2000	9.2200	9.2000	9.2190	9.2000	9.2181	9.2000

Решение снова не найдено, но количество строк с несоблюдением ограничений-неравенств уменьшилось до 1 (строки с соблюдением неравенств выделены серым).

Г) Формируем новый вектор pb2 в виде: **pb2** = [4.8 7.3 9 9.7 6.9]. Исходное значение для изменяемого 3-го компонента pb2 отмечены зелёным цветом. После запуска оптимизатора с вставленным новым вектором pb2 получаем отладтаблицу (табл. 3.5), свидетельствующую, что решение для всех точек параметра P1ma найдено:

pefma2=1 – табл. pTabOtlad_linprogmax 23_Jun_2016 15h52m37s.

Таблица 3.5
Таблица 1. pTabOtlad_linprogmax 23_Jun_2016 15h52m37s

P1ma	0.2000	0.2000	1.2000	1.2000	2.2000	2.2000	3.2000	3.2000	4.2000	4.2000	5.2000	5.2000	6.2000	6.2000
pyma2	7.9179	7.9179	7.2330	7.2330	7.2146	7.2146	7.2100	7.2100	7.2084	7.2084	7.2084	7.2084	7.2084	7.2084
pefma2	1	1	1	1	1	1	1	1	1	1	1	1	1	1
pA2*px ma2 pb2	pA2*px ma2(P1 ma)	pb2	pA2*px ma2(P1 ma)	pb2	pA2*px ma2(P1 ma)	pb2	pA2*px ma2(P1 ma)	pb2	pA2*px ma2(P1 ma)	pb2	pA2*px ma2(P1 ma)	pb2	pA2*px ma2(P1 ma)	pb2
	4.6134	4.8000	4.3649	4.8000	4.3601	4.8000	4.3589	4.8000	4.3619	4.8000	4.3618	4.8000	4.3616	4.8000
	7.1816	7.3000	7.0158	7.3000	7.0118	7.3000	7.0108	7.3000	7.0139	7.3000	7.0138	7.3000	7.0137	7.3000
	9.0000	9	9.0000	9	9.0000	9	9.0000	9	9.0000	9	9.0000	9	9.0000	9
	7.7958	9.7000	7.7628	9.7000	7.7649	9.7000	7.7654	9.7000	7.7712	9.7000	7.7714	9.7000	7.7715	9.7000
	6.2964	6.9000	6.1163	6.9000	6.1132	6.9000	6.1125	6.9000	6.1128	6.9000	6.1128	6.9000	6.1128	6.9000
pAeq2* pxma2 pbeq2	pAeq2* pxma2(P lma)	pbeq2	pAeq2* pxma2(P lma)	pbeq2	pAeq2* pxma2(P lma)	pbeq2	pAeq2* pxma2(P lma)	pbeq2	pAeq2* pxma2(P lma)	pbeq2	pAeq2* pxma2(P lma)	pbeq2	pAeq2* pxma2(P lma)	pbeq2
	4.2000	4.2000	4.2000	4.2000	4.2000	4.2000	4.2000	4.2000	4.2000	4.2000	4.2000	4.2000	4.2000	4.2000
	3.5000	3.5000	3.5000	3.5000	3.5000	3.5000	3.5000	3.5000	3.5000	3.5000	3.5000	3.5000	3.5000	3.5000
	9.2000	9.2000	9.2000	9.2000	9.2000	9.2000	9.2000	9.2000	9.2000	9.2000	9.2000	9.2000	9.2000	9.2000

Д) Оценка разности ц-функций решения (табл. 3.5) и исходной (табл. 3.2), когда решению не было найдено. Процедура оценки отражена в табл. 3.6. Для оценки разности следует скопировать первые 3 строки из исходной (табл. 3.2) и последней (табл. 3.5) отладтаблиц в пустой Уорд-документ, удалить повторяющиеся столбцы, вставить 2 допстроки с именами из табл. 3.6, скопировать строки pyma2_resh и pyma2_ishod в сесфайл и сформировать из них векторы. Скопировать также имена 2 последних строк табл. 3.6, как команд Матлаба (точка перед знаком деления в 5-й строке обязательна), выделить все скопированное и запустить вычисления. Результаты вычислений копируются в 4-ю и 5-ю строки Уорд-таблицы.

Таблица 3.6. Оценка разности ц-функций решения (табл. pTabOtlad_linprogmax 23_Jun_2016 15h52m37s) и исходной (табл. pTabOtlad_linprogmax 15_Jun_2016 23h55m45s).

P1ma	0.2000	1.2000	2.2000	3.2000	4.2000	5.2000	6.2000
pyma2_resh	7.9179	7.2330	7.2146	7.2100	7.2084	7.2084	7.2084
pyma2_ishod	8.4280	6.9903	6.9868	6.9836	6.9812	6.9782	6.9758
D=pyma2_resh- pyma2_ishod	-0.5101	0.2427	0.2278	0.2264	0.2272	0.2302	0.2326
Otnos=D*100./ pyma2_resh, %	-6.4424	3.3555	3.1575	3.1401	3.1519	3.1935	3.2268

Как следует из табл. 3.6, найденное решение уточняет значения ц-функции из табл. 3.2 по абсолютной величине в диапазоне от 3.2 до 6,5 %, т.е. на небольшие величины. Очевидно, что такое уточнение нельзя рассматривать, как правило: при других условиях величины уточнений могут быть большими.

3.4.1.3. Программа pTabLinProgRuchOtladHtm построения отладтаблицы.

Эта программа-построитель отладочной таблицы используется для обоих типов «ручных» однопараметрических линпрогоптимизаторов (см. рис. 3.1 и 3.2). Она автоматически меняет заголовки и имена файлов отладтаблиц, а также имена переменных в таблице, в зависимости от типа запущенного оптимизатора. Программа построителя представлена на рис. 3.6. Назначение ряда команд и всех блоков команд указано в построчных комментариях.

```

1. Построитель отладочной html-таблицы pTabLinProgRuchOtladHtm для отладки
% исходных данных однопараметрической линпрогоптимизации "ручных" оптимизаторов pLinProgMinRuch.m
% и pLinProgMaxRuch.m.
2. po=[];T1_po2=[]; T2_pyo2=[];T3_peco2=[];T4_pnroZagCel={};T5_prvoZagCel={};T6_pnroKlay=[]; T7_prvoKlay=[];
% Очистка результатов предыдущего запуска построителя отладтаблицы.
3. T10_pnroKlayCel={}; T11_prvoKlayCel={};T12_TabOtladSodergCel={};T14_stbLthAoCel={};
4. T15_stbLthAeqoCel={};T18_TabOtladStbImenCel={};T19_pLinPrTabOtladCel={};

```

```

5. if pLmi_pLma==1 % Формирование имён переменных, значений условий и части результ. линпрограммизатора.
6. po=P1;
7. Sod_T4_1={'pA*pxmi1(P1)'};
8. Sod_T4_2={'pb'};
9. Sod_T5_1={'pAeq*pxmi1(P1)'};
10. Sod_T5_2={'pbeq'};
11. Sod_T13={'P1','pymi1','pefmi1'};
12. Sod_T16={'pA*pxmi1 pb'};
13. Sod_T17={'pAeq*pxmi1 pbeq'};
14. pAo=pA;
15. pAeqo=pAeq;
16. pyo=pymi1;
17. pnro=pnrmi1;
18. prvo=prvmi1;
19. pefo=pefmi1;
20. elseif pLmi_pLma==-1 % Формирование имён переменных, значений условий и части результ. линпрограммизатора.
21. po=P1ma;
22. Sod_T4_1={'pA2*pxma2(P1ma)'};
23. Sod_T4_2={'pb2'};
24. Sod_T5_1={'pAeq2*pxma2(P1ma)'};
25. Sod_T5_2={'pbeq2'};
26. Sod_T13={'P1ma','pyma2','pefma2'};
27. Sod_T16={'pA2*pxma2 pb2'};
28. Sod_T17={'pAeq2*pxma2 pbeq2'};
29. pAo=pA2;
30. pAeqo=pAeq2;
31. pyo=pyma2;
32. pefo=pefma2;
33. pnro=pnrma2;
34. prvo=prvma2;
35. end
36. for k3=1:length(po) % Создание строк ро, уо, ефо с удвоением количества их значений .
37. T1_po2(2*k3-1)=po(k3);
38. T1_po2(2*k3)=po(k3);
39. T2_py2(2*k3-1)=pyo(k3);
40. T2_py2(2*k3)=pyo(k3);
41. T3_pefo2(2*k3-1)=pefo(k3);
42. T3_pefo2(2*k3)=pefo(k3);
43. T4_pnroZagCel(2*k3-1)=Sod_T4_1; % Формирование ячкстроки заголовков столбцов
44. % неравенств типа "pA2*pxma2 pb2".
45. T4_pnroZagCel(2*k3)=Sod_T4_2;
46. T5_prvoZagCel(2*k3-1)=Sod_T5_1;
47. T5_prvoZagCel(2*k3)=Sod_T5_2; % Ячкстрока заголовков столбцов неравенств.
48. if isempty(pAo) % Проверка пустого значения матрицы pAo.
49. T6_pnroKlay=[T6_pnroKlay []];
50. else
51. T6_pnroKlay=[T6_pnroKlay pnro(:,k3)];
52. end
53. if isempty(pAeqo) % Проверка пустого значения матрицы pAeqo.
54. T7_prvoKlay=[T7_prvoKlay []];
55. else
56. T7_prvoKlay=[T7_prvoKlay prvo(:,k3)];
57. end
58. end
59. T8_pxyef=cat(1,T1_po2, T2_py2, T3_pefo2); % Вертикальное склеивание строк с удвоен. кол.значений.
60. T9_pyefCell=num2cell( T8_pxyef); % Перевод в ячк-формат (cell-формат) матрицы T8.
61. if isempty(pA) % Проверка пустого значения матрицы pA. Формирование ячкзначений вых. об. рпго.
62. T10_pnroKlayCel(1,1:2*length(po))={'[]'};
63. else
64. T10_pnroKlayCel=num2cell(T6_pnroKlay);
65. end
66. if isempty(pAeqo) % Проверка пустого значения матрицы pAeqo. Формирование ячкзначений вых. об. ррго.
67. T11_prvoKlayCel(1,1:2*length(po))={'[]'};
68. else
69. T11_prvoKlayCel=num2cell(T7_prvoKlay);
70. end
71. % Матрица значений отладтаблицы.
72. T12_TabOtladSodergCel=cat(1,T9_pyefCell, T4_pnroZagCel,T10_pnroKlayCel,T5_prvoZagCel,T11_KlayCel);
73. T13_StblmenCel=Sod_T13; % Столбец имён 3 верхних строк отладтаблицы в ячкформате.
74. if isempty(pAo) % Провер. пуст. знач. matr. Ao. Формир. столбца пуст. загол. для строк (компонент) рпго.

```

```

75. T14_stbLthAoCel={''}; % Столбец пустых ячеек заменяется одной пустой ячейкой.
76. else
77. T14_stbLthAoCel(1:length(pAo(:,1)),1)={''}; % Столбец пустых ячеек по столбцу pAo(:,1).
78. end
79. if isempty(pAeqo) % Провер. пуст. знач. matr. Aeqo. Формир. столбца пуст. загол. для строк (компонент) prvo.
80. T15_stbLthAeqoCel={''}; % Столбец пустых ячеек заменяется одной пустой ячейкой.
81. else
82. T15_stbLthAeqoCel(1:length(pAeqo(:,1)),1)={''}; % Столбец пустых ячеек по столбцу pAeqo(:,1).
83. end
84. T16_Cel=Sod_T16;
85. T17_Cel=Sod_T17;
86. % Столбец имён строк отладтаблиц в ячкформате.
87. T18_TabOtladStbImenCel=cat(1, T13_StbImenCel, T16_Cel, T14_stbLthAoCel, T17_Cel, T15_stbLthAeqoCel);
88. T19_pLinPrTabOtladCel=cat(2, T18_TabOtladStbImenCel, T12_TabOtladSodergCel); % Отладтаблица линрол-
% запуска в ячкформате.
89. % Формирование заголовка и имени файла в инж-формате отладтаблицы линпрогона.
90. T20_SumDatKor=datestr(clock); T20_SumDatKor(3)='_'; T20_SumDatKor(7)='_'; T20_SumDatKor(15)='h';
91. T20_SumDatKor(18)='m'; T20_SumDatKor(21)='s'; % Дата и время запуска линпрогоптимизации.
92. if pLmi_pLma==1 % Собственное имя отладтаблицы.
93. T21_ImjaTb=char('pTabOtlad_linprogmin');
94. elseif pLmi_pLma== -1
95. T21_ImjaTb=char('pTabOtlad_linprogmax');
96. end
97. ImFileandZagIngOtladTab=[T21_ImjaTb blanks(2) T20_SumDatKor]; % Имя файла инжотладтаблицы и её заго-
% ловка.
98. report pLinProgOtladTab.rpt % Выдача отладтаблицы в html-формате. Работает только после создания и на-
% стройки html-шаблона отладтаблицы на основе генератора отчётов с именем pLinProgOtladTab.rpt. Иначе выдаёт
% ошибку.

```

Рис. 3.6. Построитель отладтаблицы для линпрогоптимизаторов рис. 3.1 и 3.2. Перед копированием в текст-файл Матлаба нумерацию строк следует удалить: Матлаб воспринимает эти номера как ошибки.

Строка с именем программы является предпоследней в линпрогоптимизаторах, и закомментирована. Если после пробного запуска оптимизатора обнаруживается, что для ряда значений параметра (или для всех) решение не найдено, то строка раскомментируется для формирования отладтаблицы. Формирование html-шаблона отладтаблицы – в разд. 3.4.2.

3.4.2. Формирование html-шаблона отладтаблиц.

Шаблон отладтаблицы формируется в Генераторе отчётов, который затем в построителе pTabLinProgRuchOtladHtm (строка с командой report...) преобразует содержание отладтаблицы, сформированное в ячкформате (строка с T19_pLinPrTabOtladCel ...) в html-формат: заявленные в MATLAB Report Generator'е другие формы таблицы (pdf, rtf и doc) в Матлабе 2009 не реализуются.

Формирование шаблона отладтаблиц pLinProgOtladTab.rpt и настройка Генератора отчётов производится в следующем порядке.

И. Запустите любой из 2 оптимизаторов: pLinProgMinRuch.m или pLinProgMaxRuch.m при закомментированной строке с именем «pLinProgRuchOtladTab». Проверьте появление в Вычпространстве переменной T19_pLinPrTabOtladCel, т.е. имени содержания отладтаблицы в ячк-формате.

К. Командами Start (кнопка в левом нижнем углу окна Матлаба) _Matlab_Matlab-Report-Generator_Report-File-List вызывается панель Генератора отчетов. На появившейся панели нажимается кнопка «Crtate and edit a new file», а в списке «Name» дважды щёлкается строка «Table»: панель примет вид, аналогичный рис. 3.6 А, без ввода настроек.

Л. В дереве «Report Generator» выделите 1-ю строку «Report». Сохраните файл с именем pTabLinProgOtladTab.rpt. Проконтролируйте: имя сохранённого файла должно появиться в заголовке окна Репорт Эксплорера и в ветви «Report» дерева «Report Generator».

М. Во 2-е поле «Filename» на рис. 3.7, А, отмеченном звёздочкой, вставляется выражение %<ImFileandZagIngOtladTab>, в угловых скобках которого стоит командия файлов отладтаблиц линпрогоптимизаторов, создаваемое в строке 97 Построителя отладтаблиц pTabLinProgOtladHtm. Это командия будет формировать имена html-файлов отладтаблиц, сохраняемых в действующей папке. После вставки это имя вместе с именем действующей папки появится на панели «Report Options» в 1-й строке, отмеченной звёздочкой. В дальнейшем в этом месте будет появляться имя последнего сохранённого файла отладтаблицы последнего запуска одного из двух линпрогоптимизаторов.

Н. Выделяется 2-я ветвь в дереве Репорт генератора – см. рис. 3.7, Б. В строки, помеченные звёздочка-

ми вставляются указанные в них значения. Сохраните введенные изменения.

О. Командами меню File_Generate-M-file запустите генерацию М-файла html-шаблона отладтаблицы pTabLinProgOtladTab. Проконтролируйте появление в текущей папке файла buildpLinProgOtladTab.m и его же появление в окне Редактора-отладчика.

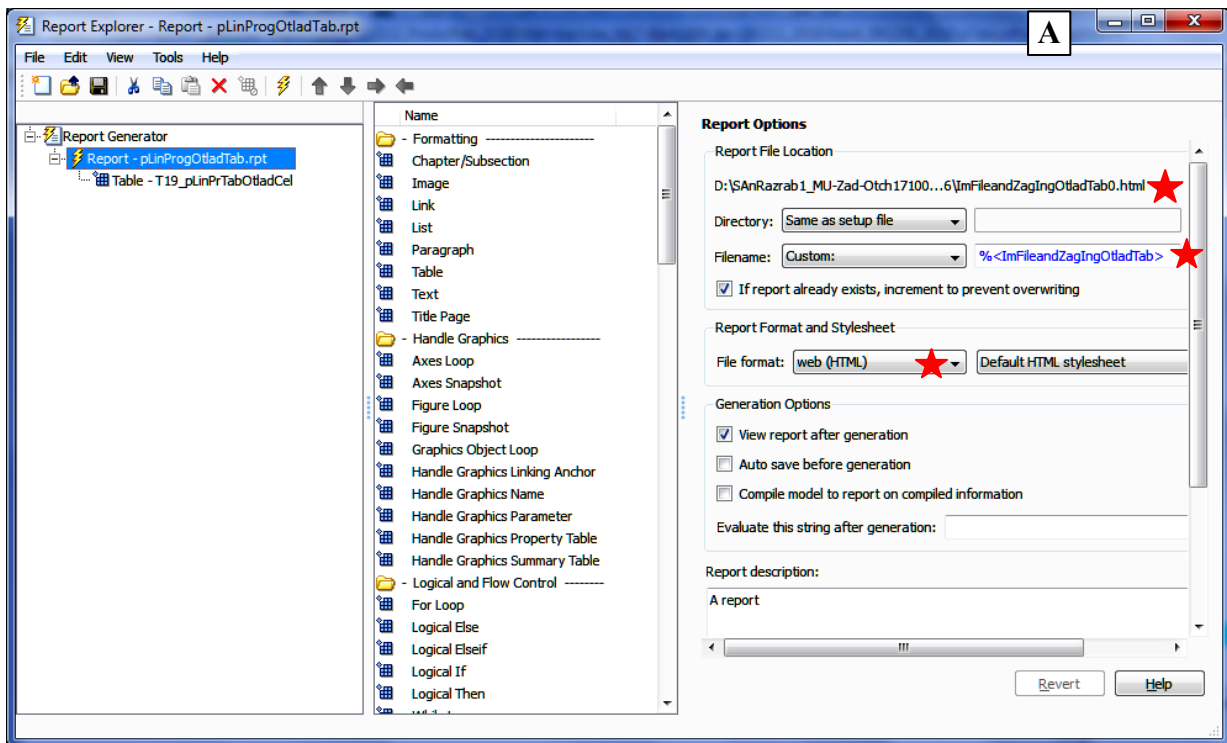
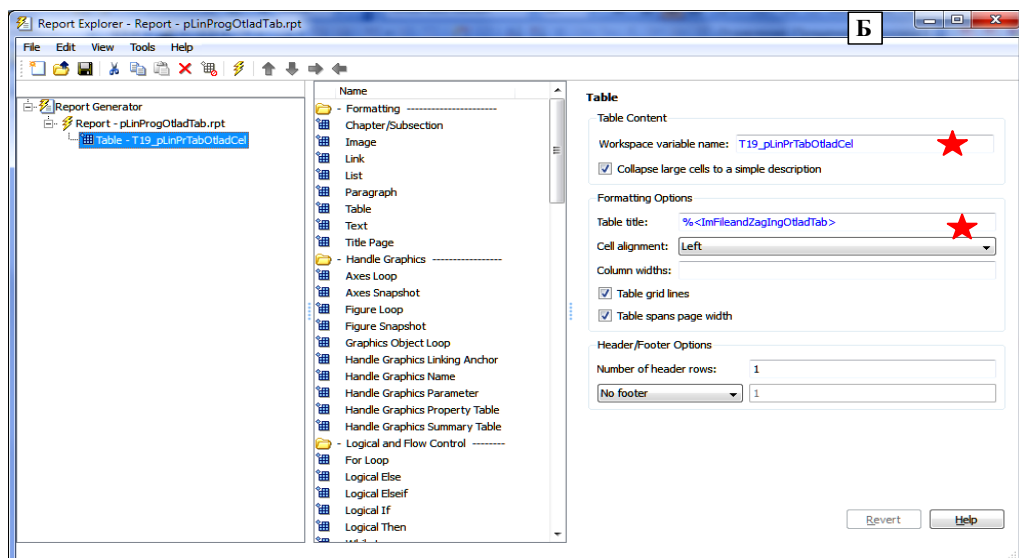


Рис. 3.7. Вид Репорт Эксплорера при выделении 1-й ветви в дереве Репорт Генератора (А) после сохранения с заданным именем pTabLinProgOtladTab.rpt и ввода настройки во вторую строку со звездочкой. В упомянутую строку вставляется команда файлов отладтаблиц, создаваемое в строке 97 Построителя отладтаблиц рис.3.6. Продолжение рис.3.7 – на следующей странице.

П. Нажатием кнопки «Report»  на панели инструментов запускается Генератор отчетов на преобразование итог-таблицы из ячк- в html-формат: в действующей папке появятся файл со структурой имени типа «Таблица 1. pTabOtlad_linprogmax 15_Jun_2016 23h55m45s». Одновременно появится Окно сообщения о ходе преобразования и html-отладтаблица вида рис. 3.5. Настройка шаблона и Генератора отчетов для отладтаблиц линпрогоптимизаторов закончена.

Р. В завершение настройки следует снять закомментированность предпоследних строк у обоих линпрогоптимизаторов (рис. 3.1 и 3.2) и запустить их для проверки правильности формирования отладтаблиц.



Продолж. рис. 3.7. Настройки во второй ветви Репорт Генератора: отмечены звёздочками. 1-я вставка – коман-

дия из строки 88 рис. 3.6. Вставка 2 - командия из строки 97 в угловых скобках с начальным знаком процентов.

3.5. Построение итогтаблиц однопараметрической линпрогооптимизации для оптимизаторов с ручной настройкой pLinProgMinRuch.m и pLinProgMaxRuch.m

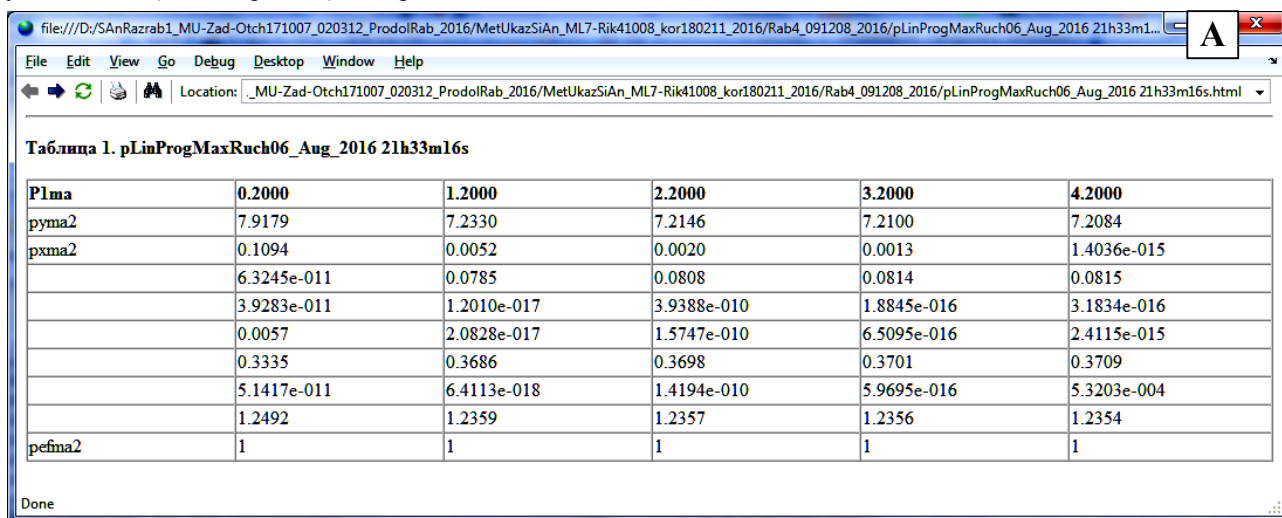
Результат работы каждого линпрогооптимизатора представляет собой набор из 4 переменных (например, P1ma, рума2, рхма2, рефма2), значения которых помещены в Вычпространство. Они не являются единым объектом. Результаты ряда линпрогооптимизаций неудобно сравнивать друг с другом. В текстовый редактор переменные копируются в виде набора чисел. Чтобы этот набор представить в виде инженерной таблицы необходимо задействовать табличные инструменты редактора. Для исключения указанных недостатков в состав оптимизаторов в предпоследнюю строку введен формирователь итогтаблицы результата линпрогооптимизации - pLinProgRuchP1OptTabFormer, который также производит сохранение файла оптимизатора в действующей папке с именем заголовка созданной итогтаблицы. Ниже приводится описание итогтаблиц и текста программы формирователя.

3.5.1. Описание итогтаблиц

Оба вида таблиц представлены на рис. 3.8.

Свойства таблиц.

А. Заголовок таблицы включает (например для табл. А, pLinProgMinRuch03_Aug_2016 13h35m39s) имя оптимизатора, создавшего таблицу, и даты и времени формирования таблицы. Причём, дата и время составляют уникальный (неповторимый) номер таблицы.



P1ma	0.2000	1.2000	2.2000	3.2000	4.2000
рума2	7.9179	7.2330	7.2146	7.2100	7.2084
рхма2	0.1094	0.0052	0.0020	0.0013	1.4036e-015
	6.3245e-011	0.0785	0.0808	0.0814	0.0815
	3.9283e-011	1.2010e-017	3.9388e-010	1.8845e-016	3.1834e-016
	0.0057	2.0828e-017	1.5747e-010	6.5095e-016	2.4115e-015
	0.3335	0.3686	0.3698	0.3701	0.3709
	5.1417e-011	6.4113e-018	1.4194e-010	5.9695e-016	5.3203e-004
	1.2492	1.2359	1.2357	1.2356	1.2354
pefma2	1	1	1	1	1

Таблица 1. pLinProgMaxRuch03_Aug_2016 23h55m45s

Б

P1ma	0.2000	1.2000	2.2000	3.2000	4.2000
рума2	7.9179	7.2330	7.2146	7.2100	7.2084
рхма2	0.1094	0.0052	0.0020	0.0013	1.4036e-015
	6.3245e-011	0.0785	0.0808	0.0814	0.0815
	3.9283e-011	1.2010e-017	3.9388e-010	1.8845e-016	3.1834e-016
	0.0057	2.0828e-017	1.5747e-010	6.5095e-016	2.4115e-015
	0.3335	0.3686	0.3698	0.3701	0.3709
	5.1417e-011	6.4113e-018	1.4194e-010	5.9695e-016	5.3203e-004
	1.2492	1.2359	1.2357	1.2356	1.2354
pefma2	1	1	1	1	1

Рис. 3.8. Виды итогтаблиц. А – для линпрогминимизатора pLinProgMinRuch; снимок окна Интернет-просматривателя с таблицей. Б - для линпрогмаксимизатора pLinProgMaxRuch; копия из окна Интернет-просматривателя с таблицей.

Б. Как видно из заголовка окна Интернет-просматривателя (рис. 3.8,А) и поля «Location» имя файла итогтаблицы совпадает с её заголовком. Сам же файл находится в действующей папке сессии Матлаба.

В. При формировании ячек-содержания итогтаблицы формирователем pLinProgRuchP1OptTabFormer производится также сохранение файла оптимизатора с именем заголовка созданной оптимизатором итогтаблицы – см. строки «соруfile ...» в Формирователе рис. 3.10. Таким образом, в действующей папке появятся 2 файла с одинаковыми именами, но с разными расширениями: html – для файла итогтаблицы и m – для оптимизатора, создавшего эту таблицу. Эти файлы будут размещены рядом – см. рис. 3.9.

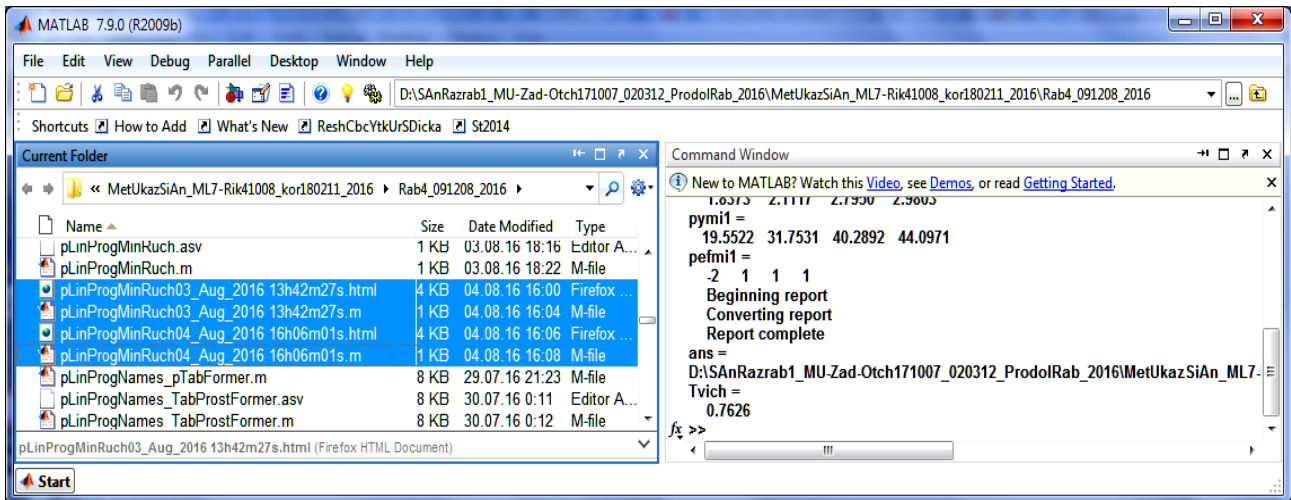


Рис. 3.9. Размещение 2 файлов итогтаблиц и сохранённых с именами заголовков этих таблиц 2 файлов оптимизаторов, создавших эти таблицы. 4 файла выделены в действующей папке.

Рассматриваемое сохранение обеспечивает создание постоянной связи итогтаблицы с условиями р-линпрогоптимизации, выдавшими результат этой таблицы. Эти условия (т.е. оптимизатор) всегда можно вызывать для дополнительных исследований, включая сравнения с рядом последующих результатов.

Г. Как видно из рис. 3.8, Б, итогтаблица просто копируется в текстовый редактор с использованием системных средств копирования и вставки Windows. Правда, при такой простой вставке следует обратить внимание на возможное нежелательное автоматическое форматирование данных таблицы, которое следует ликвидировать ручными инструментами текстредактора.

3.5.2. Формирователь итогтаблиц р-линпрогоптимизации - pLinProgRuchP1OptTabFormer

Программа формирователя – на рис. 3.10. Пояснения к командам – в построчных примечаниях. Описание команд и операторов Матлаба, используемых в Формирователе – в Приложении 3.

% Построитель итоговых таблиц р-линпрогоптимизации pLinProgRuchP1OptTabFormer
% для "ручных" оптимизаторов pLinProgMinRuch.m и pLinProgMaxRuch.m.

if pLmi_pLma==1 % Формирование результата и имён выходных переменных линпрогминимизатора pLinProgMinRuch.m.

T1_p1xyefmi=cat(1,P1,pym1,pxmi1,pefmi1); % Вертикальное склеивание вых. данных линпрогминимизации в таблицу.

T2_p1xyefmiCell=num2cell(T1_p1xyefmi); % Преобразование числовой табл.Т1 в ячейковый формат (ячек-формат)

T3_stbXmi_Cell(1:(length(pxmi1)-1),1)={' '}; % создание ячек-столбца пустых строк в кол., равном длине вектора pxmi1-1

T4_stbNameTabMinCell=['P1';'pym1';'pxmi1';T3_stbXmi_Cell; 'pefmi1']; % создание ячек-столбца с именами строк табл. ИтогРМинПростТаб.

T5_ProstTabOptCell=cat(2,T4_stbNameTabMinCell,T2_p1xyefmiCell); % Данные для табл. ИтогРМинПростТаб в ячек-формате.

elseif pLmi_pLma== -1 % Формирование результата и имён выходных переменных линпрогмаксимизатора pLinProgMaxRuch.m.

T1_p1xyefma=cat(1,P1ma,pyuma2,pxma2,pefma2); % Вертикальное склеивание вых. данных линпрогминимизации в таблицу.

```

T2_p1xyefmaCell=num2cell(T1_p1xyefma); % Преобразование числовой табл.Т1 в ячейковый формат
(ячк-формат)
T3_stbXma_Cell(1:(length(pxma2)-1),1)={' '}; % создание ячк-столбца пустых строк в кол., равном
длине вектора pxma1-1
T4_stbNameTabMaxCell=['P1ma';'pyma2';'pxma2';T3_stbXma_Cell; 'pfma2']; % создание ячк-столбца
с именами строк табл. ИтогРМаксПростТаб.
T5_ProstTabOptCell=cat(2,T4_stbNameTabMaxCell,T2_p1xyefmaCell); % Данные для табл. Итог-
РМаксПростТаб в ячк-формате.
end
% Формирование заголовка и имени файла в html-формате итогтаблицы линпрогзапуска "ручных" оп-
тимизаторов.
T6_SumDatKor=datestr(clock); T6_SumDatKor(3)='_';T6_SumDatKor(7)='_'; T6_SumDatKor(15)='h';
T6_SumDatKor(18)='m';T6_SumDatKor(21)='s'; % Дата и время запуска линпрогручоптимизатора.
if pLmi_pLma==1 % Собственное имя итогтаблицы.
    T9_ImjaZagTbRuch1=char('pLinProgMinRuch');
    T11_ImjaZagTbRuchA=[T9_ImjaZagTbRuch1 T6_SumDatKor]; % Загол. итогтаблицы р-
линпрогминимизации.
    copyfile('pLinProgMinRuch.m', [T11_ImjaZagTbRuchA '.m'],'f') % Копирование ф.
pLinProgMinRuch.m с именем заг. итогтаб. р-линпрогмин.
elseif pLmi_pLma==-1
    T9_ImjaZagTbRuch1=char('pLinProgMaxRuch');
    T11_ImjaZagTbRuchA=[T9_ImjaZagTbRuch1 T6_SumDatKor]; % Загол. итогтаблицы р-
линпрогмаксимизации.
    copyfile('pLinProgMaxRuch.m', [T11_ImjaZagTbRuchA '.m'],'f') % Копирование ф. pLinProg-
MaxRuch.m с именем заг. итогтаб. р-линпрогмакс.
end
report pLinProgItogRuchTab.rpt % Выдача итогтаблицы в html-формате. Работает только после соз-
%дания и настройки html-шаблона итогтаблицы на основе генератора отчётов с именем
%pLinProgItogRuchTab.rpt. Иначе выдаёт ошибку.

```

Рис. 3.10. Формирователь итогтаблиц pLinProgRuchP1OptTabFormer. Работает после создания и на-
стройки html-шаблона итогтаблиц в генераторе отчётов с именем pLinProgItogRuchTab.rpt – см. разд.
3.5.3. Для работы код следует скопировать из электронной версии Пособия и поместить в отдельный тек-
стовый М-файл с упомянутым именем и создать шаблон html-таблиц согласно разд. 3.5.3.

3.5.3. Настройка html-шаблона итогтаблиц pLinProgItogRuchTab.rpt для формирователя pLinProgRuchP1OptTabFormer

Шаблон итогтаблицы формируется в Генераторе отчётов, который затем в формирователе (последняя строка, рис. 3.10) преобразует итогтаблицы, выполненные в ячкформате (строка для переменной T5_ProstTabOptCell на рис. 3.10) в html-формат.

Формирование шаблона итогтаблиц pLinProgItogRuchTab.rpt и настройка Генератора отчётов производится в следующем порядке.

А. Запустите любой из 2 оптимизаторов: pLinProgMinRuch.m или pLinProgMaxRuch.m при раскомментированной строке с Формирователем и с закомментированной в Формирователе строкой «report pLinProgItogRuchTab». Проверьте появление в Вычпространстве переменной T5_ProstTabOptCell, т.е. данных итогтаблицы в ячк-формате.

Б. Командами Start (кнопка в левом нижнем углу окна Матлаба)_Matlab_Matlab-Report-Generator_Report-File-List вызывается панель Генератора отчетов. На появившейся панели нажимается кнопка «Crtate and edit a new file», а в списке «Name» дважды щёлкается строка «Table»: панель примет вид, аналогичный рис. 3.11 А, без ввода настроек.

В. В дереве «Report Generator» выделите 1-ю строку «Report». Сохраните файл с именем pLinProgItogRuchTab.rpt. Проконтролируйте: имя сохранённого файла должно появиться в заголовке окна Репорт Эксплорера и в ветви «Report» дерева «Report Generator».

Г. Во 2-е поле «Filename» на рис. 3.11, А, отмеченное звёздочкой, вставляется выражение %<T11_ImjaZagTbRuchA>, в угловых скобках которого стоит командия файлов итогтаблиц р-линпрогоптимизаторов, создаваемое в строке с упомянутым именем Построителя итогтаблиц pLinProgRuchP1OptTabFormer. Это командия будет формировать имена html-файлов итогтаблиц, сохраняемых в действующей папке. После вставки это имя вместе с именем действующей папки появится на панели

«Report Options» в 1-й строке. В дальнейшем в этом месте будет появляться имя последнего сохранённого файла итогтаблицы последнего запуска одного из двух р-линпрогоптимизаторов.

Д. Две остальные звёздочки – для контроля: в квадрате «If report...» не должно быть галочки, а во втором Контрольном поле должно стоять «web (HTML)».

Е. Выделяется 2-я ветвь в дереве Репорт генератора – см. рис. 3.11, Б. В строки, помеченные звёздочками вставляются указанные в них значения. Сохраните введённые изменения.

Ж. Командами меню File_Generate-M-file запустите генерацию М-файла html-шаблона итогтаблицы pTabLinProgItogRuchTab. Проконтролируйте появление в текущей папке файла buildpLinProgItogRuchTab.m и его же появление в окне Редактора-итогчика.

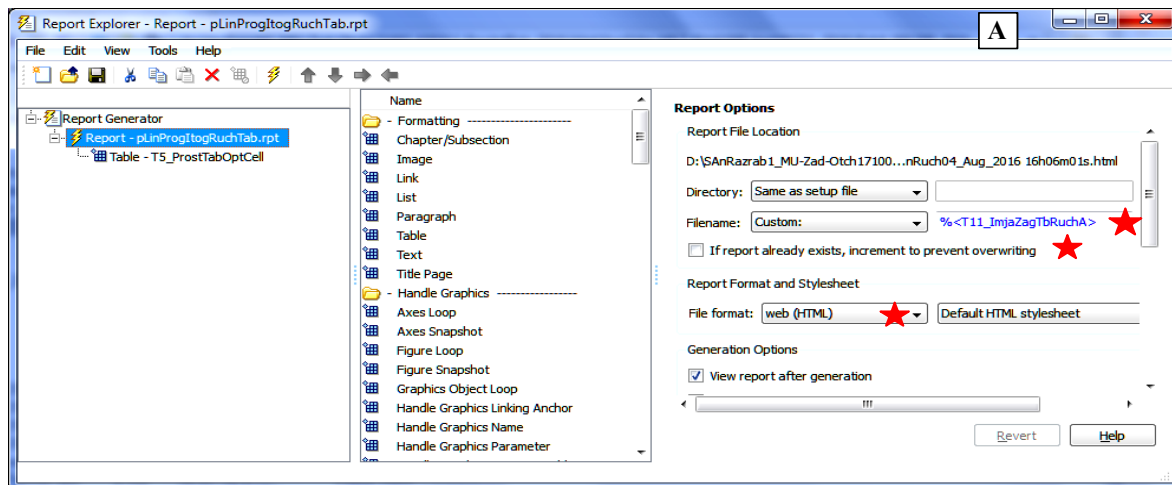
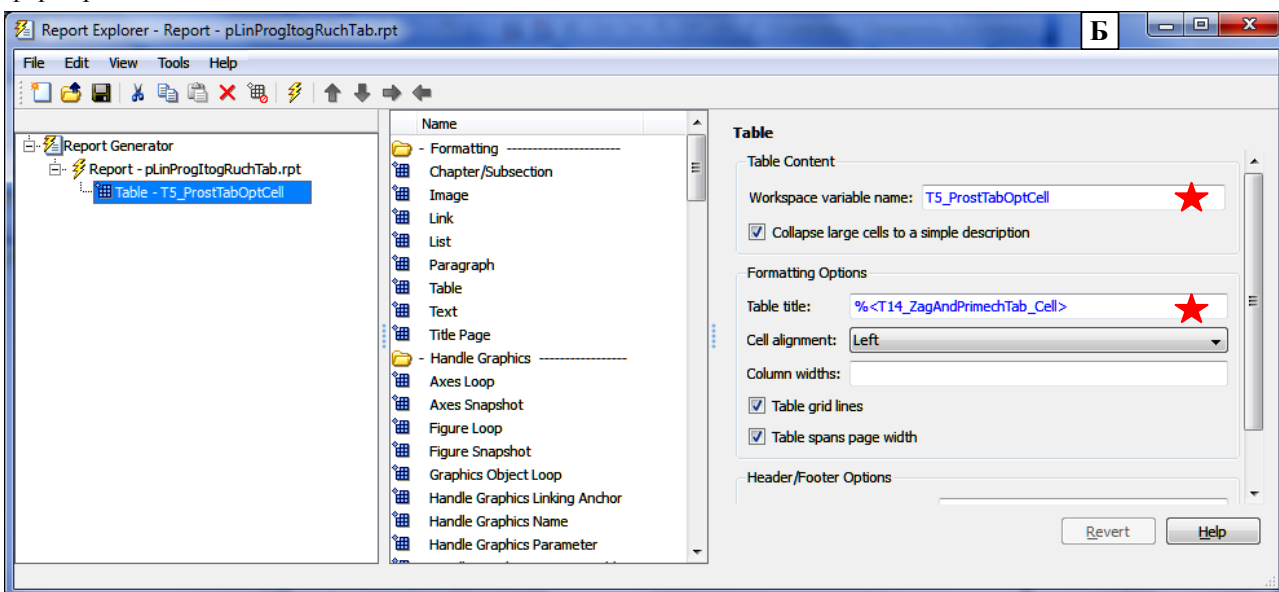


Рис. 3.11. Вид Репорт Эксплорера при выделении 1-й ветви в дереве Репорт Генератора (А) после сохранения с заданным именем pTabLinProgItogRuchTab.rpt и ввода настройки во вторую строку со звездочкой. В упомянутую строку вставляется командия файлов итогтаблиц, создаваемое в строке «T11_...» Построителя итогтаблиц рис.3.10. Продолжение рис.3.11 – Б.

3. Нажатием кнопки «Report»  на панели инструментов запускается Генератор отчетов на преобразование итог-таблицы из ячк- в html-формат: в действующей папке появятся файл со структурой имени типа «Таблица 1. pLinProgMaxRuch03_Aug_2016 23h55m45s». Одновременно появится Окно сообщения о ходе преобразования и html-итогтаблица вида рис. 3.5. Настройка шаблона и Генератора отчетов для итогтаблиц р-линпрогоптимизаторов закончена.

И. В завершение настройки следует запустить оба р-линпрогоптимизатора для проверки правильности формирования итогтаблиц.



Продолж. рис. 3.11. Настройки во второй ветви Репорт Генератора: отмечены звёздочками. 1-я вставка – командия из строки Построителя рис. 3.10. Вставка 2 – заголовок итогтаблицы и примечание к нему в угловых скобках с начальным знаком процентов.

3.6. Универсальный р-линпрогоптимизатор pLinProgTabGenOptimizer

3.6.1. Описание

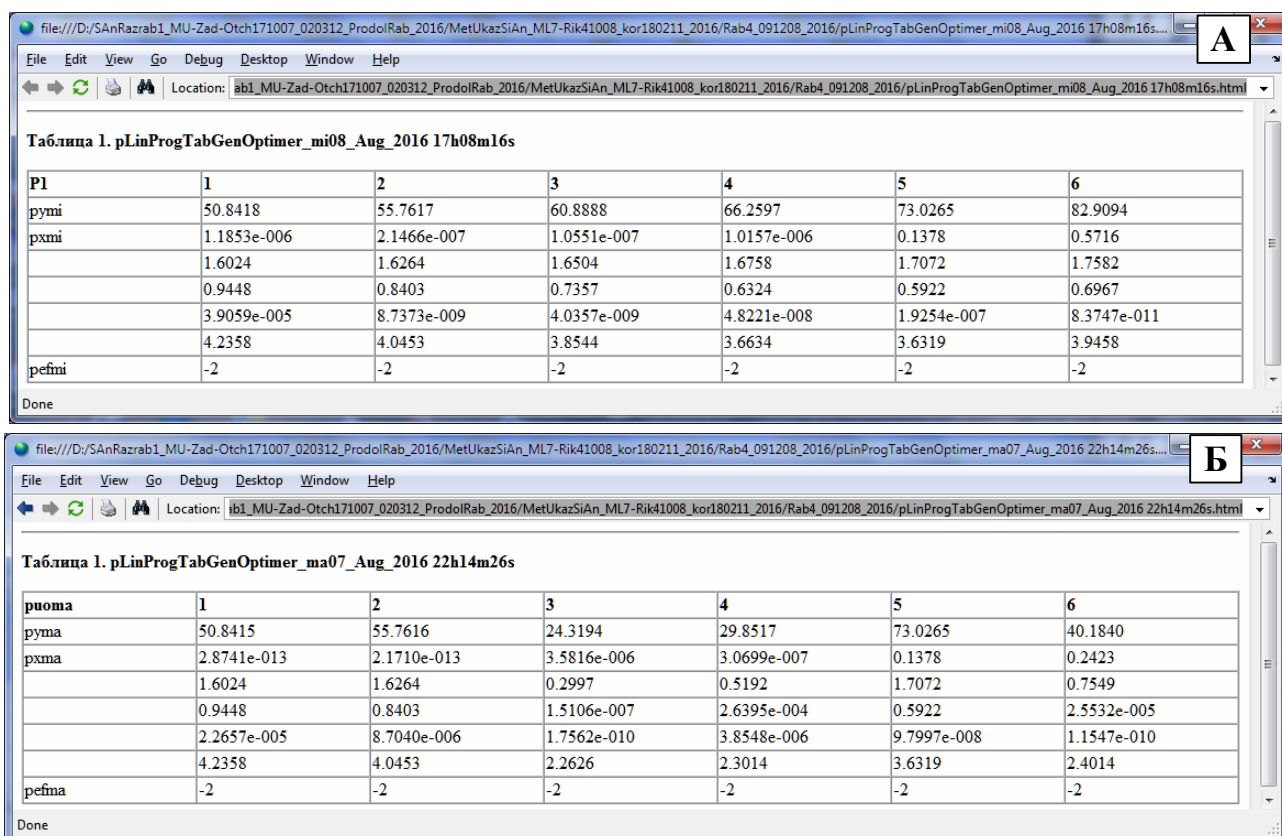
В универсальном р-оптимизаторе имена переменных носят шаблонный вид и не отражают тип оптимизации. В Вычпространство эти переменные копируются оптимизатором вместе с генерируемыми оптимизатором выходными переменными (рх, ру, реф, рnr, рrv), отражающим тип оптимизации (например, рхmi/ рхma). В этом оптимизаторе одни и те же условия можно использовать для вычисления р-минимума и р-максимума путём переключения команд (pLmi_pLma=1, min /-1, max). При производстве ряда (или массовых) р-линпрогоптимизаций преимущество универсального перед ручным в отношении трудоёмкости многократно возрастает. Программа оптимизатора представлена на рис. 3.12. Итогтаблицы, сформированные им – на рис. 3.13.

```
% Универсальный однопараметрический табгенерирующий линпрогоптимизатор pLinProgTabGenOptimizer
% Все переменные указания начинаются с буквы р для быстрого их отличия в Вычпространстве от переменных других линпро-
% указаний.
% Все переменные указания заканчиваются буквой о (оптимизационные). При запуске все переменные записываются в Вычпро-
% странство.
%; Кроме того, значения всех переменных получают новые имена через линпрогоптимизатор в Вычпространстве, отражающие
% вид оптимизации.
% Настройки универсального табличного линпрогоптимизатора с параметром pLinProgUnivesAvtomat_Tab
tic
рхo=[];руo=[];реfo=[];рnrо=[];рrvо=[];% Очистка переменных предыдущего запуска.
рхmi=[];руmi=[];реfmi=[];рnrmi=[];рrvmi=[]; рхma=[];руma=[];реfma=[];рnrma=[];рrvma=[];
pLmi_pLma=1; % Поиск линпрогмина. Закоммент. при поиске линпрогмакса.
% pLmi_pLma=-1; % Поиск линпрогмакса. Закоммент. при поиске линпрогмина.
% options=optimset('param1',число,'param2','строк-значение',...); % Изменение настроек линпрогфункции Матлаба linprog(...):
% раскомментировать строку и заполнить шаблон optimset(...).
% Аргументы линпрогогфункции Матлаба linprog(...)
ра=pLmi_pLma;
пуo=[1 2 3 4 5 6];
рnrо=[]; рrvо=[]; % Очистка выходных объектов от результата предыдущего вычисления.
for k=1:length(пуo)
    pfo = [-5.6 4.4*пуo(k) 6 -3 9]; % Вектор коэффициентов ц-функции.
    рAo = [1 -1 1 2 3; 3 2*пуo(k) 4 1 -2; 3 2 0 4 3; 1 3 5 4 -2; 3 4 7 1 -2];рbo = [5.71 2.3 17 6.5 11];
    % рAo=[]; рbo=[]; % Если огранич.-неравенства не используются. то раскомментировать строку и зкоммент. данные с рAo,
    рbo.
    рAeqo = [0 1.5 0 0 2; 3 1 2.5 0 0.7; 4.1 -2.3 11 3.4 4.7]; рbeqo = [7.75 3.58*0.1*пуo(k) 7.6];
    % рAeqo=[]; рbeqo=[]; % Если огранич.-равенства не используются. то раскомментировать строку и зкоммент. данные с
    рAeqo, рbeqo.
    pLBo = zeros(1,5);pUBo = [6.5 4.5 5 8.2 3.1];
    [хо,уo,efo]= linprog(ра*pfo,рAo,рbo,рAeqo,рbeqo,pLBo,pUBo); % Вычисление линпрогоптимума.
    рхo(:,k)=хо; руo(k)=ра*уo; реfo(k)=efo;
    % Копирование в Вычпространство с переименованием выходных переменных по типу оптимизации (мин/ макс).
    if pLmi_pLma==1
        рхmi=рхo; руmi=руo; реfmi=реfo;руomi=руo; % При вычислении р-мин.
    else
        рхma=рхo; руma=руo; реfma=реfo; руоma=руo;% При вычислении р-макс.
    end
    % Вычисление выходных объектов: векторов огранич.-неравенств и огранич.-равенств.

    if isempty(рAo) % Проверка пустого значения матрицы Ao.
        рnrо(k)=[];
    else
        рnrо(:,k)=[рAo*хо, рbo'];
    end
    if isempty(рAeqo) % Проверка пустого значения матрицы Aeqo.
        рrvо(k)=[];
    else
        рrvо(:,k)=[рAeqo*хо, рbeqo'];
    end
end
end
рхo, руo, реfo
% Запуск р-линпрогтабформатора
% pLinProgUnverOfldadTab % Текст-файл формирователя р-отладтаблицы линпрогоптимизатора.
pLinProgUnverP1TabStroyker % Текст-файл строителя р-итогтабл. и копировщика файла этого оптимизатора с именем файла
табл.
toc
```

Рис. 3.12. Универсальный р-линпрогоптимизатор pLinProgTabGenOptimizer. На рис. 3.13 показаны итогтаблицы, сформированные оптимизатором. В состав его входят 2 текстфайла: формирователя отладтаб-

лицы pLinProgUnverOtlad и построителя итогтаблицы pLinProgUnverP1TabStroyker с копированием линпрогоптимизатора (сопряжённый файл) в действующую папку с именем заголовка итогтаблицы.



А

file:///D:/SAnRazrab1_MU-Zad-Otch171007_020312_ProdolRab_2016/MetUkazSiAn_ML7-Rik41008_kor180211_2016/Rab4_091208_2016/pLinProgTabGenOptimizer_mi08_Aug_2016 17h08m16s.html

Location: ib1_MU-Zad-Otch171007_020312_ProdolRab_2016/MetUkazSiAn_ML7-Rik41008_kor180211_2016/Rab4_091208_2016/pLinProgTabGenOptimizer_mi08_Aug_2016 17h08m16s.html

Таблица 1. pLinProgTabGenOptimizer_mi08_Aug_2016 17h08m16s

P1	1	2	3	4	5	6
руми	50.8418	55.7617	60.8888	66.2597	73.0265	82.9094
рхми	1.1853e-006	2.1466e-007	1.0551e-007	1.0157e-006	0.1378	0.5716
	1.6024	1.6264	1.6504	1.6758	1.7072	1.7582
	0.9448	0.8403	0.7357	0.6324	0.5922	0.6967
	3.9059e-005	8.7373e-009	4.0357e-009	4.8221e-008	1.9254e-007	8.3747e-011
	4.2358	4.0453	3.8544	3.6634	3.6319	3.9458
pefmi	-2	-2	-2	-2	-2	-2

Done

Б

file:///D:/SAnRazrab1_MU-Zad-Otch171007_020312_ProdolRab_2016/MetUkazSiAn_ML7-Rik41008_kor180211_2016/Rab4_091208_2016/pLinProgTabGenOptimizer_ma07_Aug_2016 22h14m26s.html

Location: ib1_MU-Zad-Otch171007_020312_ProdolRab_2016/MetUkazSiAn_ML7-Rik41008_kor180211_2016/Rab4_091208_2016/pLinProgTabGenOptimizer_ma07_Aug_2016 22h14m26s.html

Таблица 1. pLinProgTabGenOptimizer_ma07_Aug_2016 22h14m26s

руома	1	2	3	4	5	6
рума	50.8415	55.7616	24.3194	29.8517	73.0265	40.1840
рхма	2.8741e-013	2.1710e-013	3.5816e-006	3.0699e-007	0.1378	0.2423
	1.6024	1.6264	0.2997	0.5192	1.7072	0.7549
	0.9448	0.8403	1.5106e-007	2.6395e-004	0.5922	2.5532e-005
	2.2657e-005	8.7040e-006	1.7562e-010	3.8548e-006	9.7997e-008	1.1547e-010
	4.2358	4.0453	2.2626	2.3014	3.6319	2.4014
pefma	-2	-2	-2	-2	-2	-2

Done

Рис. 3.13. Виды итогтаблиц, выдаваемых построителем pLinProgUnverP1TabStroyker из состава pLinProgTabGenOptimizer. Как видно из заголовка окна Интернет-просматривателя и его поля «Location» имя файла таблицы совпадает с её заголовком.

Построитель pLinProgUnverP1TabStroyker наряду с итогтаблицами копирует в действующую папку файл pLinProgTabGenOptimizer с именем файла итогтаблицы. Таким образом, в действующей папке формируется пара файлов с одинаковыми именами (сопряжённые файлы), но с разными расширениями: html – для файла итогтаблицы, m – для файла pLinProgTabGenOptimizer, создавшего эту таблицу – см. рис. 3.14. Сохранение файла оптимизатора, создавшего данную итогтаблицу, позволяет отказаться от сохранения в Вычпространстве переменных-условий: вместо сбора всех переменных из Вычпространства всегда можно вызвать файл оптимизатора со всеми этими переменными-условиями для повторного анализа результата и возможного исследования влияния изменения условий на результат.

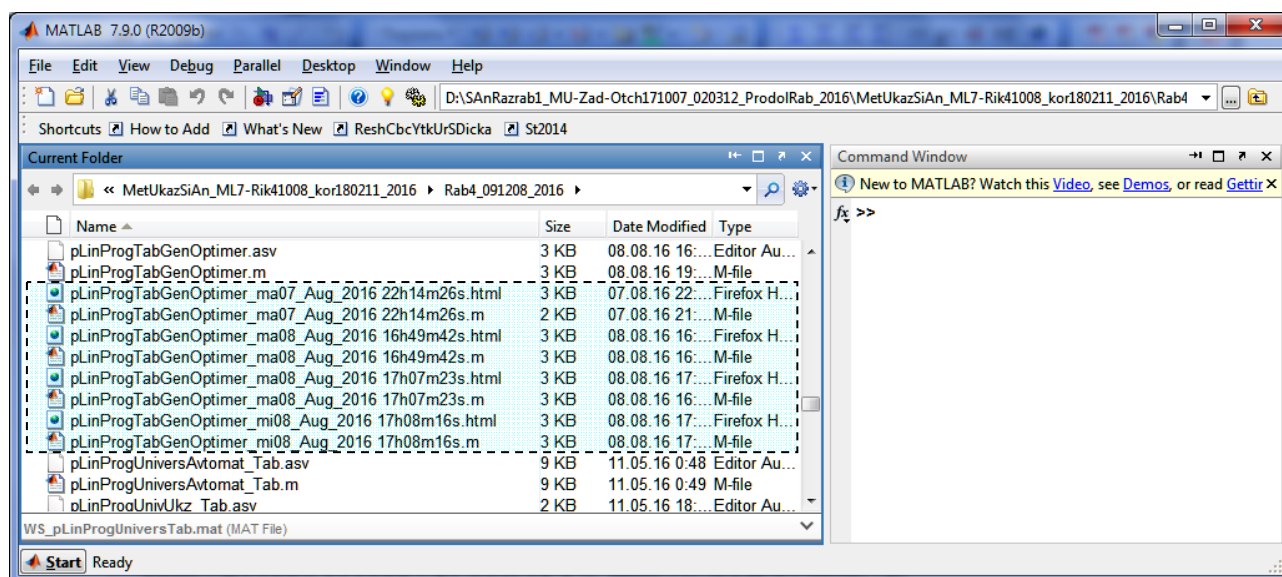


Рис. 3.14. Участок действующей папки Rab4_091208_2016 с 4 парами сопряженных файлов.

3.6.2. Построитель итогтаблиц pLinProgUnverP1TabStroyker

Пояснения к командам программы – в комментариях, а также – в Приложении 3. Настройка шаблона итогтаблиц – на рис. 3.16.

```
% Построитель итоговых таблиц р-линпрогоптимизации pLinProgUnverP1TabStroyker
% для универсального р-оптимизатора pLinProgTabGenOptimizer.m с копированием р-оптимизатора
% в действующую папку с именем заголовков соответствующей итогтаблицы.

if pLmi_pLma==1 % Формирование результата и имён выходных переменных р-линпрогминимизации .
    T1u_p1xyefmi=cat(1,puomi,pymi,pxmi,pefmi); % Вертикальное склеивание вых. данных линпрогминимизации в таблицу.
    T2u_p1xyefmiCell=num2cell(T1u_p1xyefmi); % Преобразование числовой табл.Т1 в ячейковый формат (ячк-формат)
    T3u_stbXmi_Cell(1:(length(pxmi)-1),1)={' '}; % создание ячк-столбца пустых строк в кол., равном длине вектора pxmi
    T4u_stbNameTabMinCell=['P1';'pymi';'pxmi';T3u_stbXmi_Cell; 'pefmi']; % создание ячк-столбца с именами строк табл.
ИтогP1МинТаб.
    T5u_ProstTabOptCell=cat(2,T4u_stbNameTabMinCell,T2u_p1xyefmiCell); % Данные для табл. ИтогP1МинТаб в ячк-формате.

elseif pLmi_pLma==1 % Формирование результата и имён выходных переменных р-линпрогмаксимизации.
    T1u_p1xyefma=cat(1,puoma,pyma,pxma,pefma); % Вертикальное склеивание вых. данных линпрогминимизации в таблицу.
    T2u_p1xyefmaCell=num2cell(T1u_p1xyefma); % Преобразование числовой табл.Т1 в ячейковый формат (ячк-формат)
    T3u_stbXma_Cell(1:(length(pxma)-1),1)={' '}; % создание ячк-столбца пустых строк в кол., равном длине вектора pxma-1
    T4u_stbNameTabMaxCell=['puoma';'pyma';'pxma';T3u_stbXma_Cell; 'pefma']; % создание ячк-столбца с именами строк табл.
ИтогP1МаксТаб.
    T5u_ProstTabOptCell=cat(2,T4u_stbNameTabMaxCell,T2u_p1xyefmaCell); % Данные для табл. ИтогP1МаксТаб в ячк-формате.
end
% Формирование заголовка и имени файла в html-формате итогтаблицы р-оптимизатора pLinProgTabGenOptimizer.m.
T6u_SumDatKor=datestr(clock); T6u_SumDatKor(3)='_'; T6u_SumDatKor(7)='_'; T6u_SumDatKor(15)='h';
T6u_SumDatKor(18)='m'; T6u_SumDatKor(21)='s'; % Дата и время запуска линпрогоптимизатора.
if pLmi_pLma==1 % Собственное имя итогтаблицы.
    T9u_ImjaZagTbUnver1=char('pLinProgTabGenOptimizer_mi');
    T11u_ImjaZagTbUnver=[T9u_ImjaZagTbUnver1 T6u_SumDatKor]; % Загол. итогтаблицы р-линпрогминимизации.
    copyfile('pLinProgTabGenOptimizer.m', [T11u_ImjaZagTbUnver '.m'],'f') % Копирование ф. pLinProgTabGenOptimizer.m с именем
заг. итогтаб. р-линпрогмин.
elseif pLmi_pLma==1
    T9u_ImjaZagTbUnver1=char('pLinProgTabGenOptimizer_ma');
    T11u_ImjaZagTbUnver=[T9u_ImjaZagTbUnver1 T6u_SumDatKor]; % Загол. итогтаблицы р-линпрогмаксимизации.
    copyfile('pLinProgTabGenOptimizer.m', [T11u_ImjaZagTbUnver '.m'],'f') % Копирование ф. pLinProgTabGenOptimizer.m с именем
заг. итогтаб. р-линпрогмакс.
end
report pLinProgItogUnverTab.rpt % Выдача итогтаблицы в html-формате. Работает только после создания и настройки
% html-шаблона итогтаблицы на основе генератора отчетов с именем LinProgItogUnverTab.rpt. Иначе выдаёт ошибку.
```

Рис. 3.15. Программа построителя файла итогтаблицы и сопряженного ему файла р-оптимизатора. Используемые в программе операторы и функции Матлаба описаны в Приложении 3.

Настройка шаблона итогтаблиц производится аналогично разд. 3.5.3 с заменой имени файла шаблона и значений полей вставки, указанных на рис. 3.16. Имя шаблона: pLinProgItogUnverTab.rpt.

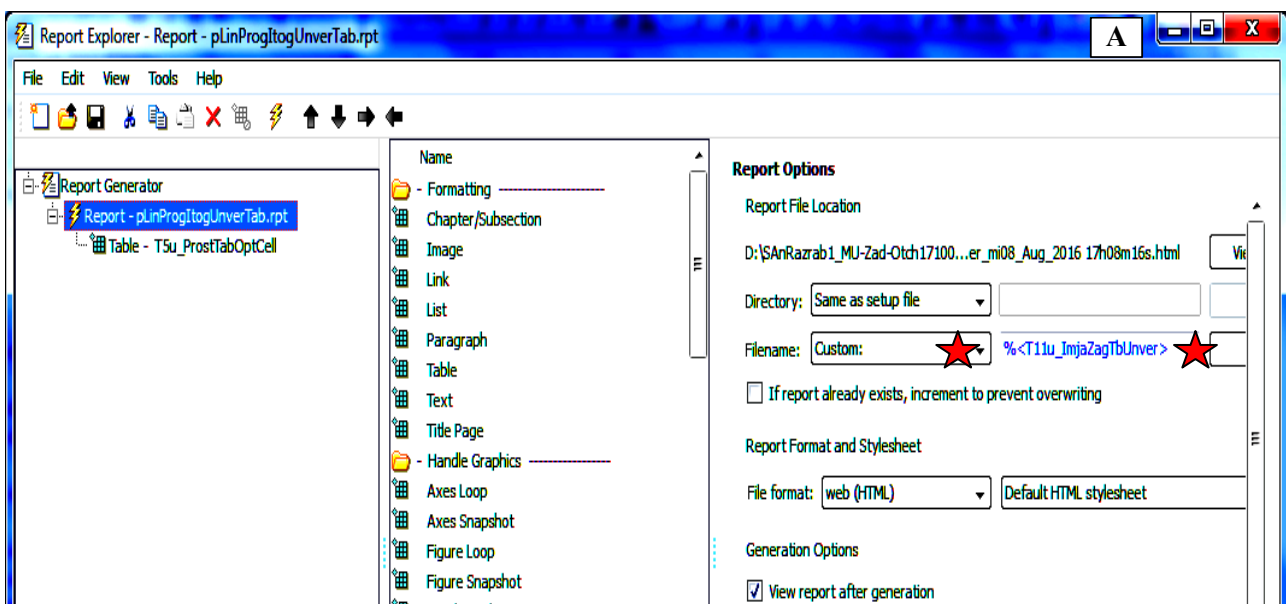


Рис. 3.16. Настройка полей Репортэксплорера (отмечены звёздочки) шаблона итогтаблиц при выделенной 1-й ветви в дереве Репортгенератора. В поле 2-й звёздочки вставляется в угловые скобки с предшествующим знаком % имя команды T11u_ImjaZagTbUnver, которая создаст имя файла итогтаблицы,

например, pLinProgTabGenOptimizer_mi08_Aug_2016 17h08m16s.html. Рис. 3.16, Б – на следующей странице.

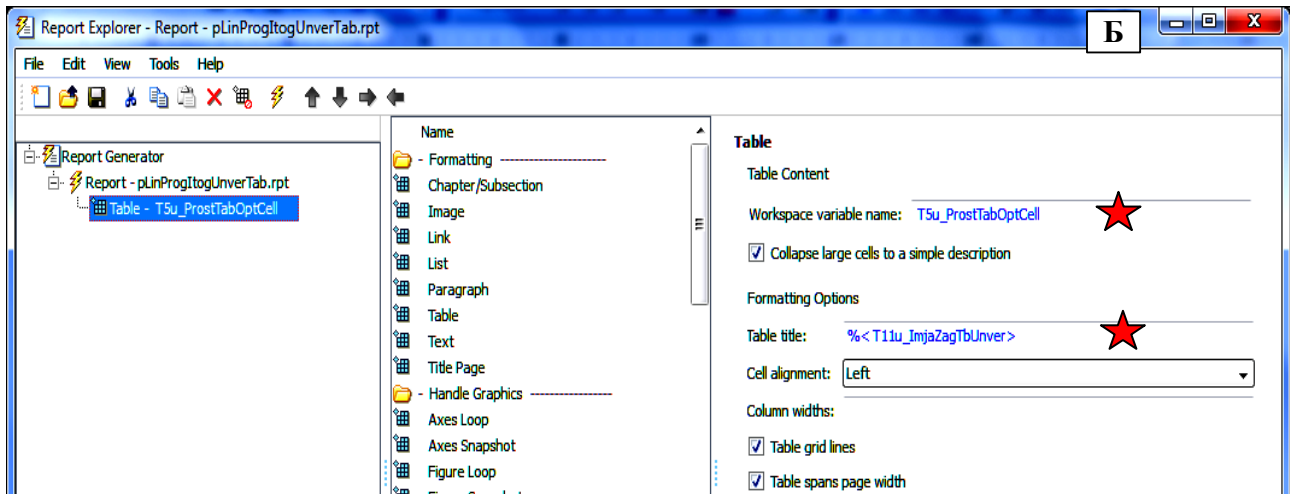


Рис. 3.16, Б. Страница Репортэксплорера шаблона итогтаблицы при выделении 2-й ветви в дереве Репортгенератора. Поля для вставки объектов из построителя pLinProgUnverP1TabStroyker: имени переменной содержания итогтаблицы в ячкформате T5u_ProstTabOptCell и команды формирования заголовка отладтаблицы T11u_ImjaZagTbUnver, совпадающей с командой формирования имени файла отладтаблицы – см. рис. 3.16, А.

3.6.3. Формирователь отладтаблиц pLinProgUnverOtladTab

Программа формирователя – на рис. 3.17. Пояснения к командам – в комментариях. Описание операторов и функций Матлаба, используемых в формирователе, – в Приложении 3. Формирователь входит в состав универсального р-линпрогоптимизатора pLinProgTabGenOptimizer (см. рис. 3.12). Пример сформированной отладтаблицы – на рис. 3.18. Настройки шаблона отладтаблицы в Репортэксплорере – на рис. 3.19.

```
% Форм ровыватель отладочной html-таблицы pLinProgUnverOtladTab для отладки
% исходных данных в р-оптимизаторе pLinProgTabGenOptimizer
po=[];T1_po2u=[]; T2_pyo2u=[];T3_pefo2u=[];T4_pnroZagCelu={};T5_prvoZagCelu={};T6_pnroKlayU=[];
T7_prvoKlayU=[]; % Очистка
T10_pnroKlayCelU={}; T11_prvoKlayCelU={};T12_TabOtladSodergCelU={};T14_stbLthAoCelU={}; % pe-
зультатов предыдущего
T15_stbLthAeqoCelU={};T18_TabOtladStbImenCelU={};T19_pLinPrTabOtladCelU={}; % запуска по-
строителя отладтаблицы.
if pLmi_pLma==1 % Формирование имён переменных, значений условий и части результ. линпрог-
минимизатора.
    po=puo;
    Sod_T4_1u={'pAo*pxmi(puo)'};
    Sod_T4_2u={'pbo'};
    Sod_T5_1u={'pAeqo*pxmi(puo)'};
    Sod_T5_2u={'pbeqo'};
    Sod_T13u={'puo';'pyumi';'pefmi';};
    Sod_T16u={'pAo*pxmi pbo'};
    Sod_T17u={'pAeqo*pxmi pbeqo'};
elseif pLmi_pLma==-1 % Формирование имён переменных, значений условий и части результ. лин-
прогмаксимизатора.
    po=puo
    Sod_T4_1u={'pAo*pxma(puo)'};
    Sod_T4_2u={'pbo'};
    Sod_T5_1u={'pAeqo*pxma(puo)'};
    Sod_T5_2u={'pbeqo'};
    Sod_T13u={'puo';'pyuma';'pefma';};
    Sod_T16u={'pAo*pxma pbo'};
    Sod_T17u={'pAeqo*pxmao pbeqo'};
end
for k3=1:length(po) % Создание строк ро, уо, ефо с удвоением количества их значений .
    T1_po2u(2*k3-1)=po(k3);
```

```

T1_po2u(2*k3)=po(k3);
T2_py02u(2*k3-1)=pyo(k3);
T2_py02u(2*k3)=pyo(k3);
T3_pfo2u(2*k3-1)=pfo(k3);
T3_pfo2u(2*k3)=pfo(k3);
T4_pnroZagCelu(2*k3-1)=Sod_T4_1u; % Формирование ячкстроки заголовков столбцов
% неравенств типа "pAo*pxma pbo".
T4_pnroZagCelu(2*k3)=Sod_T4_2u;
T5_prvoZagCelu(2*k3-1)=Sod_T5_1u;
T5_prvoZagCelu(2*k3)=Sod_T5_2u; % Ячкстрока заголовков столбцов неравенств.
if isempty(pAo) % Проверка пустого значения матрицы pAo.
    T6_pnroKlayU=[T6_pnroKlayU []];
else
    T6_pnroKlayU=[T6_pnroKlayU pnro(:,k3)];
end
if isempty(pAeqo) % Проверка пустого значения матрицы Aeqo.
    T7_prvoKlayU=[T7_prvoKlayU []];
else
    T7_prvoKlayU=[T7_prvoKlayU prvo(:,k3)];
end
end
T8_pxyefU=cat(1,T1_po2u, T2_py02u, T3_pfo2u); % Вертикальное склеивание строк с удвоен.
кол.значений.
T9_pyefCellU=num2cell( T8_pxyefU); % Перевод в ячк-формат (cell-формат) матрицы T8.
if isempty(pAo) % Проверка пустого значения матрицы pA. Формирование ячкзначений вых. об.
pnro.
    T10_pnroKlayCelU(1,1:2*length(po))={'[]'};
else
    T10_pnroKlayCelU=num2cell(T6_pnroKlayU);
end
if isempty(pAeqo) % Проверка пустого значения матрицы pAeqo. Формирование ячкзначений вых. об.
prvo.
    T11_prvoKlayCelU(1,1:2*length(po))={'[]'};
else
    T11_prvoKlayCelU=num2cell(T7_prvoKlayU);
end
% Матрица значений отладтаблицы.
T12_OtladTabSodergCelU=cat(1,T9_pyefCellU,
T4_pnroZagCelu,T10_pnroKlayCelU,T5_prvoZagCelu,T11_prvoKlayCelU);
T13_StbImenCelU=Sod_T13u; % Столбец имён 3 верхних строк отладтаблицы в ячкформате.
if isempty(pAo) % Провер. пуст. знач. matr. Ao. Формир. столбца пуст. загол. для строк (компонент)
pnro.
    T14_stbLthAoCelU={' '}; % Столбец пустых ячеек заменяется одной пустой ячейкой.
else
    T14_stbLthAoCelU(1:length(pAo(:,1)),1)={' '}; % Столбец пустых ячеек по столбцу pAo(:,1).
end
if isempty(pAeqo) % Провер. пуст. знач. matr. Aeqo.Формир. столбца пуст. загол. для строк (компо-
нент) prvo.
    T15_stbLthAeqoCelU={' '}; % Столбец пустых ячеек заменяется одной пустой ячейкой.
else
    T15_stbLthAeqoCelU(1:length(pAeqo(:,1)),1)={' '}; % Столбец пустых ячеек по столбцу pAeqo(:,1).
end
T16_CelU=Sod_T16u;
T17_CelU=Sod_T17u;
% Столбец имён строк отладтаблицы в ячкформате.
T18_TabOtladStbImenCelU=cat(1, T13_StbImenCelU, T16_CelU, T14_stbLthAoCelU,
T17_CelU,T15_stbLthAeqoCelU);
T19_pLinPrTabOtladCelU=cat(2,T18_TabOtladStbImenCelU, T12_OtladTabSodergCelU); % Содержание
универ. отладтаблицы в ячкформате.
% Формирование заголовка и имени файла в html-формате отладтаблицы линпрогзапуска.
T20_SumDatKor=datestr(clock); T20_SumDatKor(3)='_';T20_SumDatKor(7)='_'; T20_SumDatKor(15)='h';
T20_SumDatKor(18)='m';T20_SumDatKor(21)='s'; % Дата и время запуска линпрогоптимизации.
if pLmi_pLma==1 % Собственное имя отладтаблицы.
    T21_ImjaTbU=char('pUnverOtladTab_linprogmin');
elseif pLmi_pLma==-1

```

```

T21_ImjaTbU=char('pUnverOtladTab_linprogmax');
end
ImFileandZagUnverOtladTabU=[T21_ImjaTbU blanks(2) T20_SumDatKor]; % Имя файла отладтаблицы
и её заголовка.
report pLinProgUnverOtladTab.rpt % Выдача отладтаблицы в html-формате. Работает только после соз-
дания и настройки
% html-шаблона отладтаблицы на основе генератора отчётов с именем pLinProgOtladTab.rpt. Иначе
выдаёт ошибку.

```

Рис. 3.17. Формирователь отладтаблиц для универсального р-линпрогоптимизатора pLinProgTabGenOptimizer.

Таблица 1. pUnverOtladTab_linprogmin 09_Aug_2016 19h32m12s

puo	1	1	2	2	3	3	4	4	5	5	6	6
pyumi	50.8418	50.8418	55.7617	55.7617	60.8888	60.8888	66.2597	66.2597	73.0265	73.0265	82.9094	82.9094
pefmi	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2
pAo*pxmi pbo	pAo*pxmi(puo)	pbo	pAo*pxmi(puo)	pbo	pAo*pxmi(puo)	pbo	pAo*pxmi(puo)	pbo	pAo*pxmi(puo)	pbo	pAo*pxmi(puo)	pbo
	12.0499	5.7100	11.3498	5.7100	10.6484	5.7100	9.9467	5.7100	9.9183	5.7100	11.3475	5.7100
	-1.4876	2.3000	1.7762	2.3000	5.1364	2.3000	8.6095	2.3000	12.5904	2.3000	17.7088	2.3000
	15.9125	17	15.3887	17	14.8640	17	14.3419	17	14.7233	17	17.0686	17
	1.0598	6.5000	0.9901	6.5000	0.9208	6.5000	0.8627	6.5000	0.9566	6.5000	1.4382	6.5000
	4.5517	11	4.2972	11	4.0425	11	3.8033	11	4.1238	11	5.7330	11
pAeqo*pxmi pbeqo	pAeqo*pxmi(puo)	pbeqo	pAeqo*pxmi(puo)	pbeqo	pAeqo*pxmi(puo)	pbeqo	pAeqo*pxmi(puo)	pbeqo	pAeqo*pxmi(puo)	pbeqo	pAeqo*pxmi(puo)	pbeqo
	10.8753	7.7500	10.5302	7.7500	10.1844	7.7500	9.8406	7.7500	9.8245	7.7500	10.5290	7.7500
	6.9295	0.3580	6.5589	0.7160	6.1876	1.0740	5.8212	1.4320	6.1433	1.7900	7.9768	2.1480
	26.6157	7.6000	24.5158	7.6000	22.4118	7.6000	20.3198	7.6000	20.2222	7.6000	24.5085	7.6000

Рис. 3.18. Вид отладтаблицы для р-линпрогоптимизатора рис. 3.12, настроенного на вычисление минимума.

Настройка шаблона отладтаблицы производится аналогично разд. 3.5.3 с заменой имени файла шаблона и значений полей вставки на указанные на рис. 3.19. Имя шаблона: pLinProgUnverOtladTab.

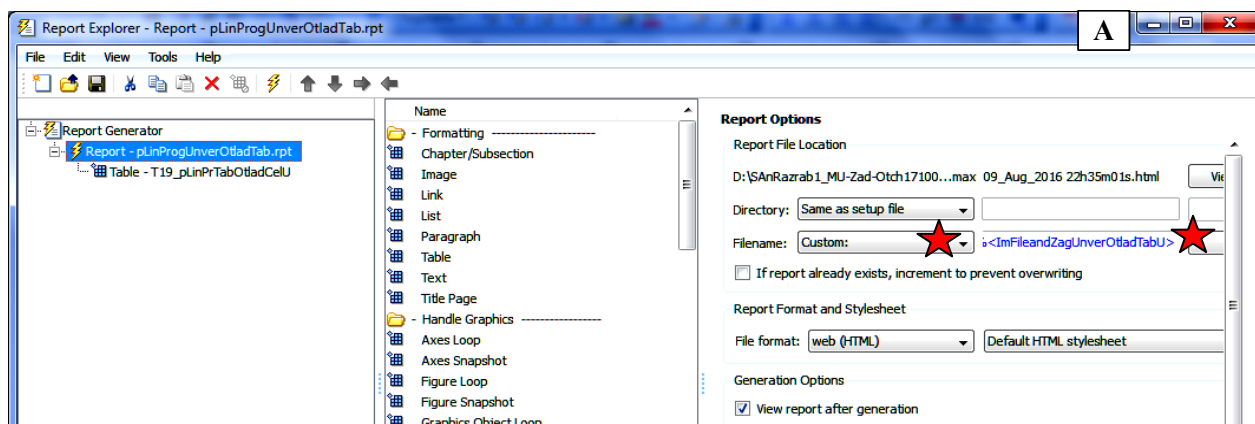
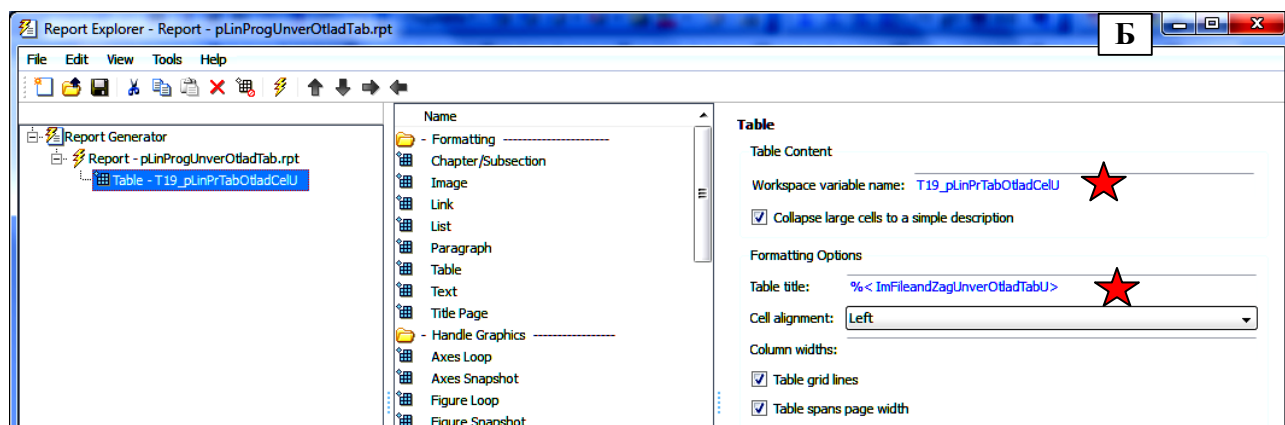


Рис. 3.19. Настройка полей Репортэксплорера (отмечены звёздочками) шаблона отладтаблицы при выделенной 1-й ветви в дереве Репортгенератора. В поле 2-й звёздочки вставляется в угловые скобки с предшествующим знаком % имя команды ImFileandZagUnverOtladTabU, которая создаст имя файла отладтаблицы, например, pUnverOtladTab_linprogmin 09_Aug_2016 19h32m12s.html – см. заголовок окна и поле «Location» на рис. 3.18. Продолжение рисунка – на следующей странице.



Продолжение рис. 3.19. Страница Репортэксплорера при выделении 2-й ветви в дереве Репортгенератора. Поля для вставки объектов из формователя pLinProgUnverOtladTab: имени переменной содержания отладтаблицы в ячкформате T19_pLinPrTabOtladCelU и команды формирования заголовка отладтаблицы ImFileandZagUnverOtladTabU, совпадающей с командой формирования имени файла отладтаблицы – см. рис. 3.19, А.

3.6.4. Пример использования формователя pLinProgUnverOtladTab для коррекции условий р-линпрогоптимизации

Рассмотрим пример коррекции условий р-линпрогминимизации с использованием отладтаблиц формователя pLinProgUnverOtladTab. В качестве исходной берём таблицу рис. 3.18, копируя её в рис. 3.20. Как видно из таблицы при всех значениях параметра пуо значения рефми равно -2, т.е. р-оптимизатор pLinProgTabGenOptimizer ни при каком значении параметра не нашёл линпрогминимум.

file:///D:/SanRazrab1_MU-Zad-Otch171007_020312_ProdolRab_2016/MetUkazSiAn_ML7-Rik41008_kor180211_2016/Rab4_091208_2016/pUnverOtladTab_linprogmin_09_Aug_2016 19h32m12s.html

Location: \\anRazrab1_MU-Zad-Otch171007_020312_ProdolRab_2016/MetUkazSiAn_ML7-Rik41008_kor180211_2016/Rab4_091208_2016/pUnverOtladTab_linprogmin_09_Aug_2016 19h32m12s.html

Таблица 1. pUnverOtladTab_linprogmin 09_Aug_2016 19h32m12s

пуо	1	1	2	2	3	3	4	4	5	5	6	6
пуо	50.8418	50.8418	55.7617	55.7617	60.8888	60.8888	66.2597	66.2597	73.0265	73.0265	82.9094	82.9094
рефми	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2
рАо*рхми	рАо*рхми(пуо)	рбо	рАо*рхми(пуо)	рбо	рАо*рхми(пуо)	рбо	рАо*рхми(пуо)	рбо	рАо*рхми(пуо)	рбо	рАо*рхми(пуо)	рбо
рбо	12.0499	5.7100	11.3498	5.7100	10.6484	5.7100	9.9467	5.7100	9.9183	5.7100	11.3475	5.7100
	-1.4876	2.3000	1.7762	2.3000	5.1364	2.3000	8.6095	2.3000	12.5904	2.3000	17.7088	2.3000
	15.9125	17	15.3887	17	14.8640	17	14.3419	17	14.7233	17	17.0686	17
	1.0598	6.5000	0.9901	6.5000	0.9208	6.5000	0.8627	6.5000	0.9566	6.5000	1.4382	6.5000
	4.5517	11	4.2972	11	4.0425	11	3.8033	11	4.1238	11	5.7330	11
рАео*рхми	рАео*рхми(пуо)	рбео	рАео*рхми(пуо)	рбео	рАео*рхми(пуо)	рбео	рАео*рхми(пуо)	рбео	рАео*рхми(пуо)	рбео	рАео*рхми(пуо)	рбео
рбео	10.8753	7.7500	10.5302	7.7500	10.1844	7.7500	9.8406	7.7500	9.8245	7.7500	10.5290	7.7500
	6.9295	0.3580	6.5589	0.7160	6.1876	1.0740	5.8212	1.4320	6.1433	1.7900	7.9768	2.1480
	26.6157	7.6000	24.5158	7.6000	22.4118	7.6000	20.3198	7.6000	20.2222	7.6000	24.5085	7.6000

Done

Рис. 3.20. Исходная отладтаблица при коррекции условий р-линпрогоптимизации с выделением значений в 3 строках величины рАо*рхми(пуо), максимально не удовлетворяющих неравенству (3.4).

Выполним коррекцию условий линпрогоптимизации согласно правилам разд. 3.4.1.1, предполагая, что в действующей папке находятся все необходимые текстфайлы Матлаба и шаблоны отладтаблиц и итогтаблиц: р-оптимизатор pLinProgTabGenOptimizer, формователь отладтаблицы pLinProgUnverOtladTab, построитель итогтаблицы pLinProgUnverP1TabStroyker, который также копирует в действующую папку файл р-оптимизатора, присваивая ему имя заголовка итогтаблицы; шаблон отладтаблицы pLinProgUnverOtladTab.rpt и сопутствующий ему файл buildpLinProgOtladTab.m, шаблон итогтаблицы pLinProgItogUnverTab.rpt и сопутствующий ему файл buildpLinProgItogUnverTab.m – всего 7 М-файлов.

А. В оптимизаторе pLinProgTabGenOptimizer (рис. 3.12) следует закомментировать строку с pLinProgUnverP1TabStroyker и раскомментировать строку с формователем pLinProgUnverOtladTab. Следует

запустить оптимизатор, который выдаст отладтаблицу типа рис. 3.20.

Б. Закройте эту таблицу. Выйдите из Матлаба, не закрывая его. Перейдите в действующую папку по системному пути и откройте файл отладтаблица посредством Уорда.

В. Просматриваем в полученной таблице строки комплекса в которых нарушено неравенство $p_{Ao} * p_{xmi}(p_{uo}) \leq p_{bo}$. (3.4)

В каждой такой строке выделяем инструментом Уорда (например, зелёным цветом) наибольшее значение величины $p_{Ao} * p_{xmi}(p_{uo})$ и пересохраняем файл таблицы. Таких строк 3 – см. рис. 3.20.

Г. Округляем в большую сторону до 0,1 выделенные величины. Добавляем к каждой по 0,1 и используем их для формирования нового вектора p_{bo} , у которого первые 3 значения сформированы по выделенным значениям, а остальные 2 – прежние. Таким образом, новое значение вектора p_{bo} следующее:

$p_{bo} = [12.2 \ 17.9 \ 17.2 \ 6.5 \ 11]$ (вместо $p_{bo} = [5.71 \ 2.3 \ 17 \ 6.5 \ 11]$). (3.5)

Д. В оптимизаторе `pLinProgTabGenOptimizer` закомментируем строку с прежним p_{bo} . На дополнительную строку вставляем новое значение p_{bo} из (3.5). Кроме того, закомментируем команду «`pLBo = zeros(1,5)`» и вставим вместо неё новое значение вектора нижних границ α -области «`pLBo = -2 * ones(1,5)`». Запускаем оптимизатор, выдающий новую отладтаблицу – рис. 3.21. Как видно из таблицы, оптимальные решения найдены для всех значений параметра p_{uo} .

Рис. 3.21. Отладтаблица 1-й итерации коррекции – см. п.Д.

Е. Оценка разности α -функций решения (таблица рис. 3.21) и исходной (рис. 3.20), когда решения не было найдено. Процедура оценки отражена в табл. 3.6. Для оценки разности следует скопировать первые 3 строки из исходной (рис. 3.20) и последней (рис. 3.21) отладтаблиц в пустой Уорд-документ, удалить повторяющиеся столбцы, вставить 2 допстроки с именами из табл. 3.22, скопировать строки `p_yimi_res` и `p_yimi_ishod` в сесфайл и сформировать из них векторы. Скопировать также имена 2 последних строк табл. 3.6, как команд Матлаба (точка перед знаком деления в 5-й строке обязательна), выделить все скопированное и запустить вычисления. Результаты вычислений копируются в 4-ю и 5-ю строки Уорд-таблицы.

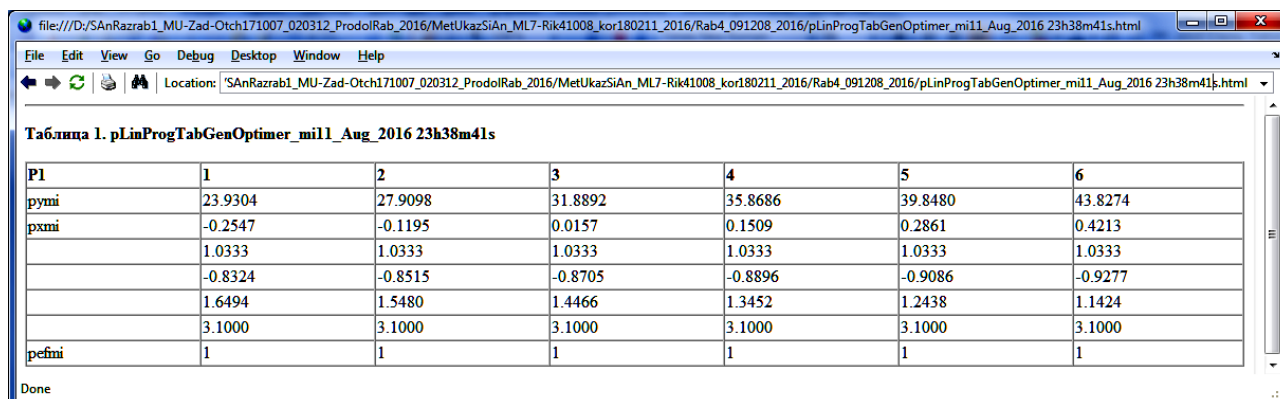
Таблица 3.22. Оценка разности α -функций решения (табл. `pUnverOtladTab_linprogmin 11_Aug_2016 22h30m22s`, рис. 3.21) и исходной (`pUnverOtladTab_linprogmin 09_Aug_2016 19h32m12s`, рис. 3.20).

p_{uo}	1	2	3	4	5	6
<code>p_yimi_res</code>	23.9304	27.9098	31.8892	35.8686	39.8480	43.8274
<code>p_yimi_ishod</code>	50.8418	55.7617	60.8888	66.2597	73.0265	82.9094
<code>D=p_yimi_res- p_yimi_ishod</code>	-26.9114	-27.8519	-28.9996	-30.3911	-33.1785	-39.0820
<code>Otnos=D*100./ p_yimi_res,</code> %	-112.4570	-99.7925	-90.9386	-84.7290	-83.2626	-89.1725

Как следует из табл. 3.22, найденное решение уточняет значения α -функции из табл. 3.20 по абсолютной величине в диапазоне от 83 до 112 %, т.е. с уменьшением в 2 раза.

Ж. В линпрогоптимизаторе pLinProgTabGenOptimizer закомментируйте строку с формирователем отлад-таблицы pLinProgUnverOtladTab и раскомментируйте строку с построителем итогтаблицы pLinProgUnverP1TabStroyker. Запустите линпрогоптимизатор: в результате получите итогтаблицу типа рис. 3.22 и копию файл этого линпрогоптимизатора, сохраненную в действующей папке с именем заголовка полученной итогтаблицы (для таблицы рис. 3.22 - pLinProgTabGenOptimizer_mi11_Aug_2016 23h38m41s.m.).

Отладка условий р-линпрогоптимизации и сама линпрогоптимизация выполнены успешно.



P1	1	2	3	4	5	6
rymi	23.9304	27.9098	31.8892	35.8686	39.8480	43.8274
pxmi	-0.2547	-0.1195	0.0157	0.1509	0.2861	0.4213
	1.0333	1.0333	1.0333	1.0333	1.0333	1.0333
	-0.8324	-0.8515	-0.8705	-0.8896	-0.9086	-0.9277
	1.6494	1.5480	1.4466	1.3452	1.2438	1.1424
	3.1000	3.1000	3.1000	3.1000	3.1000	3.1000
pefmi	1	1	1	1	1	1

Рис. 3.22. Итогтаблица р-линпрогоптимизации, полученная после коррекции условий оптимизации в п. Д.

3.6.5. Процедура применения универсального р-линпрогоптимизатора pLinProgTabGenOptimizer

А. Запускается Матлаб. Создаётся действующая папка с подходящим именем (например, pLinProgUnverOptimizer). В действующей папке следует создать все необходимые текстфайлы Матлаба и шаблоны отладтаблиц и итогтаблиц: р-оптимизатор pLinProgTabGenOptimizer (рис. 3.12), формирователь отладтаблицы pLinProgUnverOtladTab (рис. 3.17), построитель итогтаблицы pLinProgUnverP1TabStroyker (рис. 3.15), который также копирует в действующую папку файл р-оптимизатора, присваивая ему имя заголовка итогтаблицы; шаблон отладтаблицы pLinProgUnverOtladTab.rpt (рис. 3.19) и сопутствующий ему файл buildpLinProgOtladTab.m, шаблон итогтаблицы pLinProgItogUnverTab.rpt (рис. 3.16) и сопутствующий ему файл buildpLinProgItogUnverTab.m – всего 7 М-файлов.

При наличии электронной версии монографии указанные на рис. 3.12, 3.17 и 3.15 программы можно скопировать в отдельные текстфайлы Редактора с присвоением им упомянутых соответствующих имён.

Б. Создание и настройка шаблона отладтаблицы pLinProgUnverOtladTab.rpt производится аналогично разд. 3.5.3 с заменой имени файла шаблона и значений полей вставки, указанных на рис. 3.19. Сопутствующий шаблону файл buildpLinProgOtladTab.m создаётся после настройки шаблона командами меню Репортэксplorер: «File_Generate M-File...». Аналогично создаётся и настраивается шаблон итогтаблицы pLinProgItogUnverTab.rpt (значения полей вставки – на рис. 3.16) и сопутствующий файл buildpLinProgItogUnverTab.m.

В. Раскомментируйте в р-оптимизаторе pLinProgTabGenOptimizer строки с файлами формирователя отладтаблиц pLinProgUnverOtladTab и построителя итогтаблиц pLinProgUnverP1TabStroyker. Не меняя содержания оптимизатора запустите его на поиск минимума, а потом – максимума. Оптимизатор должен выдать 4 html-таблицы: 2 отладтаблицы и 2 итогтаблицы. Если не выдаёт – проверьте и исправьте выполненную в п. Б настройку шаблонов соответствующих таблиц.

Г. Вставьте в оптимизатор вместо существующих данные решаемой задачи. Установите значение триггеркоманды, соответствующее требуемому типу оптимизации: pLmi_pLma=1, минимум/-1, максимум). Противоположное значение закомментируйте. Закомментируйте строку с формирователем отладтаблицы pLinProgUnverOtladTab.

Д. Запустите оптимизатор – он выдаст итогтаблицу типа рис. 3.22 или 3.13 или промежуточную между двумя указанными, в которой для части значений параметра рио флаг оптимизации (например, pefmi) равен 1, а

ждя остальных равен -2. В 1-м случае р-линпрогоптимизация выполнена. Для остальных случаев следует использовать коррекцию условий оптимизации с использованием отладтаблиц, создаваемых формирователем pLinProgUnverOtladTab.

Е. Коррекция условий р-линпрогоптимизации выполняется согласно разд. 3.6.4 или в несколько другом варианте действий – согласно разд. 3.4.1.1 и 3.4.1.2. Заканчиваться коррекция должна получением отладтаблицы, в которой $pfmi/ pfma = 1$ для всех значений параметра ρ_{io} , а также вычислением разности значений ц-функций для исходной отладтаблицы и откорректированной согласно п. Е разд. 3.6.4 и получением итогтаблицы р-линпрогоптимизации с откорректированными условиями оптимизации.

Ж. При решении ряда или множества задач р-линпрогоптимизации нужно иметь ввиду следующее. В Вычпространстве значения всех переменных р-линпрогоптимизатора pLinProgTabGenOptimizer сохраняются только для последнего запуска: очередной запуск меняет все их значения. После каждого запуска оптимизатор в действующей папке создаёт 2 файла с именем созданной итогтаблицы: html-файл итогтаблицы и копию m-файла самого оптимизатора. Таким образом, условия получения итогтаблицы сохраняются в m-файле до тех пор, пока исполнитель не удалит этот файл. Это означает, что Исполнитель может менять условия р-линпрогозадач, полагаясь на автоматическое их сохранение для последующего анализа результатов р-оптимизации множества задач.

4. Задание 4. Решение примеров и задачи линейного программирования посредством универсальных линпрогооптимизаторов

Применение универсальных линпрогуказаний `LinProgUnivUkz_Tab`, `LinProgUnivUkz` и линпрогооптимизаторов `LinProgUnivesAvtomatMU_Tab`, `LinProgUnivesAvtomatMU` для решения примеров линпрогооптимизации небольшой и крупной размерностей. Вычисление зависимости линпрогооптимумов от одного параметра p с использованием универсального p -линпрогооптимизатора `pLinProgTabGenOptimizer` и построителя итогтаблиц `pLinProgUnverP1TabStroyker`. Коррекция условий p -линпрогооптимизации посредством формирователя отладтаблиц `pLinProgUnverOtladTab` в случае частичного или полного отсутствия решения. Построение графика зависимости π -функции от параметра p . Самостоятельное решение тематической линпрогозадачи из разд 7.

4.1. Применение универсальных линпрогуказания `LinProgUnivUkz_Tab` и линпрогооптимизатора `LinProgUnivesAvtomatMU_Tab` для решения примеров линпрогооптимизации с размерностью 5...10

1. Создайте на диске D (не системном) папку «Rab4Familija(vasha)», в которую следует помещать все файлы, связанные с выполнением Задания 4 по Системному анализу. Создайте в папке Rab4 сессионный файл (сессфайл) «R4SessFamilija(vasha)N(N – вар.зад.4)», мат-файл «WS_R4Familija(vasha)N.mat».

2. Откройте Матлаб. В поле «Current Directory» вставьте полное имя папки Rab4, используя кнопку поиска папки . Создайте объекты среды для выполнения работы 4.

- Файл «WS_R4Familija(vasha)N.mat» (файл Вычпространства работы 4): после открытия папки Rab4 откройте меню «Desktop» и поставьте галочку в строке «Workspace». В открывшемся окне Вычпространства командами меню «File_Save Workspace As» сохраните пустой файл с указанным именем.

- Сессионный файл «R4SessFamilija(vasha)N(N – вар.зад.4)» путем открытия нового М-файла в окне Редактора и сохранения его в папке Rab4 командами File-Save_As.

3. В 1-й позиции 1-й строки сессфайла поставьте знак % и введите заголовок «1. Дата (вставьте дату начала исполнения задания). Примеры вычисления линпрогооптимумов посредством универсальных линпрогуказания `LinProgUnivUkz_Tab` и итогтабличного линпрогооптимизатора `LinProgUnivesAvtomatMU_Tab`. Вар. N (номер задания)». «1.1. Линпрогмаксимум при наличии всех ограничений».

4. Ознакомьтесь со свойствами матлабовской функции линейного программирования `linprog(...)` (разд. 2.2). Изучите описание Универсального линпрогооптимизатора (линпрогуказание и сам оптимизатор) – разд. 2.4.4.

5. Скопируйте в вашу папку из электронной версии монографии (см. конец разд. 2.2.3) программы: линпрогуказания `LinProgUnivUkz_Tab` (рис. 2.12) и итогтабличного линпрогооптимизатора `LinProgUnivesAvtomatMU_Tab` (рис. 2.13, А). Каждую в отдельный текстфайл с упомянутым именем. Запустите линпрогуказание `LinProgUnivUkz_Tab`: Матлаб выдаст в Комокно результат поиска минимума с сообщением об ошибке - файл шаблона итогтаблицы линпрогминимума `ItoTabLPrmi.rpt` не обнаружен.

6. Создайте шаблон итогтаблицы линпрогминимума согласно разд. 2.4.4.2. Пункт А. не нужно выполнять, т.к. его задание выполнено в предыдущем п. 5. В результате должна быть получена итогтаблица линпрогминимума типа рис. 2.15, А.

7. Раскомментируйте в линпрогуказании `LinProgUnivUkz_Tab` строку с командой `Lmi_Lma=-1` и закомментируйте вышележащую. Запустите линпрогуказание кнопкой «Run...»,  Матлаб выдаст в Комокно результат поиска максимума с сообщением об ошибке - файл шаблона итогтаблицы линпрогмаксимума `ItoTabLPrma.rpt` не обнаружен.

8. Создайте шаблон итогтаблицы линпрогмаксимума согласно разд. 2.4.4.2, п. 3) без п. А). Получите итогтаблицу линпрогмаксимума типа рис. 2.15, А.

9. Создайте пустой Улорд-файл с именем «Rab4Familija(vasha)N», в который следует поместить ряд результатов работы. На 1-ю страницу введите заголовок «1. Дата (вставьте дату начала исполнения задания). Примеры вычисления линпрогооптимумов посредством универсальных линпрогуказания `LinProgUnivUkz_Tab` и итогтабличного линпрогооптимизатора `LinProgUnivesAvtomatMU_Tab`. Вар. N (номер задания)». «1.1. Линпрогмаксимум при наличии всех ограничений».

10. Закомментируйте весь раздел «Аргументы линпрогогфункции...» в указании `LinProgUnivUkz_Tab` и вставьте на новые строки в этот раздел данные вашего варианта из табл. дз5.1 разд. 5 с именами и в формате

линпрогуказания. Лишние данные в выбранном варианте с номерами, большими заданного n , не используются.

11. Запустите линпрогуказание: Матлаб выдасть итогтаблицу максимума типа рис. 2.15, А.

Если в итогтаблице флаг оптимизации $efma \neq 1$, то выполните корректировку условий оптимизации согласно разд. 2.5 с получением итогтаблицы с $efma=1$.

Войдите в папку Rab4 через диск D и откройте файлы обеих итогтаблиц с помощью Уорда. Выделите каждую таблицу, скопируйте их в буфер и вставьте в Уорд-файл, созданный в п. 9. Под 1-й таблицей выполните подпись: «Рис. 1. Результат вычисления линпрогмаксимума после запуска указания LinProgUnivUkz_Tab». Под 2-й: «Рис. 2. Результат вычисления линпрогмаксимума после коррекции bo и $beqo$ (некорректируемую переменную удалите)». Пересохраните Уорд-файл.

В сессфайле сделайте следующие 2 закомментированные записи: «Результат вычисления линпрогмаксимума после при наличии ограничений неравенств и равенств запуска указания LinProgUnivUkz_Tab» и «Результат вычисления линпрогмаксимума при наличии ограничений неравенств и равенств после коррекции bo и $beqo$ (некорректируемую переменную удалите). Под каждую запись в сессфайле вставьте имя своей итогтаблицы. Пересохраните сессфайл и Уорд-файл.

12. Вставьте в сессфайл и в Уорд-файл закомментированный заголовок рубрики: «1.2. Линпрогминимум при наличии всех ограничений». В линпрогуказании LinProgUnivUkz_Tab закомментируйте строку «Lmi_Lma=-1;» и раскомментируйте – «Lmi_Lma=1;». Выполните п. 11 для установленной конфигурации линпрогуказания с учётом необходимости изменения имён вставляемых объектов с максимума на минимум.

В результате в действующей папке должны появиться 2 файла итогтаблиц минимума (если была коррекция линпрогусловий, или 1 файл – без коррекции). В сессфайле в рубрике 1.2 – 2 (или 1 без коррекции) ссылки на файлы итогтаблиц минимума по типу ссылок из п. 11. В Уорд-файле – копии 2 таблиц минимума в рис.3 и 4 с подписями по типу подписей из п.11.

13. Вставьте в сессфайл и в Уорд-файл закомментированный заголовок рубрики: «1.3. Линпрогминимум при наличии только одного ограничения Ao » (для нечётных вариантов, для чётных - $Aeqo$). В линпрогуказании LinProgUnivUkz_Tab закомментируйте строки с $Aeqo$, $beqo$ (для чётных - Ao, bo) и раскомментируйте строку « $Aeqo=[]$; $beqo=[]$;» (для чётных - « $Ao=[]$; $bo=[]$;»). Выполните п. 11 для установленной конфигурации линпрогуказания с учётом необходимости изменения имён вставляемых объектов.

В результате в действующей папке должны появиться 2 файла итогтаблиц минимума (если была коррекция линпрогусловий, или 1 файл – без коррекции). В сессфайле в рубрике 1.3 – 2 (или 1 без коррекции) ссылки на файлы итогтаблиц минимума по типу ссылок из п. 11. В Уорд-файле – копии 2 таблиц минимума в рис.5 и 6 с подписями: «Результат вычисления линпрогминимума при наличии ограничений неравенств Ao , bo (для чётного варианта - $Aeqo$, $beqo$) после запуска указания LinProgUnivUkz_Tab» и «Результат вычисления линпрогминимума после коррекции bo (для чётного - $beqo$). Пересохраните сессфайл и Уорд-файл.

14. Вставьте в сессфайл и в Уорд-файл закомментированный заголовок рубрики: «1.4. Линпрогмаксимум при наличии только одного ограничения $Aeqo$ » (для нечётных вариантов, для чётных - Ao). В линпрогуказании LinProgUnivUkz_Tab закомментируйте строки с Ao , bo (для чётных - $Aeqo, beqo$) и раскомментируйте строку « $Ao=[]$; $bo=[]$;» (для чётных - « $Aeqo=[]$; $beqo=[]$;»). В линпрогуказании также закомментируйте строку «Lmi_Lma=1;» и раскомментируйте – «Lmi_Lma=-1;». Выполните п. 11 для установленной конфигурации линпрогуказания с учётом необходимости изменения имён вставляемых объектов.

В результате в действующей папке должны появиться 2 новых файла итогтаблиц максимума (если была коррекция линпрогусловий, или 1 файл – без коррекции). В сессфайле в рубрике 1.4 – 2 (или 1 без коррекции) ссылки на файлы итогтаблиц максимума по типу ссылок из п. 11. В Уорд-файле – копии 2 таблиц максимума в рис.7 и 8 с подписями: «Результат вычисления линпрогмаксимума при наличии ограничений неравенств $Aeqo$, $beqo$ (для чётного варианта - Ao , bo) после запуска указания LinProgUnivUkz_Tab» и «Результат вычисления линпрогминимума после коррекции $beqo$ (для чётного - bo). Пересохраните сессфайл и Уорд-файл.

15. Вставьте в сессфайл и в Уорд-файл закомментированный заголовок рубрики: «1.5. Линпрогмаксимум (для чётного варианта, для нечётного – линпрогминимум) при отсутствии ограничений неравенств и равенств». В линпрогуказании LinProgUnivUkz_Tab закомментируйте строки с Ao , bo , $Aeqo$, $beqo$ и раскомментируйте строки « $Ao=[]$; $bo=[]$; $Aeqo=[]$; $beqo=[]$;». Для нечётных вариантов в линпрогуказании также закомментируйте строку «Lmi_Lma=-1;» и раскомментируйте – «Lmi_Lma=1;». Для чётных вариантов этого делать не нужно.

Выполните п. 11 для установленной конфигурации линпрогуказания с учётом необходимости изменения имён вставляемых объектов. Коррекция условий при этих оптимизациях не нужна.

В результате в действующей папке должен появиться 1 новый файл итогтаблицы максимума для чётного варианта (минимума - для нечётного). В сессфайле в рубрике 1.5 – 1 ссылка на файл итогтаблицы максимума (минимума – для нечётного варианта) по типу ссылок из п. 11. В Уорд-файле – копия 1 таблицы максимума(минимума – для нечётного варианта) в рис.9 с подписью: «Результат вычисления линпрогмаксимума (линпрогминимума – для нечётного варианта) при отсутствии ограничений неравенств A_o, b_o и ограничений равенств A_{eqo}, b_{eqo} ». Пересохраните сессфайл, Уорд-файл и файл Вычпространства.

16. По результатам вычислений создайте в Уорд-файле табл.1, аналогичную табл.4.1 (дана для нечётного варианта, для чётного внесите соответствующие поправки), но со своими названиями величин и их значений.

Таблица 4.1. Зависимость липрогоптимума, вычисленного связкой инструментов LinProgUnivUkz_Tab и LinProgUnivesAvtomatMU_Tab, от вида и типа ограничений, вариант N.

Номер рубрики	1.1	1.2	1.3	1.4	1.5
Файл данных	TabLPrma_19_Aug_2016 15h24m34 N1_64207	TabLPrmi_19_Aug_2016 15h23m32 N1_64134	TabLPrmi_19_Aug_2016 15h36m52 N1_65061	TabLPrma_19_Aug_2016 16h07m24 N1_67181	TabLPrmi_19_Aug_2016 16h14m54 N1_67702
Флаг оптимизации, имя/значение	efma_1_64207/1	efmi_1_64134/1	efmi_1_65061/1	efma_1_67181/1	efmi_1_67702/1
Ц-функция, имя/значение	yua_1_64207_19_Aug_2016/ 35.0662	yui_1_64134_19_Aug_2016/ -2.7571	yui_1_65061_19_Aug_2016/ -66.2382	yua_1_67181_19_Aug_2016/ 63.8865	yui_1_67702_19_Aug_2016/ -202.6000
A_o, b_o	Есть	Есть	Есть	Нет	Нет
A_{eqo}, b_{eqo}	Есть	Есть	Нет	Есть	Нет

4.2. Применение универсальных линпрогуказания LinProgUnivUkz и линпрогоптимизатора LinProgUnivesAvtomatMU для решения примеров линпрогоптимизации с размерностью 1000...8000

17. Введите в сессфайл заголовок рубрики «2. Вычисление линпрогоптимума посредством универсальных линпрогуказания LinProgUnivUkz и линпрогоптимизатора LinProgUnivesAvtomatMU для размерности $n=$ (укажите размерность согласно варианту N из табл. дз5.2 разд. 4). Вар. N (номер задания)». «2.1. Вычисление линпрогминимума (для нечётных вариантов, линпрогмаксимума – для чётных) при наличии ограничений неравенств A_o, b_o и равенств A_{eqo}, b_{eqo} ».

18. Скопируйте в вашу папку из электронной версии монографии (см. конец разд. 2.2.3) программы: линпрогуказания LinProgUnivUkz (рис. 2.9) и линпрогоптимизатора LinProgUnivesAvtomatMU (рис. 2.5). Каждую в отдельный текстфайл с упомянутым именем. Программу LinProgUnivesAvtomatMU нужно копировать следующим образом: выделите объект-надпись (но не её текст) на рис. 2.5, скопируйте объект в буфер, откройте пустой рабочий Уорд-файл и вставьте туда объект, выделите весь текст объекта и удалите нумерацию строк нажатием кнопки «Нумерованный список...» , отправьте текст в буфер (выделение сохраняется), перейдите в Редактор Матлаба, откройте пустой М-файл и вставьте туда программу из буфера, сохранив её с именем LinProgUnivesAvtomatMU.

19. Вставьте в линпрогуказание LinProgUnivUkz значения $n, n_1 \dots n_4$ для варианта задания из табл. дз5.2 разд. 4. Для вариантов с $n \geq 5000$ необходимо увеличить число отображаемых строк в Комокне по сравнению с номиналом, равным 5000. Командами меню Матлаба «File_Preferences_Command Window» вызывается вкладка установок Комокна и в поле «Number of lines...» устанавливается величина, на 1000 большая, чем содержится тысяч в целой части 1-го множителя величины n .

Для нечётных вариантов в линпрогказании следует сменить тип оптимизации: нужно закомментировать строку скомандой «Lmi_Lma=-1;» и раскомментировать с командой «Lmi_Lma=1;» для вычисления линпрогминимума.

20. Запустите линпрогуказание LinProgUnivUkz. Из Комокна скопируйте в рубрику 2.1 сессфайла результат оптимизации в формате рис. 2.8, А: для вектора x нужно оставить первые 10 строк и последние 10, разделив их многоточечной строкой с указанием количества пропущенных строк. Строки с результатом нужно закомментировать.

Если Матлаб не нашёл решение ($efmi/efma = -2$), то также скопируйте в сессфайл результат квазиоптимизации в формате рис. 2.8, А с заголовком «1-й запуск». Далее очистите Комокно командой `clc` и повторите запуск `LinProgUnivUkz`: будут сгенерированы новые случайные значения условий оптимизации, и, возможно, будет найдено решение, которое нужно скопировать в сессфайл в указанном формате с заголовком «2-й запуск. Решение». Если вновь решение не будет найдено, то уберите из заголовка «Решение», очистите Комокно и вновь запустите указание. Скопируйте также его результат в сессфайл с заголовком по характеру решения: «3-й запуск, Решение»/ «3-й запуск, Решение не найдено».

21. Введите в сессфайл заголовок рубрики «2.2. Вычисление линпрогмаксимума (для нечётных вариантов, линпрогминимума – для чётных) при наличии только ограничений неравенств Ao , bo (для нечётных вариантов, для чётных – только ограничений равенств $Aeqo$, $beqo$)».

22. В указании `LinProgUnivUkz` смените тип оптимизации. В указании `LinProgUnivUkz` для нечётных вариантов: закомментируйте 2 строки с $Aeqo$ и $beqo$, раскомментируйте строку с « $Aeqo=[]$; $beqo=[]$;». Для чётных вариантов: закомментируйте 2 строки с Ao и bo , раскомментируйте строку с « $Ao=[]$; $bo=[]$;».

23. Запустите линпрогуказание `LinProgUnivUkz`. Из Комокна скопируйте в рубрику 2.2 сессфайла результат оптимизации согласно п.20. Если решение не будет найдено, повторите действия из п. 20 по такому случаю.

4.3. Вычисление зависимости линпрогоптимумов от одного параметра p с использованием универсального p -линпрогоптимизатора `pLinProgTabGenOptimizer` и построителя итогтаблиц `pLinProgUnverP1TabStroyker`

24. Введите в сессфайл и в Уорд-файл заголовок рубрики «3. Вычисление зависимости линпрогоптимумов от одного параметра p посредством универсальных p -линпрогоптимизатора `pLinProgTabGenOptimizer` и построителя итогтаблиц `pLinProgUnverP1TabStroyker`. Вар. N (номер задания)». «3.1. Вычисление p -линпрогминимума (для нечётных вариантов, p -линпрогмаксимума – для чётных) при наличии ограничений неравенств Ao , bo и равенств $Aeqo$, $beqo$ ». «1-й запуск».

25. Скопируйте в вашу папку из электронной версии монографии (см. конец разд. 2.2.3) программы: p -линпрогоптимизатора `pLinProgTabGenOptimizer` (рис. 3.12), формирователя отладтаблицы `pLinProgUnverOtladTab` и построителя итогтаблиц `pLinProgUnverP1TabStroyker` (рис. 3.15). Каждую в отдельный текстфайл с упомянутым именем. Для чётных вариантов переключите оптимизатор с минимума на максимум: закомментируйте команду « $pLmi_pLma=1$;» и раскомментируйте - « $pLmi_pLma=-1$;». Запустите линпрогоптимизатор `pLinProgTabGenOptimizer` с имеющимися в нем условиями : Матлаб выдаст в Комокно результат поиска минимума с сообщением об ошибке - файл шаблона итогтаблицы `pLinProgItogUnverTab.rpt` (для обоих типов p -линпрогоптимизации) не обнаружен.

26. Создайте шаблон p -итогтаблицы универсальной согласно разд. 3.5.3 с заменой имени файла шаблона и значений полей вставки на указанные на рис. 3.16. Имя шаблона: `pLinProgItogUnverTab.rpt`. В результате должна быть получена универсальная p -итогтаблица, обеспечивающая для нечетных вариантов формирование таблицы типа рис. 3.13, А, а для чётных – 3.13, Б.

27. Закомментируйте в p -линпрогоптимизаторе `pLinProgTabGenOptimizer` строку с `pLinProgUnverP1TabStroyker` и раскомментируйте – с `pLinProgUnverOtladTab`. Запустите линпрогоптимизатор: Матлаб выдаст в Комокно результат поиска (p -минимума/ p -максимума) с сообщением об ошибке - файл шаблона итогтаблицы `pLinProgUnverOtladTab.rpt` (для обоих типов p -линпрогоптимизации) не обнаружен.

28. Создайте шаблон p -отладтаблицы универсальной согласно разд. 3.5.3 с заменой имени файла шаблона и значений полей вставки на указанные на рис. 3.19. Имя шаблона: `pLinProgUnverOtladTab.rpt`. В результате должна быть получена универсальная p -отладтаблица, обеспечивающая для нечетных вариантов формирование таблицы типа рис. 3.18, а для чётных – аналогичную по структуре таблицу, в именах величин которой буква « i » заменена на « a ».

29. Закомментируйте в p -линпрогоптимизаторе `pLinProgTabGenOptimizer` строки со следующими величинами: p_{uo} , p_{fo} , p_{Ao} , p_{Aeqo} , p_{LBo} . Раскомментируйте строку с `pLinProgUnverP1TabStroyker`. Вставьте данные варианта из табл. дз5.3 в оптимизатор на дополнительно образованные строки с точно теми же именами, что закомментировывали ранее. Для вставки значений параметра p_{uo} следует рассчитать шаг параметра shp по формуле

$$shp=(p1k-p1n)/(mp1-1). \quad (4.1)$$

Величины в формуле соответственно: конечное значение параметра, начальное и количество значений – всё из табл. дз5.3.

Запустите оптимизатор: Матлаб выдаст отладтаблицу и итогтаблицу в формате html (минимума - для нечётных вариантов, максимума – для чётных), а также создаст в действующей папке копию оптимизатора с именем файла итогтаблицы. Имена обеих таблиц с расширением html и имя копии оптимизатора с расширением m скопируйте в сессфайл. Сами таблицы скопируйте по способу п.11 в Уорд-файл в рис. 10 и 11. Подписи в рисунках: «Отладтаблица (Итогтаблица) р-линпрогминимизации для нечётных вариантов, для чётных – р-линпрогмаксимизации».

30. Если в части итогтаблицы (минимума/ максимума) или во всей нарушается равенство

$$\text{refmi}/\text{refma}=1, \quad (4.2)$$

то необходимо выполнить коррекцию условий р-линпрогоптимизации согласно правил и примера разд 3.4 и примера разд. 3.6.4. При проведении коррекции строку с `pLinProgUnverP1TabStroyker` в р-линпрогоптимизаторе `pLinProgTabGenOptimizer` следует закомментировать. После того, как будет получена отладтаблица с выполнением равенства (4.2), строку с `pLinProgTabGenOptimizer` раскомментировать, запустить оптимизатор `pLinProgTabGenOptimizer`, который выдаст итогтаблицу с правильным решением задачи р-линпрогоптимизации.

Имена файлов отладтаблиц коррекции следует помещать в сессфайл в рубрики с заголовками: «1-й шаг коррекции», «2-й шаг коррекции» и т.д.. В рубрику последнего удачного шага коррекции помещают также и имя итогтаблицы с решением. Сами отладтаблицы и итогтаблицы с решением копируют по способу п.11 в Уорд-файл, помещая их в рис. 12, 13 и т.д. с подписями: «1-й шаг коррекции», «2-й шаг коррекции» и т.д. В этих таблицах, начиная с исходной, выделяют зелёным цветом значения компонент вектора rbo , используемых для формирования новых значений этого вектора для следующего шага коррекции. Сформированный вектор помещают под таблицей, по которой он формировался, с определением: «Вектор b для шага i : $\text{rbo}=[\dots]$ ». Аналогично оформляются и другие объекты, используемые для очередного шага коррекции (например, вектор границы ц-области). В рисунке i -го шага коррекции, когда получено решение р-линпрогоптимизации, должны быть отладтаблица и итогтаблица с ярлыками «А» и «Б», выполненными посредством инструмента «Надпись» (кнопка вызова -  на панели «Рисование»), поставленными и закрепленными в строках заголовков таблиц, аналогично рис.3.8, Б.

31. По результатам коррекции составьте в Уорд-файле табл.2 разности ц-функций из итогтаблиц решения и исходной, аналогичную в оформлении и в формате величин табл. 3.22 по методике п. Е из разд. 3.6.4.

32. Введите в сессфайл заголовок рубрики: «3.2. Построение зависимости $\text{F}_{\text{cmin}}(\text{puo})$ » для нечётных вариантов (для чётных - $\text{F}_{\text{cmax}}(\text{puo})$). Постройте эту зависимость по методике разд. 3.3. Сохраните фигур-файл с именем « $\text{F}_{\text{cmin_puo_vN_Familija}$ » для нечётных вариантов, для чётных - « $\text{F}_{\text{cmax_puo_vN_Familija}$ ». Оформите фигур-файл зависимости согласно рис. 3.4 разд. 3.3. Вставьте снимок фигуры в Уорд-файл в рисунок с соответствующим номером и с подписью, аналогичной рис. 3.4. Пересохраните Уорд-файл, сессфайл, фигур-файл и файл Вычпространства.

33. По результатам выполнения Задания 4 должны быть созданы следующие выходные объекты задания в действующей папке.

1) Сессфайл « $\text{R4SessFamilija(vasha)N(N - вар.зад.4)}$ », у которого 3 рубрики и 8 нумерованных подрубрик, содержащих: 6 имён файлов итогтаблиц (если не было коррекции) или 11 имён (если была одношаговая коррекция) в рубриках 1 и 3; имя 1 файла отладтаблицы (или столько имён, сколько было шагов коррекции) в подрубрике 3.1; результаты линпрогоптимизации для крупной размерности в подрубриках 2.1 и 2.2, сокращённые в части представления компонент вектора x (10 первых и 10 последних); команда построения графика $\text{F}_{\text{cmin}}(\text{puo})/\text{F}_{\text{cmax}}(\text{puo})$ в подрубрике 3.2.

2) Уорд-файл « $\text{Rab4Familija(vasha)N.doc}$ », у которого 2 рубрики (1 и 3, 2-я пропущена) и 6 нумерованных подрубрик, содержащих следующие данные. 6 копий по п.11 файлов итогтаблиц (если не было коррекции) или 11 копий (если была одношаговая коррекция) в рубриках 1 и 3. Копия по п.11 1 файла отладтаблицы (или столько копий, сколько было шагов коррекции) в подрубрике 3.1. Итогтаблицы и отладтаблицы помещены в рисунки с номерами 1...12 (или более – в зависимости от количества шагов коррекции условий р-линпрогоптимизации); под каждой отладтаблицей должен быть помещён вариант вектора rbo , сформирован-

ный по этой таблице. После подрубрики 1.5 должна находиться табл.1 «Зависимость липрогоптимума, вычисленного связкой инструментов LinProgUnivUkz_Tab и LinProgUnivesAvtomatMU_Tab, от вида и типа ограничений, вариант N». В конце подрубрики 3.1 должна находится табл.2 разности ц-функций из итогтаблиц решения и исходной. В подрубрике 3.2 должен находится снимок фигур-окна с графиком зависимости $F_{\min}(\text{пуо})/F_{\max}(\text{пуо})$.

3) Файл Вычпространства «WS_R4Familija(vasha)N.mat».

4) Все файлы, обеспечивающие формирование результатов в пп. 1...3, должны быть также в папке.

5. Варианты данных к Заданию 4

Табл. дз5.1, дз5.2 и дз5.3

5.1. Структура линпрогпримеров (ЛПП) для данных табл. дз5.1

Ц-функция (дз5.1), ограничения-неравенства (дз5.2), ограничения-равенства (дз5.3), ц-область (дз5.1).

Конкретное значение размерности n ЛПП определяется вторым столбцом табл. дз5.1: лишние данные в выбранном варианте с номерами, большими n ЛПП, не используются.

Целевая функция:

$$y_{41}(x) = f_1x_1 + f_2x_2 + f_3x_3 + f_4x_4 + f_5x_5 + f_6x_6 + \dots + f_{10}x_{10}. \quad (\text{дз5.1})$$

Ограничения-неравенства:

$$\left. \begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 + a_{15}x_5 + a_{16}x_6 + \dots + a_{110}x_{10} &\leq b_1, \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 + a_{25}x_5 + a_{26}x_6 + \dots + a_{210}x_{10} &\leq b_2, \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 + a_{35}x_5 + a_{36}x_6 + \dots + a_{310}x_{10} &\leq b_3, \\ a_{41}x_1 + a_{42}x_2 + a_{43}x_3 + a_{44}x_4 + a_{45}x_5 + a_{46}x_6 + \dots + a_{410}x_{10} &\leq b_4, \\ a_{51}x_1 + a_{52}x_2 + a_{53}x_3 + a_{54}x_4 + a_{55}x_5 + a_{56}x_6 + \dots + a_{510}x_{10} &\leq b_5. \end{aligned} \right\} \quad (\text{дз5.2})$$

Ограничения-равенства:

$$\left. \begin{aligned} a_{e11}x_1 + a_{e12}x_2 + a_{e13}x_3 + a_{e14}x_4 + a_{e15}x_5 + a_{e16}x_6 + \dots + a_{e110}x_{10} &= b_{e1}, \\ a_{e21}x_1 + a_{e22}x_2 + a_{e23}x_3 + a_{e24}x_4 + a_{e25}x_5 + a_{e26}x_6 + \dots + a_{e210}x_{10} &= b_{e2}, \\ a_{e31}x_1 + a_{e32}x_2 + a_{e33}x_3 + a_{e34}x_4 + a_{e35}x_5 + a_{e36}x_6 + \dots + a_{e310}x_{10} &= b_{e3}. \end{aligned} \right\} \quad (\text{дз5.3})$$

Ц-область: нижние 1) и верхние 2) границы:

$$\left. \begin{aligned} 1) \text{ все компоненты ц-вектора } x \text{ больше нуля,} \\ 2) x_1 \leq g_1, x_2 \leq g_2, x_3 \leq g_3, x_4 \leq g_4, x_5 \leq g_5, x_6 \leq g_6 \dots x_{10} \leq g_{10}. \end{aligned} \right\} \quad (\text{дз5.4})$$

Коэффициенты f_i , a_{ij} , b_i , a_{eij} , b_{ei} и g_i функции $y_{41}(x)$ и ограничений (34.2), (34.3) и (34.4 2)) – в табл. 34.1.

ТАБЛИЦА дз5.1
Коэффициенты линпрогпримеров

Вар.	Размерность, n ЛПП	Коэффициенты ц-функции										Коэффициенты 1-й строки неравенств (дз5.2)									
		f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	a_{16}	a_{17}	a_{18}	a_{19}	a_{110}
1	7	2.2	-6	-0.8	7.2	-0.5	6.7	3.4	1.6	9.3	15	3.1	-0.4	8.5	3.6	8.7	3.5	1.1	0.6	8.6	4.4
2	5	7.3	3.6	3.8	8.9	5.3	7.6	1.4	-0.7	7.8	2.9	2.7	1.8	8	12.5	4.8	7.1	2.2	1.5	7.3	2.4
3	7	1.9	3.8	4.6	2.1	10	7.1	1.6	-0.4	10.2	10.7	2.6	-1.3	8.5	13.6	12.5	2.2	2	-2.1	10.3	9.9
4	8	2.2	-4.1	-2.3	-0.8	10.4	7.3	2.7	-1	7.9	1.3	1.7	2.1	8.3	2.1	10.2	5.3	0.2	-1.2	8.4	3.7
5	10	4.6	2.9	0.2	6.3	9.8	3.4	1.3	-2.8	9	7.3	2	-2	10.8	10.2	12.7	4	2.4	0.2	8.5	7.2
6	7	1.7	3.2	1.5	2.8	-1.6	6.9	2.3	1.8	8.8	1.9	1.4	1.3	8.2	4.8	13.7	6.4	0.4	-1	9.1	5.3
7	6	3.4	0.2	2.9	-0.4	3.9	7.6	2.6	2.2	5.5	8.4	2.6	-2	7.2	2.1	-1.4	7.9	2.2	-0.7	6.3	11.6

Вар.	Размерность, n, ЛПН	Коэффициенты ц-функции										Коэффициенты 1-й строки неравенств (дз5.2)									
		f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	a11	a12	a13	a14	a15	a16	a17	a18	a19	a110
8	5	7.4	-1.7	3.4	0	-2.8	5.6	3	0.5	6.7	3.5	3	-1.4	5.6	6.9	10.8	4.9	0.7	-1.7	9.9	10.7
9	6	8.3	-7.1	4.5	9.3	-1.9	7.1	2.2	-1.6	10.1	1.4	3.2	-2	8.2	11.2	13.7	4.9	0.8	1.5	6.1	6.3
10	8	5.4	-7.7	2.5	-0.5	-0.8	7.8	3.7	1	9.1	1.9	3.9	2	9.6	13	6.8	3.5	1.4	-0.1	7.2	3.5
11	7	1.8	-0.5	-0.8	1.3	-2.1	5.9	1.9	-1.7	10.5	14.7	2	-1.3	5.5	12	4.8	6.1	0.9	-1	10.1	8.5
12	9	3.4	0.5	-0.2	6.3	4.1	5.8	1.5	0.8	8.7	10.8	1.2	1.5	7.2	3.1	10.2	4.5	3	2.3	8.4	12
13	9	2	-2.5	-2.3	2.5	10.5	7.8	1.3	1.2	8.7	8.6	2.1	-1.3	10	12.5	5.5	3.4	1.7	1.7	10	4
14	5	6.2	-4.4	2.6	-1.5	7.7	6.2	2.1	1.4	9.5	3	1.2	-1.3	5.3	1.2	2	6.9	1.3	1.7	7	12.5
15	6	3.8	-2	-2.2	5.6	4.1	6.6	1	-1.3	9.5	2.8	3.3	1.2	9.6	14.3	-1.3	5.2	1.2	2.1	11.2	5.8
16	7	3.6	2.4	-0.8	7.3	-0.7	7.1	3.9	1.6	10.1	4.4	3.7	-0.1	10.4	5.7	12.6	2.6	0.5	2.2	10.3	5.2
17	10	8.2	-3.5	3.2	3.3	-2.2	6.5	2.6	-1.3	6	4.3	1.6	1.9	10.8	10.3	3.1	2.1	0.6	-1.9	7.8	10
18	7	1.4	-0.5	1.4	2	-3.7	4.4	1.4	2	7.5	10.2	1.6	-1.1	7.8	4.7	6	7.5	2.2	-0.5	10	5.6
19	9	6.1	-4.1	4.2	4.9	-2	5.7	3.4	0.8	8.8	9.9	1.1	0.8	10.6	8.2	2.5	3.6	2.4	2.3	8.3	3.9
20	6	8.8	-1.1	4.4	-1	-1.5	4.5	2	1.7	7.3	13.1	1.2	0.4	9.5	7.7	8.5	7.4	2.5	-0.6	6.2	11.9
21	8	7.4	-4.7	2.4	8.5	-1.5	4.9	2.7	-1.6	9.5	14.4	2	-1.2	6.9	2.7	6	2.1	0.4	2.3	8.5	10.4
22	7	6.3	-4.6	4.1	0.8	10.3	6	2.3	2.1	7.3	8.2	2.8	1.9	6	8.9	12.2	2.8	0.7	-1.2	9.3	9.4
23	5	3.3	-2.3	3	-1.9	0.7	5.8	2.8	1.6	9.9	3.2	2.9	-1.2	8.8	8.8	11.4	5.3	0.3	-0.9	6.4	3.4
24	8	5.9	-3.3	-2.5	3.4	-1.7	3	1.6	1.8	10.8	14.1	2.7	-1.2	5.9	10.4	4.3	3.8	1.7	2.3	6.9	6.4
25	7	4.1	3.8	3.3	-0.7	1	7.6	2.7	-2.8	6.6	13.6	1.8	0.1	10.8	9.8	7.5	6.5	1.4	-0.3	11	9.2
26	5	4.2	-3.9	-1.7	3	8.2	3.4	2.7	-0.2	8.8	13.9	1.5	-1.8	9.4	4.9	10.5	4.9	1.2	1.7	11.3	9
27	10	6.8	-6	-2.3	-1.1	6.3	6	1.2	-2.8	8.9	7.2	3.9	0.7	6.1	2.2	7.5	5.6	2	2.3	11	7.2
28	5	4.8	4	2.5	-1	8.1	3.8	1.3	1.3	9.6	12	2.4	1.2	9.4	6.3	11	8.9	2.1	1.1	10.8	5.7
29	9	1.2	-0.7	-3.4	1.3	10.5	3.1	2	0.3	8.7	14.5	3.2	1.8	9.3	4.9	13.1	7.9	2.4	0.6	6.9	12
30	5	5.3	-2.4	3.1	-1.7	-3.6	3.9	2.7	-1.3	7.3	12.3	2.5	-0.3	6.2	11.9	6.6	2.9	2.4	1	9.2	2.5
31	6	7	0.3	2.5	3.7	7.6	3.4	4	2.1	6.2	14.9	2.4	0.7	7.7	6.1	1.1	7.7	1.2	0.8	7.9	6.2
32	5	7.2	2.9	0.1	3.2	2.5	4.3	3	-0.7	5.2	13.7	3.2	-2.6	7.1	13.2	10	5.3	1.2	1.8	11.3	12.1
33	7	3.1	-7.8	5.5	7.6	4.3	7.2	3.8	-2.4	8.6	12	1.9	-0.1	9	7.5	3.3	8.3	2.9	-0.1	9.8	4
34	8	6	-1.5	4	2.2	7.9	3.1	3	1.9	7.7	13.1	2.2	1.3	9.7	10.1	1.4	8.1	2.9	-2	6.5	7.6
35	5	2.5	-5.5	-2.7	2.7	2.2	4	2.8	1.6	8.8	6.6	2.3	-2.3	9.5	14.3	-1.9	3.7	2.2	-1.4	7.6	5.9
36	8	5.1	1.8	3.9	5.6	1.7	6.3	3.4	-2.1	7.3	12.2	2.3	-0.7	8.1	4	14	5.9	1.4	1.9	7.8	6.5
37	7	8.8	3.8	0	6.6	3.1	5.3	1.8	0.2	8.3	5.6	2	2.3	7.8	4.5	7	7.9	2	-0.1	11.4	13
38	5	2.2	1.3	5.1	-0.8	1.1	7.5	1.5	-1.5	7.2	4	3.9	1.8	5.6	8.8	15	4.3	2.5	2.4	9.5	9.2
39	10	3.3	-5.4	-1.2	-0.8	7.6	5.9	1.4	-0.7	10.6	5	2.7	-1.7	8	5.6	11.5	7	1.5	-1.1	8.9	4.3
40	5	8.1	-1	0.3	5.5	-2.1	6.8	2.5	-0.8	9	7.3	3.5	-0.6	8.8	14.6	2.8	3	2.3	0.2	8.6	3.9
41	6	1.2	3	4.9	4	0	7.1	1.1	2.1	7.9	3.7	3.9	0.2	6.6	4.9	0.2	5.6	1.3	1.4	7.4	2.6
42	7	6.7	-2.1	-0.3	2	6.9	4.2	3.9	-2.2	6.1	10.4	2.3	0.1	5.5	11.1	2.4	8.2	2.8	1.5	10.4	12.6
43	8	7.3	-0.7	-3.7	-1	10.3	5.8	2.1	-1.6	8.1	13.1	3.7	0.8	7.3	2.9	-1.1	6.4	1.1	2	7.9	9.7
44	6	3.6	1.9	5.4	8.6	9.5	7.4	1.7	-0.3	8.6	6.8	3.8	-0.2	8.1	7.6	6.1	6.1	2.9	-0.3	7.6	6
45	9	5.5	-4.6	5.5	4.8	5.1	4.6	2.4	1.3	10.4	7.3	1.8	-1.8	5.9	5.9	1.7	4.1	1.6	-2.1	7.9	3.9
46	6	8.3	-1.1	3.1	9.4	1.2	4.7	1.1	0.1	9.2	10.4	2.9	0.5	8.4	15	0.6	7.8	0.8	1.3	11.2	4.4
47	5	4.5	-3.9	3.6	9.1	-0.2	3.9	2.5	1.2	6.8	2.8	2.5	-1	5.7	1.9	14.6	8	0.1	0.9	11.5	10.8
48	6	2	-6.6	-3.7	6.9	10.6	6.9	3.9	-1.7	10	9	3.3	-0.4	10.8	11	8.8	7.2	1.1	-1.7	6.4	3.6

Вар.	Размерность, n, ЛПН	Коэффициенты ц-функции										Коэффициенты 1-й строки неравенств (дз5.2)									
		f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	a11	a12	a13	a14	a15	a16	a17	a18	a19	a110
49	7	8.9	-1.2	3.3	3.3	8.4	3.9	2.7	1.5	6.7	3.1	2.3	-2.1	6.7	9.5	9	4.9	1.1	2.5	11.4	11.5
50	5	7.2	-2.4	4.2	8.7	-3.4	5.7	3.8	-0.6	8.3	12	3.5	-0.2	7.4	9.5	14.5	3.5	1.8	2	8.9	9.4
51	6	3.8	1.2	-2.5	6.5	-3.1	6.6	3.5	-2.6	8	10.4	3.8	1.7	10.7	5.9	15	4.5	1.9	-0.2	6.3	7.9
52	10	7.2	2.4	3.8	2.6	4.7	6.1	3.5	-1.3	9.5	7.2	1.3	-2.9	8.1	14	4.6	6.7	2.9	-2.5	10.6	9.4
53	5	5.2	3.1	1.9	3.8	-1.4	4.4	1.7	1.3	5.7	7.7	1.7	1.8	9.1	6	13.8	7.9	0.8	-1.8	10.9	7.6
54	7	8.3	2.2	1	9.1	7.4	4.5	2.9	-0.5	9.3	13.8	3.1	-0.8	6.4	7.8	-1.7	4.3	1.7	0.4	8.3	9.6
55	9	7.8	-0.8	5.8	-0.8	1.2	7.5	1.2	-2.2	10.8	2.9	1.1	-1.3	9.1	14	-2	5.4	1.3	1.5	9	7.9
56	9	7.8	1.4	2.8	3.2	2.9	7.2	2.2	1.8	10.5	3.4	1.3	1.3	6.5	10.7	7.3	8.1	1.2	-0.4	11.6	8.3
57	5	4.1	-2.8	-0.7	1.3	4.4	5.8	2.8	-2.4	8.4	7.9	2.2	-1	8.6	8.3	10	8.3	1	2	10.5	4.1
58	7	2.4	3	-1.4	-1.8	10.7	5.6	1.9	2	9.7	1.7	1.8	-2	10.9	4.7	0.8	7.5	1.2	1.7	8.7	3.6
59	5	8.6	-2.8	4	-1.7	7.7	6.6	1.3	1.9	6.5	4.6	2.8	-2.1	7.5	3	0.1	7.5	1.2	1.7	7	3.4
60	8	5.3	3.6	-3.5	0	10.9	5.1	2.9	-1.6	5.7	2.9	1.6	2.2	9.9	11.7	6.6	3.4	1.1	-2	7.9	3.7
61	7	5.9	3.4	-1.8	5.2	-3.4	4.7	2.1	-1.9	10.2	7.7	2.3	2.5	7.7	10.4	13.3	5.2	1.9	0.5	10.1	3.4
62	6	1.4	-4.5	3.8	3.1	5.6	7.5	1.6	1.7	9.7	10.5	1.3	1.8	9.5	12.9	14.7	4	2.9	-2	6.6	6.2
63	9	4.9	3.6	-4.7	-1.9	3.5	7.1	2.7	0.3	5.9	9.2	2.3	-1	9.9	3.4	1.2	5.9	3	-2.1	10.5	3.4
64	6	7.4	-0.5	-2.4	2.8	-4	7.4	2.3	2	10.7	3	1.3	0.2	7.6	7.3	0.9	4.9	2.1	0.3	7	3.4
65	8	4.9	-0.1	1.2	-0.4	9.3	7.3	2.9	-1.6	6	14.9	1.7	-2	8.5	11.6	5.8	7.1	0.2	0.4	10.8	9.9
66	9	6	-1.4	4.2	1.6	7	6.2	1.1	-2.9	9.1	12.7	3.1	1.1	8.9	10.3	7.6	8.2	2.9	1.1	7.6	10.9
67	7	8.3	1.8	-1.8	-1.1	3.6	6.3	3.9	-1.8	5.1	12	1.9	-1.6	5.5	14.9	11.7	8.9	1.1	0.1	6.4	3.7
68	9	2.4	-2.6	5.4	8.7	-1.7	7.1	2.9	-1.8	6.1	5.2	3.5	-0.1	6.8	10.5	7.9	4.5	0.8	-0.2	11.9	2.8
69	6	6.1	-6.2	2.1	5.3	-0.4	7.2	1.6	1.5	6.1	7.8	1.1	-0.1	7.4	6.6	10.6	7	2.3	1.1	7.3	5.2
70	8	2.3	-3.1	-2.3	-1.9	4.3	6.9	3.5	-0.6	7.7	11.7	3	-2.8	10.5	7.7	13.4	2.3	1.1	-2.1	6.6	3.2
71	9	6.2	3.4	-2.9	2.4	5.4	3.8	2.4	-2.3	8.9	8.9	2.2	-2.6	8.1	1.7	-1.7	8	3	0.5	11.9	9.8
72	5	2.2	-1.3	0.3	4.3	-3.8	4.2	1.3	1.4	8.8	7.7	3.9	2.1	9	9.7	0.4	6.7	3	0.2	6.2	7.1
73	9	6.7	-5.3	5.7	7.4	3.1	5.1	3.3	2.2	6.9	8.3	1.7	0.3	6.8	12.8	6.7	4	1.5	0.3	7	5.9
74	6	3.9	-2.3	-3.4	-1.7	4	5	2.8	-0.3	9.9	3.4	2.5	-1.7	10.7	13.5	14	3.2	1.2	-0.8	7.2	3.5
75	8	1.6	1.1	-1.5	1.6	-1.4	7.7	1.6	0.3	9.4	4.8	1.6	-2.6	9	7.5	9.2	8.8	1.1	0.9	6.1	10.5
76	7	3	-6.9	1.1	9.4	6.2	4.7	3.2	-0.4	9.2	6.8	4	-1	8.7	1.3	5.3	4	0.6	-2	8.8	5.3
77	6	6.5	-5.4	-0.1	8.7	-2.6	6.5	2.9	1.5	8.4	10.8	3.6	-0.4	5.9	8.3	7.8	7.9	0.3	-1.3	11.3	8.3
78	7	2.2	0.2	5.5	3.8	6.4	3.6	2.7	1.4	8.6	3.6	1.2	-1.9	9.7	7	1.7	8.4	1.8	0.9	9.7	5.5
79	8	2.4	0.8	-4.3	2.8	-1.2	5.1	1.9	1.8	5.3	1.9	2.1	-1.4	6.3	14.4	2.5	2.7	2.4	-1.6	8.8	2.3
80	7	6.3	-2.3	-3.9	10	-2.7	7.9	3.8	2	8.1	7	3.4	0.9	7.2	1.8	10.3	5	2.6	-1.7	6	6.7
81	9	7.5	-7.3	-0.1	7.3	0.9	3.8	3.8	-1.3	10	2.8	2.5	0.8	8	4.3	-1.4	6.7	0.1	1.8	6.8	5.9
82	7	2.8	-4.1	5.4	1.9	7.8	5.4	3.9	-0	8.9	13	2.6	0.3	6.6	4.7	1.1	8.4	0.3	-0.5	7.8	10.9
83	5	7.4	1.7	5.5	6.5	1	4.3	2.6	1.2	6	11.1	1.1	2	7.4	5.5	9	6.9	1.1	-0.8	7.7	10.7

Вар.	Размерность, n, ЛПН	Коэффициенты ц-функции										Коэффициенты 1-й строки неравенств (дз5.2)									
		f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	a11	a12	a13	a14	a15	a16	a17	a18	a19	a110
84	6	5.1	-5.5	3.2	5.3	9.5	6.8	3.8	1.2	9.7	11.9	1.8	-0.4	6.7	13.9	-1.2	3.8	0.6	-0.7	7.9	6.5
85	8	2.4	2.2	1.6	5.9	0.8	5	2.5	1.3	5.7	9	3.8	-3	6.1	14.8	0.8	7.9	1.2	0.2	8.1	12.7
86	10	1.7	0.4	0.5	-1.1	10.7	7	2.8	0.1	8	5.4	1.4	-1.7	10	9	-0.7	4.4	1.7	-0.4	7.7	4
87	5	5.2	-4.6	2.6	8.9	1.5	5.9	2.7	-2.2	10	2.6	2.8	-1.4	5	8.7	-0.1	2.6	3	0.2	8.9	7.4
88	6	7.5	-3.2	3.7	3.4	-3	7.1	1.6	0.6	5.8	9.7	3.7	1.9	7.6	14.2	13.8	3.2	0.2	-1.4	11.9	3.2
89	7	8.3	-6.1	-4.6	4.3	10.1	6.6	3.2	2.1	5.9	3.7	2	1	10.2	6.8	13.5	7.9	1.5	-1.6	8.1	7
90	9	7.4	-3.1	5.7	1.5	-0.7	6.9	3.4	2.3	8.5	9.2	2.1	1.5	10.7	7.9	6.2	2	0.3	0.3	9	10.7

Продолжение 1 табл. дз5.1

Вар.	Размерность, n, ЛПН	Коэффициенты 2-й строки неравенств (дз5.2)										Коэффициенты 3-й строки неравенств (дз5.2)									
		a21	a22	a23	a24	a25	a26	a27	a28	a29	a210	a31	a32	a33	a34	a35	a36	a37	a38	a39	a310
1	7	1.9	1.7	9.3	-3.2	9.1	3.5	0	1.4	6.1	7.4	1.9	2.3	6.9	6.8	9.3	4.1	1	-1.2	10.7	6
2	5	0.5	-1.3	6.6	10	2.8	4.6	1.4	1.1	2.7	6.4	0.5	1.4	9.3	7.1	10.3	1	4.4	-0.2	10.8	12.2
3	7	2.3	1.3	9.1	-2.3	13.4	4.9	1	2.6	3	11	2.3	0.1	10.5	7.1	1.4	5.3	3.4	-0.5	13	1.4
4	8	2.1	0.2	9.7	3.8	8.3	8	2.1	1.7	4.6	3.6	2.1	2.3	5.5	6.4	13.7	3.2	2.8	-0.6	3.2	15.7
5	10	1	0.4	9.7	9.2	1.2	8.5	3.7	2.1	5.4	7	1	1	5.1	5.6	1.5	1.6	3.5	1.6	8.6	8.3
6	7	1.7	0.1	8.4	-2	11.4	7.6	-1.3	1.3	7	8.8	1.7	-2	7.1	6.8	8.7	2.4	5.3	2.9	13.7	14.5
7	6	0.2	1.4	8.1	1	4.4	4.9	3.3	3.5	2.8	3	0.2	-0.8	7	7.2	1.2	6.2	5.6	-0.9	8.2	6.4
8	5	2.2	0.6	9.7	7.4	13	8.2	4.2	3.3	4.6	7.5	2.2	0.7	11.5	7.3	7.9	5.4	4.1	-1.4	4.4	3.8
9	6	2.4	1.7	8.3	4.2	14.3	6	0.9	3.2	4.7	3.1	2.4	-2.5	7.2	6.9	11.2	5.4	4.8	0.9	12.2	5.4
10	8	1.7	1.1	9.6	3	13.3	4.3	-1.4	1.6	1	3.7	1.7	-0.6	10.3	4.4	13.3	1.8	1.5	-0.3	9.1	3.5
11	7	2.7	-0.8	6.5	8.2	9.6	6.3	-1.2	1.9	5.7	3.6	2.7	0.7	8.1	1.9	8.3	5.6	0.7	1.5	8	11.2
12	9	2.4	2	11.5	-2.3	14.6	3.7	4.9	2.6	6.9	3.2	2.4	0.1	11.8	5.9	12	1.4	1.7	0.2	8.6	5.3
13	9	2.2	1.5	6.8	0.7	14.4	3.1	1.4	1.3	2.8	9.4	2.2	2.1	5	3.6	4	6.2	5.7	2.9	12.3	-2.1
14	5	2.3	-1.3	11.5	1.6	6.2	7.8	5.1	2.2	0.6	9.4	2.3	-1.7	9.8	5.2	7.4	1.3	1.8	1	9.6	14.2
15	6	0.2	-0.2	10.7	3.5	10.1	6.5	1.2	2.1	5.9	5.7	0.2	1.1	4.5	4.8	12.2	4.8	3.9	2.4	3.7	15.6
16	7	0.9	0.4	7.9	4.4	14.1	3.6	5.6	2.5	5.8	4	0.9	-2.7	4.1	4.8	3.1	3.1	4.7	2.7	9.4	12.2
17	10	1.9	-1.2	7.1	-3	12.6	5.3	3.1	2.3	3.3	4.3	1.9	1.8	4.7	7.6	3.4	2.2	3.7	0.8	8.8	0.5
18	7	1.3	1.8	10.8	2	6.4	6.7	5.1	3.2	2.3	8.4	1.3	-1.8	9.9	6	2	4.3	1.8	-1.2	4.6	6
19	9	2.6	0.1	7.5	6.1	4.6	5.8	2.3	2.8	4.1	10.5	2.6	0.3	4.5	7.7	0.1	3.2	2.9	2.9	8.5	7.4
20	6	1.4	0.2	7.2	-2.4	7	8.2	2.4	1.4	1	8.8	1.4	2.5	5	2.9	11.6	2.3	2.6	1.8	11.3	10
21	8	1.5	0.9	11.2	1	4.4	6.6	0.8	2.6	1.8	10.3	1.5	2.4	7	9.7	9.3	5.7	3.3	-1.1	8.8	12.1
22	7	0.2	1.4	8.6	-4.3	14.1	5.3	-0.2	2	3.8	11.8	0.2	2.6	6.1	8.7	11.5	5.4	3.7	0	4	8.5
23	5	0.9	-1	9.6	6.2	5.5	4	2.8	1.3	0.7	5.7	0.9	-0.3	4.7	9.2	7.1	4.3	5.1	1.2	6	5
24	8	1.6	0	8.2	9.2	14.3	2.1	3.9	1.3	6.4	5.5	1.6	-0	6.5	1.9	12.2	5.4	2.9	-1.1	5.3	15.9
25	7	0.8	1.8	11.3	8.9	8.7	4.5	0.1	2.6	6.1	7.3	0.8	1.1	7.6	6.7	1.6	6	0.9	-1.1	8.4	-2.7
26	5	1.9	-0.4	6.4	7.7	5.2	6.4	0.7	2	1.4	10.9	1.9	2.1	8.4	9.7	8.5	3.5	5.4	0.6	4.8	13.7
27	10	2.1	0.5	9.8	2	15.7	6	1.2	1.1	0.7	3.4	2.1	-1.5	5	9.2	12.7	1.4	5.2	0.2	13.2	-2.9
28	5	0.1	1.5	9.8	7.8	2.9	9	5.9	2.8	0.9	7.3	0.1	-2.9	9	8.3	3.1	5.9	5.9	-1.5	11.6	14.4

Вар.	Размерность, n, ЛПН	Коэффициенты 2-й строки неравенств (дз5.2)										Коэффициенты 3-й строки неравенств (дз5.2)									
		a21	a22	a23	a24	a25	a26	a27	a28	a29	a210	a31	a32	a33	a34	a35	a36	a37	a38	a39	a310
29	9	2.3	1.5	9	8.4	11.4	5.6	3.8	1.5	4.4	5	2.3	0.1	11.2	3.9	12.8	1.8	1.3	1.6	5.6	13
30	5	0.4	1.7	11.2	7	5.5	4.1	-1.9	1	6.2	7.7	0.4	0.5	5.1	4.9	8.4	5.5	5	2.4	2.3	13.1
31	6	2.7	0.8	9.9	-0.4	13.8	6	2.4	3.4	1.6	3.2	2.7	-0.7	5.1	10.1	9.7	2.8	0.8	-0	8	9.2
32	5	2.2	-1	6.7	9.1	10.4	3	2.7	1.1	2.3	4	2.2	-1.2	4.5	2.7	5.7	1.9	5	0.1	10.1	0.5
33	7	0.8	-0.3	7.1	6.9	7.9	7.1	4.3	3.5	1	4.3	0.8	-1.2	10.4	4.2	0.4	1.5	4.6	-0.6	9.5	4.7
34	8	0.2	0.3	8.8	5.9	10.6	2.8	-0.1	2.8	0.8	4.5	0.2	-2	4.2	2.6	13.1	4.5	3.1	2	7.6	8.1
35	5	1.4	1.9	6.5	8.2	9	2.5	0.7	3.4	6.5	4.8	1.4	0.1	5.8	3.6	9.3	7	5.4	2.6	6.7	9.9
36	8	1.9	-0	11.2	-0.6	8.9	7	3.8	1.3	5.8	9.3	1.9	-2.2	11.8	4.6	12.5	6.3	1.7	1.6	9.7	15.2
37	7	2.2	-1.3	6.1	8	10.2	4.4	3.7	3.2	0.5	8.2	2.2	2.4	9	3.7	12.9	2.9	1.8	1.9	7.9	14.1
38	5	1.5	0.3	11.8	3.5	4.4	5.8	2.1	1.1	6	10	1.5	0.7	6.8	10.9	9.4	3.2	2.2	-0.3	13.2	13.8
39	10	1.9	0	9.6	-0.8	15.6	4.2	-1.8	2.6	0.4	9.3	1.9	1.3	6.9	4.9	9.3	5.4	4.8	0.7	6.7	1.1
40	5	2.7	1.6	8.9	4.5	4.2	5.7	-0.7	3.1	3.6	5.9	2.7	-1.4	4.9	3.9	5.5	6.2	1.9	0.7	6.4	2.9
41	6	2.1	0.8	9.8	-1.6	3.4	2.7	0.1	2.9	2.8	3.4	2.1	-1.7	11.3	3.2	5.9	1.4	1.4	1.9	3.3	-1.2
42	7	1.8	-0.1	11.2	-0.6	2.6	7.6	4.9	2	4.7	4.1	1.8	-1.5	10.8	2.4	6.6	5.7	5.4	2.8	2.5	-1.3
43	8	0.8	0.3	11.3	9.9	6.2	5.7	-2	2.6	6.8	7.7	0.8	-1.3	10.8	1.7	2.3	6.9	4.5	1.5	7.8	5.3
44	6	0.6	1.5	6.6	2.3	8	3.8	2.3	1.1	1.7	4.1	0.6	-2.5	6.9	7.9	13	4.3	4.1	-0.3	7.1	4.4
45	9	2.2	-1.3	11	3.5	1.6	4.9	2.1	3.1	3.4	9.9	2.2	1.8	7.6	5.9	12.4	6.7	4.9	2.5	8.1	6.9
46	6	2.6	-1.1	7.9	0.9	15.1	4.6	2	3.2	3.4	8.9	2.6	1.3	10.5	4.9	7.8	5.9	5.9	0.1	13.9	6.7
47	5	2.3	-1.4	6.6	1.8	10.4	8.7	0.9	3.4	2.4	10.7	2.3	-1.3	6.9	10.7	8	6.5	4.2	-1.2	4.7	8.1
48	6	2.6	-0.1	6.4	-4.7	5.2	3.6	0.9	3.1	6.8	6.7	2.6	1.7	4.1	6	11.4	3.5	3	-0.7	2.7	-0.5
49	7	1.4	1.4	10.8	2.8	7.9	3.7	0.9	2.8	4.5	10.8	1.4	-2	4.9	1.8	7	4.8	1	2	10.9	1.4
50	5	0.7	-1.3	6.8	-4.9	3	4	-1.7	2.6	6.3	11.4	0.7	0	9.8	8.1	7.2	5.2	3.9	2	8.8	6.6
51	6	1.2	0.9	11	-2.4	11.7	6.7	-0.3	2.7	4.2	3.8	1.2	0.5	8.6	9.4	9.9	5	2.1	2.2	12	7.7
52	10	1.4	0.2	7.9	2	6	2.2	2.2	1.2	3.2	6.9	1.4	-0.9	9.9	8.7	1.6	2.3	1.5	-0.3	7.8	9.8
53	5	1.1	1	11.6	4.2	14.3	7.1	2.4	1.3	2.1	3.8	1.1	1.2	7	7.5	5.6	6.6	3.7	-0.2	11.2	1.3
54	7	0.5	0.5	9	4.9	9.6	8.1	1.2	2.8	1.9	9.1	0.5	-2.5	8.2	7.3	9.8	2.8	3	2.5	11.7	0.4
55	9	1.3	-1	11.2	3	4.1	7.6	2.9	2.7	6.1	6.7	1.3	0.9	7.6	5	7	2.9	4.1	1.8	11.6	0.3
56	9	1.2	-1.4	7.5	-2	1.5	5.8	2.4	2.4	3.4	9.8	1.2	2.6	5.3	4.6	3.5	4.8	4	0	11.3	3.6
57	5	0.3	-0	8.2	8.4	10.5	3.7	1.9	1.4	6.6	5.6	0.3	-1.9	7.2	3.5	7.8	4.5	3.6	0.5	12.7	6.1
58	7	0.6	-1.1	6.9	5.3	1.5	3.6	0.4	2.6	2.2	7.2	0.6	1.9	7.8	2	2.3	2	5.8	2.7	4.4	9.8
59	5	1.7	1	8.2	-1.6	14.1	6.2	-1.5	2.1	5.3	3.1	1.7	1.9	5.4	8.6	13.5	4.3	2.2	2.3	4.4	14.5
60	8	1.1	-0.5	8	1.5	4.1	7.4	0.5	1.1	6.3	5.2	1.1	-2.2	4.1	2.9	1.5	5.1	2.2	0.6	8.2	2.6
61	7	1.5	-0.8	11.9	-3.2	2.2	3.6	2.2	1.4	3.7	6.4	1.5	2.6	9.6	6.8	10.3	5.1	1.7	2.8	12.5	8.7
62	6	1.5	-1	7.6	-2.2	3.7	3.8	1	2.6	2.1	3.7	1.5	-0.9	9.5	2.2	6.8	2.5	1.4	0.9	9.4	-0.3
63	9	1.9	1.3	11.1	-1.4	10.2	2.2	3.5	3.4	3.8	11.1	1.9	-1.2	9.4	7	9.2	6.1	0.9	2.9	9.4	7.2
64	6	1.3	1.8	7.7	2.6	7.3	3.1	2.8	2.5	0.9	9.5	1.3	0.7	11.6	1.8	3.4	5.6	3.6	-1	13.1	11.1
65	8	0.4	1.9	7.5	4.5	8.8	6.5	5.5	1.7	4.1	3.6	0.4	0.9	8.6	9.5	13.8	5.3	0.9	-0.1	10.1	0.1
66	9	0.6	0.5	6	5.3	3.4	4.9	5.9	2	0.3	6.5	0.6	1.1	5.6	10.1	10.3	2.2	0.8	-1	11.5	2.4

Вар.	Размерность, n, ЛПН	Коэффициенты 2-й строки неравенств (дз5.2)										Коэффициенты 3-й строки неравенств (дз5.2)									
		a21	a22	a23	a24	a25	a26	a27	a28	a29	a210	a31	a32	a33	a34	a35	a36	a37	a38	a39	a310
67	7	2.8	1.2	10.2	3.5	3.5	8.5	-1.7	1	0.9	7.3	2.8	1.8	10.3	8.9	3.9	2.8	3.2	0.9	9.6	-2.3
68	9	2.4	-1.4	11.6	8.2	15.1	6.6	-1.3	3.4	2.8	5	2.4	0.4	4.6	3.8	3.7	2.5	1.5	2.5	10.5	13.4
69	6	1.7	-1.1	8.6	7.2	10.8	8.3	2.6	2.6	6.7	4	1.7	1.6	4.2	5	9.3	2.1	2.3	1.8	11	6.6
70	8	1.5	1.2	9.6	0.6	14.5	8.8	2.1	2.8	2.2	11.7	1.5	2.3	10.4	8.4	7.4	3	3.4	-1.2	7.2	7.6
71	9	1.2	0.3	7.9	1.8	6.9	8.5	1.6	1.5	4	7.6	1.2	1.1	4.1	9.2	5.5	1.5	4.8	-0.6	2.4	5.9
72	5	0.4	-1	9.3	4.4	4.8	4.1	-1.5	3.4	3.6	10.6	0.4	1	11.1	7.1	1.6	5.2	4.8	2.9	7.8	2.5
73	9	0.6	1.6	10.4	-2.1	10.6	6.2	5.9	1.3	0.9	4.7	0.6	0.1	7.1	8.9	4.4	6.8	4	0.3	2.9	4.2
74	6	1	-0.6	6.8	9.8	15.5	8.8	1.3	1.5	0.1	5.3	1	-0.9	6.9	4.3	7.8	2.1	0.9	-0.4	2.3	-3
75	8	0.6	1.6	11.7	-0.1	3.3	8	-0.9	2.2	1.5	11.5	0.6	2.3	11.9	8.2	13.8	4.9	3.8	-0.8	8.9	1.8
76	7	0.6	0.3	6.1	5.6	5.6	5.2	4.2	1.6	2.2	5.4	0.6	1.5	4.3	2.8	13.6	6	4.5	-1.2	12.1	1.7
77	6	1.9	0.2	10.2	-2.4	14.2	7.7	1.1	2.8	1.5	7.7	1.9	0.5	8	7.3	12.8	1.9	4	1.1	10.6	-2
78	7	1.4	-1.3	11.4	-0.4	2.1	7.6	4.8	1.7	6.4	6.6	1.4	1.9	10.7	10.7	8.8	5.1	5.9	0.8	3.8	9.2
79	8	2.8	-0.5	6.2	-0.2	14.3	8.5	4.6	2.1	3.7	5.1	2.8	-1	7.6	8.2	6.9	3	2	2.5	6.4	-0.3
80	7	2.7	0.7	9	5.8	5.5	7.5	0.4	1.1	5.6	11	2.7	-2.9	9.4	8.9	4.4	6.7	4.2	-1.4	2.8	2.5
81	9	0.1	1	9.6	0.3	5.7	8.8	-1.8	2.8	3.9	11.6	0.1	-0.7	9	10.6	9.5	4.5	6	-0.3	2.8	-0.8
82	7	1.3	0.9	8.5	-0.2	4	4.8	5.5	2	0.2	9.2	1.3	2.4	11.6	2.8	6	6.9	2.1	0.7	8.4	8.5
83	5	2.3	-0.4	7.7	1.7	11.8	4.5	-1.7	1	3.4	5.2	2.3	-2.3	11.7	4.8	-0.8	5.5	4.3	1.5	13.7	12.9
84	6	0.6	0.3	10.3	0.6	2.8	6.4	-0.6	3.2	4.9	8.1	0.6	2.7	6.3	10.5	8.6	1.7	4.9	-1.1	7	10.3
85	8	1.5	-1	7.2	3.1	7.3	4.6	0.7	1.4	6.6	10	1.5	2.7	10.9	4.2	-0.4	1.6	1.3	0.8	10.2	13.6
86	10	0.7	0.4	6.3	8.7	6.3	2.2	2.9	3.3	4.4	8.2	0.7	-1.6	6.8	1.3	-0.7	6.6	1.7	-0.4	9.9	11.5
87	5	2.1	1.3	7.9	0.8	5.7	7.6	5.9	1.4	4	11.9	2.1	-2.4	5	6.3	1.4	5.6	5.2	1.6	2.2	0.9
88	6	2.5	0.5	11.6	3.2	7.6	3.5	2.4	3	6.4	3.2	2.5	1.7	7.1	5.7	5.4	6.4	3.7	2.4	3.2	6.2
89	7	2.2	-0.9	7.6	-0.4	13	5.1	3.5	1.8	3.1	4.3	2.2	0.4	6.2	4.5	6.5	2.1	4.2	-0.3	11.1	4.6
90	9	0.2	-0.5	7.7	-0.4	15.5	7.3	-0.8	3	6.1	7.1	0.2	-0.8	8.6	1.9	7.8	1.5	2.2	0.4	2.1	-1.2

Продолжение 2 табл. дз5.1

Вар.	Размерность, n, ЛПН	Коэффициенты 4-й строки неравенств (дз5.2)										Коэффициенты 5-й строки неравенств (дз5.2)									
		a41	a42	a43	a44	a45	a46	a47	a48	a49	a410	a51	a52	a53	a54	a55	a56	a57	a58	a59	a510
1	7	3.6	-0.4	7.3	9.2	3.4	2.5	4.1	2	0.7	4.1	3.4	2.8	9.7	6.5	2.5	5.3	3.8	2.3	4.9	8.3
2	5	3.1	0.5	8.8	8.9	5.3	6.6	2.8	3.3	6.4	6.8	2.7	-2.5	12	10.2	7.4	5.4	2.1	3	4.5	3.1
3	7	1.4	0.7	7.9	1.5	12	4.4	-1.8	1.5	0.8	9	3.9	1.7	6	6.2	13	2.4	5.3	0.7	12.2	4.1
4	8	3.7	-2.1	11.2	1.9	0.6	1.8	0.9	1.2	1.4	9.7	1.5	0	8.2	4	6.2	1.3	3.6	1	10.9	15.9
5	10	0.7	-2.7	10.4	7.2	1.8	6.5	4	1.3	0.6	4.5	2.3	-1.1	6.4	8.8	2.7	4	5.4	0.3	2.3	4.9
6	7	3	1.7	4.8	1.4	9.7	7.4	4.2	1.3	6.4	9.8	3.3	0.6	5.8	8.5	-0.3	5.5	1.7	1.3	10.2	6.1
7	6	3.3	-1.8	4.4	8.3	13.2	3.7	5.1	1.6	0.8	9.4	0.7	-0.8	5.6	9.8	1.4	4.3	1.1	-0.6	3.1	11

Вар.	Размерность, n, ЛПН	Коэффициенты 4-й строки неравенств (дз5.2)										Коэффициенты 5-й строки неравенств (дз5.2)									
		a41	a42	a43	a44	a45	a46	a47	a48	a49	a410	a51	a52	a53	a54	a55	a56	a57	a58	a59	a510
8	5	3.7	-0.1	4.9	11.4	8.3	6.2	5.9	1.3	3.3	7.6	2.4	-1.2	7.9	2.8	1.6	5.6	2.7	1.5	10.7	-0.1
9	6	3.4	-1.9	6.2	6.3	4.4	5.1	1.6	2.4	4.3	4.6	1.8	2	8.2	4.4	-0.6	2.3	0.7	-0.8	11.1	0.2
10	8	0.9	-1.4	7.9	13	11.7	1.1	5.3	1.7	5	6.3	3.1	-2.9	7.8	7.7	8.7	4.6	3.3	2.8	11	7.6
11	7	3	0.9	4.5	8.4	12.7	2	4.9	2.6	3	6.3	2.8	-1.1	5.3	10.1	12.1	6.9	4.6	2.5	3.8	-0.4
12	9	0.9	-1.7	9.3	11.4	5.2	3.1	1	2.4	1.9	8.8	3	0.8	11.5	9.4	4.1	1.5	5.6	2.9	5.2	9.4
13	9	1.2	2.3	10.1	1.6	5.3	6.6	5.6	1.5	3.2	8.7	3.1	-2.8	7.9	9.6	6.8	6.4	1.6	2.8	4.7	15.2
14	5	3.9	2.1	5.6	-0.3	13.2	3.2	5.4	1.9	4.3	9	2.9	-2.1	7.5	1.9	2.8	4.3	0.8	0.9	6.5	14
15	6	3	2.1	8.3	8.1	3.4	7.1	-0.5	2.2	1.8	5.8	1.4	-1	5.2	8.5	1.7	5.1	4	0.3	6	15.8
16	7	3.3	-2.6	6.5	13.8	12.6	3.6	-0.5	2	1.2	8.4	2.4	-1.2	5.7	5	5.3	4.6	2.4	1.9	3.8	11
17	10	2.4	-1.3	7.7	9.6	12.5	7.9	1.8	1.7	0.7	8.8	4	0.4	4.1	3.8	12.9	5.2	5.2	-1.2	5	4.2
18	7	3.9	2.4	10.8	7.2	7.5	1.6	1.8	2.2	6.4	4.6	2.9	0.7	5.1	6.1	2.4	6.3	0.8	0.6	3.3	8
19	9	3.3	2.1	7.3	13.7	8.2	2.2	0.6	2.2	1.2	7.8	1.8	0	9.7	7.5	10.2	2.7	4.8	-1.5	7.1	9.7
20	6	3.3	0.4	4.8	5.7	1.1	1.1	1.5	2	2.6	10.6	1.7	2.4	10	2.7	1.2	1.2	5.8	2.8	10.3	9.1
21	8	3.2	-0.5	5.7	5.4	8.6	7.7	3.5	2.5	1.5	11.5	3.8	-1	7.3	2	6.1	1.8	4.5	1.8	10.8	-0.7
22	7	3.2	1.9	10.2	0.2	4.2	3.4	2.6	2.1	4.7	5.7	2.3	1.2	11.6	6	6	1.3	0.8	1.6	6.6	12.9
23	5	1.8	-0.5	8.7	11.5	9.9	7.4	-1.2	1.2	2.6	4.1	0.8	-1.8	7.2	10.9	5	6.6	1.4	1.4	8.5	14.6
24	8	1.5	2.9	5.6	8.1	8.4	6.5	1	1.8	5.1	11.9	1	2.7	7.1	9.7	13.4	4.8	4.6	2.6	6.5	1.7
25	7	0.8	0	7.1	5.4	11.4	3.7	3.3	1.1	4.4	7.6	0.6	-1.3	9.9	2.7	11.4	2	2.1	-0.3	5.7	0.8
26	5	0.9	-1.8	6.5	1.1	4.5	3	4.3	3.4	6.5	9	0.8	2.3	4.6	5	0.8	3.2	5	2.2	7.7	-2.7
27	10	3.2	-1.9	9.6	2.1	7.2	5.2	1	2	3.2	8.9	3.7	0.8	11.3	10.3	4.3	2.3	0.8	2.3	9.1	14.8
28	5	1.1	0.9	7.5	9.8	2	4.2	6	1.7	3	6.7	1.8	2	5.7	7.6	13.6	3.1	2.4	0.1	3.2	5.7
29	9	1.8	2.8	7	8.4	9	1.6	0.6	2.7	0.5	9.7	1.3	0.8	9.4	3.4	1.2	5.5	0.5	1.8	9.5	12.2
30	5	2.3	0.4	7.9	13.6	0.4	2.9	-1.9	2.5	4.9	4.7	2.2	2.5	10.4	6.1	12.4	3.8	5.1	1.9	2.7	1.1
31	6	2.1	-0.3	7.3	-0.8	7.4	4.3	1.2	1.6	0.1	5.4	2.1	-2.7	10.1	8.3	-0.3	3.8	5.5	-0.4	6.5	14.9
32	5	0.8	-2.7	7.2	0.1	14.2	3	-0.8	1.4	1	8.5	2	-2.4	4.8	7.6	8.6	3.6	5.9	0.6	2.7	-1.2
33	7	1.9	-2.8	10.5	6.3	9	6.4	3.7	2.1	2.9	6.2	1.4	2.9	4.6	10.8	9.7	1.6	6	-0.1	5	11
34	8	2.4	1.7	8.6	-0.4	12.4	4.1	5.5	1.7	3.7	8.8	3	0.9	6.6	10.9	13.6	2.4	2	1.3	7.2	7.7
35	5	3.8	0.5	7.5	7.5	13.2	1.5	4.6	3	4.2	5.5	2.7	-0.4	9.2	7.2	8.4	6.3	1	1.8	5.7	7.6
36	8	0.6	0.3	5.6	11	9.2	1.4	0.7	1.2	0.2	4.3	2.1	0.7	10.6	2.3	13.6	5.7	4.2	-1.1	5.1	6.1
37	7	1.4	2.3	6.6	4.9	7	5.4	3.3	2.4	3.7	10.1	3.3	1.7	5.3	8.8	2.5	3.7	3.5	-0.2	4.7	7.8
38	5	3.4	-3	4.7	1.9	3.9	7.9	-1.2	1.7	4.8	11.5	4	-1.1	4.9	4.6	8.1	4	4.6	1.7	5.2	-1.8
39	10	1.3	-2.8	5	10	3.4	5.4	0.1	2.2	3.7	7.7	1.9	2.9	7.4	4.9	-0.6	1.6	0.5	0.1	11.4	9.4
40	5	2.4	1.3	11.1	3.9	7	4.7	-1.9	1.5	0.7	4.9	1	-1.4	10.4	4.2	5.8	6.2	2.5	2.6	9.5	8.9
41	6	3	0.7	11.7	7.8	9.9	3.3	0.4	2.2	6.6	6.5	0.6	1	6.3	1.1	10.5	4.7	1.1	1.4	4.6	3.2
42	7	2.6	-1.1	7.9	4.9	11.2	1.1	3	1.7	6.8	3.3	2.8	1	4.5	7.4	3.6	2.5	3.8	1.9	3.4	-0.7
43	8	3.5	-2.7	9.2	9.8	12.4	7.4	2.1	1.1	1.5	4.7	2	-2.2	9.6	5.6	9.9	4.3	5.9	0.8	3.4	14.5
44	6	3.2	-1.7	7	5.5	12.9	6	3.3	2.4	3.5	10.3	3.4	-2	8	10	6.6	3.1	3.3	-0.8	11.3	-2.1
45	9	1.4	0.7	5.7	10.9	8.2	6.6	-1.9	2.1	6.1	3.6	4	-1.7	6.4	1.7	2.2	1.2	5.6	-0.3	5.4	12.7
46	6	3.4	-2.7	8.4	0.5	2.6	1.3	1.6	1.9	1.3	6.3	3.3	-0.2	5.1	1.9	8.7	4.1	5.3	3	7.4	9.7
47	5	3	2.3	11.3	6.6	11	6.4	1.4	2.8	0.7	6.8	0.6	-2	4.5	8.8	-0.1	2.9	4.6	-0.6	3.9	-2.8
48	6	2.9	-0.8	5.3	13.3	12.9	5.5	1.4	3.3	0.8	8	2.5	1.5	7.1	1.1	8.7	2.3	2.2	2.6	6.7	1

Вар.	Размерность, n, ЛПН	Коэффициенты 4-й строки неравенств (дз5.2)										Коэффициенты 5-й строки неравенств (дз5.2)									
		a41	a42	a43	a44	a45	a46	a47	a48	a49	a410	a51	a52	a53	a54	a55	a56	a57	a58	a59	a510
49	7	2.1	1.3	9.4	5.4	2.6	4.9	-0.8	2.7	0.8	9.1	2	-0.3	9.2	10.6	6.8	4.3	2.8	2	5.7	9.9
50	5	0.9	0.3	5.7	3.4	11.9	2.3	1	1.8	0.9	3.5	1.8	1.8	9.5	7.2	9.7	6.4	5.1	1.8	6.1	14.3
51	6	1.6	1.8	11.8	11.8	4.6	3.2	1	2.2	2.3	7.2	4	0.6	11.3	2.4	2.2	5.3	3.3	0.9	8.4	3.1
52	10	2.4	-0.4	8	0.4	9.2	6.8	0.4	3.2	0.4	4.5	1.9	-1.7	9.1	10.8	2.5	4.7	3.8	2.9	4.5	8.4
53	5	1.2	2.3	6.6	5.7	10.1	5.6	0.5	2.1	2.6	11.6	1.9	-3	4.2	5.8	5.7	5.1	5.8	-1	2.6	13.7
54	7	0.9	1.9	5.8	0	1.4	6.8	2.2	1.4	2.7	8.3	1	-3	7.3	8.3	1.4	4.8	1.8	1.3	11.5	10.6
55	9	3.3	0.1	5.3	13.5	6.3	5.6	0.6	2.3	6.3	6.9	3.4	2.9	11.7	3.5	10.9	3.6	4.2	-0.9	7.4	5
56	9	1.7	-1.1	6.9	4.3	5.6	3.4	-1.7	1.1	4.3	7.4	3.1	1.8	6.4	7.1	11.7	2.8	2.5	0.9	10	4.5
57	5	2.2	0.5	7.6	11.4	5.3	3.5	-0.1	1.4	2.5	4.6	1.8	1.2	8.9	2.1	13.6	4.9	3.4	0.4	10.9	-2.8
58	7	2.4	0.1	6.4	12.7	13.9	1.5	2.9	1.8	4	5.4	3.7	-2.5	11.5	3	12.9	3.7	0.8	1.2	10.2	8.4
59	5	2.4	-0.6	10.3	11	5.3	3.7	2.8	2.6	3.9	9.4	3.5	-0	6.4	5.2	0.5	2.9	1.8	2.7	4.6	-1.6
60	8	1.9	-2.5	10.7	0.8	1.5	4.4	3.6	2.2	6.4	4.7	3.7	0.1	6.2	2.6	3.2	5.2	5.4	-0.3	4	12.4
61	7	3.3	1.5	5.6	8.8	4.2	2.8	0.6	3.3	2.7	5.8	0.6	-2.7	4.2	5.7	4.1	6.7	4.8	-1.4	7.4	-0.9
62	6	3.9	1.7	11.5	7.8	9.7	5.7	3.9	2.7	2.1	4.5	3.5	2.3	11.5	4.3	6.9	6.1	5.6	-0.2	6	4.3
63	9	1.8	-1.1	8.3	13.2	3.9	5	-0.5	1.4	1.7	3.2	3	-2.3	5.2	8.9	12.9	5.6	4.3	2.1	5.4	14.1
64	6	1.5	2.9	7	11.4	11.2	1.8	3.8	3	1.1	3.8	2.8	1	9.7	9.7	1.9	4.7	3	-0.3	13.1	-2.7
65	8	1.1	-0.7	9.4	11.7	9.1	2.5	4.7	3	5.1	11.4	2.9	-2.2	11	5.2	9.5	4.6	5.4	1.1	13.1	-0.4
66	9	0.6	0.6	10.9	8.3	7.4	6.2	4.5	3.3	5.3	10.1	3.8	2.1	4.3	8.9	-1	1.9	2.4	-0.6	10	5.1
67	7	1.8	2.4	6.8	1.6	3.2	3.8	4	1.8	1.3	3.6	2.7	-0.7	5.6	4	-0.2	5.8	4.2	1.3	2.5	9.4
68	9	1.9	-1.9	7.7	11.2	2.6	5	3.7	1.7	3.3	4	0.8	2.3	5.8	2.7	4.1	4	4.8	0.1	7.8	10.5
69	6	3.4	1.6	10.4	6.7	7.4	6.4	3.3	2.3	3.3	8.6	1.9	-1.7	10.7	8.6	5.4	6.8	5.9	-0.9	4	-1.7
70	8	2.1	2	4.8	5.5	13.4	7.8	3.2	2.7	5.9	5.4	2.3	-0.5	5.6	2.7	6.2	1.9	1.4	0.8	2.9	6.5
71	9	2	1.3	5.1	8.1	14.7	4.4	-0.6	2.7	4.6	10.3	1.1	1.7	8.5	2.7	4.1	5.2	1.4	-0.5	3.1	-0.1
72	5	1.8	-2.8	4	9.8	7.7	2.9	4.2	3.2	3.2	10.2	1.5	1.8	8.7	5.9	6	3.6	5.7	-0.4	12.4	3.1
73	9	2.3	-1	9.4	12.7	9	3	3.8	1.8	0.5	4.3	1.9	2.9	8.4	7.4	14	2.5	0.8	1.6	9.1	9
74	6	2.6	1.6	9.3	0.5	4.1	4.3	2.9	2.2	0.6	5.9	3.9	2.6	10.5	1.7	4.7	2.9	0.9	-0.1	11.9	3.1
75	8	1.5	2.2	11.2	6.3	6	6.4	4.9	2.8	2.1	6.7	3.6	2.1	7.1	2	5.9	2.7	2.5	1	5.3	12.1
76	7	3.1	-2.7	7	7.9	7	7	2.6	3	2.2	8.2	2.7	-0.6	5.3	8.4	11.1	5.2	1.2	-0.5	5.5	12.3
77	6	1.6	-2.2	8.9	-0.1	4.4	5.3	-1.9	2	3.7	4	2.9	-0.8	11.6	5.5	7.1	4.4	4.6	0.6	11.5	8.9
78	7	2.7	-0.3	10.9	10.1	6.4	3.3	-1.9	3.1	6.8	9.5	2.5	-0.4	7	9.6	2.3	5.3	3.1	1.9	3.7	9.3
79	8	1.3	-1.3	9.2	7.8	3.6	1.6	2.1	1.6	0.5	9.4	1.5	1.5	4.7	1.5	2.5	6	1.2	2.8	4.8	11
80	7	2.8	-0.2	4.1	7.3	14.1	1.7	5.6	2.7	3.9	7.7	2.8	2.6	10.8	4.1	2.1	6.7	4.6	0.1	9.1	-1
81	9	1.6	2.3	10.8	13.3	5.9	4.7	-0.1	2.1	1	6.1	2.9	-2.6	6.4	5.8	13.9	5	3.9	1.7	13.6	14.7
82	7	1.5	0.3	6.1	6.4	10	6.1	5.6	2.1	0.7	6.5	1.9	1.3	10.6	1.7	-0.1	3.8	5.9	1.3	12.7	-2
83	5	2.4	0.9	10.5	3	13.4	3	2.9	2.3	6.4	7	2.5	0.7	5.3	7.2	8	4.7	0.9	1.6	10	9.4

Вар.	Размерность, n, ЛПН	Коэффициенты 4-й строки неравенств (дз5.2)										Коэффициенты 5-й строки неравенств (дз5.2)									
		a41	a42	a43	a44	a45	a46	a47	a48	a49	a410	a51	a52	a53	a54	a55	a56	a57	a58	a59	a510
84	6	3.5	-0.4	9.4	0.6	7.5	2.4	-0.9	1.7	3.2	10.2	2.4	-0.2	10.3	9	12.3	1.2	2.9	1.2	4.2	0.9
85	8	2.8	1.6	10.1	5	9.2	5.2	-0.7	1.4	6.6	10.8	3.5	0.5	11.3	2	12.4	3.1	0.5	-0.5	13.3	-1.6
86	10	2.1	1.9	5.9	1.6	7.3	5.8	5.6	2.8	5.6	4.7	1.8	-0.6	7.6	5.2	7.5	6.5	4.7	1.1	12.6	2.6
87	5	1.9	2.4	7.2	0.6	9	1.1	2.6	3.3	6.3	3	0.6	-0.1	9.9	2.3	2.4	1.6	2.3	-0.4	3.7	0.2
88	6	2.3	1.8	6.7	0.8	4.7	8	3.4	1.8	6.8	9.9	1.2	-2	8.1	4.6	11.3	3.5	3.2	-0.3	10.6	12.7
89	7	1.6	1.4	10.7	6.7	8.7	7	1.6	3.5	5	7.3	4	3	7.1	7.7	10.7	2.9	3.6	0.1	12.1	5.9
90	9	0.9	-1.4	5.2	6.6	12.5	3.9	-0.8	1.5	3.7	6	2.7	-1.4	7.9	4.8	9.6	3.4	1.8	-1.2	12.3	3.2

Продолжение 3 табл. дз5.1

Вар.	Размерность, n, ЛПН	Коэффициенты 1-й строки равенств (дз5.3)										Коэффициенты 2-й строки равенств (дз5.3)									
		ae11	ae12	ae13	ae14	ae15	ae16	ae17	ae18	ae19	ae110	ae21	ae22	ae23	ae24	ae25	ae26	ae27	ae28	ae29	ae210
1	7	2.4	4	4.1	-4.2	0.1	3.1	3.5	0.1	4.2	5.4	3.2	3	5	-0.6	5.9	6.2	1.5	3.3	2.7	7.1
2	5	0.6	4.4	-0.2	0.3	3.6	7.9	2.3	2.6	9.7	-2.9	3.8	4.3	-0.3	1.8	2.1	4.1	3.7	3.6	2.4	10.7
3	7	4.1	2.8	1.8	-2.6	-1.5	8	2.7	1	6.2	0.3	1.1	3	3.4	-0.4	7.8	0.7	3.3	3.3	4.1	4.1
4	8	6.3	2.1	2.8	-4.8	5.1	2.1	1.6	1.6	7.7	11.8	3.7	3.9	3.5	-0.8	6.2	5.8	2	1.5	1.8	4
5	10	3.6	4.6	2.1	-3.4	1	7.4	3.7	-0	9.1	10.7	2.5	3	-1.2	-2.3	-0.9	-1.6	-0.2	1.2	4.9	10.3
6	7	5	1.4	-2.1	-5.6	4.1	2.3	2.1	2.2	3.7	7.2	4.5	1.6	1.4	-2.2	7.6	-0.9	-0.9	1.8	2.9	8.5
7	6	4.2	3.8	3.1	-4.2	2.7	0.7	1.1	2.6	8.2	8.9	1.1	4	-1.5	2	7.4	4.5	1.9	2.1	4.9	5
8	5	4	4.4	4.1	-1.7	-2.7	2.7	2.1	0.8	4.7	12.6	4.2	1.3	3.3	-0.1	7.4	5.5	0.2	3.6	4.4	3.3
9	6	6.9	4	1.3	0.4	4.2	7.2	1.5	-1.1	7.2	13.7	4.5	3.6	-0.5	-3.9	1.6	5.1	-0.6	3.4	6.8	8.2
10	8	2.6	4.1	2.7	-3.9	-1.8	0.7	1.2	2.9	4.8	4.2	2.4	3.6	4.5	2.1	7.3	4.1	3.4	1.5	6.8	7.8
11	7	2.5	3.7	0.4	-5.6	-1.9	4.3	1.6	-1.2	2.6	-1.2	2.4	2.8	5.9	-0.5	2.3	3.1	-0.1	2.1	5.3	10.2
12	9	2.5	5.2	0.1	-4.1	1.8	7.9	3	-1.4	9.4	4.8	3.3	4.8	3.6	-6.9	-0.1	-1.9	5	2.6	4.9	9.1
13	9	4	2	0.9	-4.5	3.4	6.4	3.5	-0.5	3.5	10.5	4.6	3.9	-0	-3.4	1.3	1.2	5.6	2	5.8	3.4
14	5	4.8	2	3	-2.6	5.8	6.9	2.6	-0.8	4.1	4.2	1.3	3.9	0.7	2.8	6.7	4.4	-0.9	2.6	3	7.3
15	6	4.3	4.7	3.9	-5.6	2.5	6	2.9	2.1	2.2	0.8	5.2	4.8	0.3	-2	4	2.6	4	1.6	6.5	7.6
16	7	6.7	2.1	-2.2	-3.6	-2.4	7	1	1.7	6.3	11.9	4.8	4	2.7	-6.4	6.2	-1.5	1.4	1.5	3.7	9.2
17	10	4.6	2.9	0.5	-2.5	1.8	6.4	2.9	-0.1	8.6	-1.8	2.6	4.7	1.7	-3.1	2.7	8.4	4.8	3	1.9	10.5
18	7	4.8	3.5	2.3	-4.2	-2.1	1.6	3.8	2	6.2	-0.5	5	3.7	-0.2	-6.8	0.9	2.9	5.8	2.8	3.8	10.8
19	9	3.6	5.1	4.2	-0.6	-0.3	2.9	1.1	-0	7.8	4.3	3.4	4.1	-0.1	-6.6	6.7	6.2	4.3	2.3	2.8	10.1
20	6	5.3	3.2	2.5	-0.8	0.3	6.9	2.5	2.6	6.5	10.6	5.3	4.7	2.2	-0.3	1	-0.4	0.1	1.1	2.3	9.2
21	8	3	4.8	0.3	-5.7	4.2	7.1	1.4	0.2	2.2	-1.1	4.9	3.9	3.3	-1.5	7.4	5.8	-0.5	1.8	3.3	9
22	7	2.1	3.2	2.5	-1.7	-1.2	2.8	3.2	2.4	8.5	8.8	1.2	3.3	5.5	-0.2	4.8	8.5	-0.8	1.2	4.5	4.5
23	5	3.6	4.5	-0.2	-3.6	3.9	6.1	3.8	-1.3	5.2	3.8	5.6	1.3	-1.2	2.6	6.2	5.6	3.8	1.5	5.9	3
24	8	1	4.3	0	-1.9	1.7	1	2.4	-0	9.3	12.4	4.6	4.6	-1.1	-1	3.1	2	4.5	2.4	6.1	7.2
25	7	1	2.4	-1.8	-0.4	3.8	5.9	2.6	1.5	4.3	9.8	4.7	4.6	3.3	-0.5	-0.1	1.8	4.9	3.6	2.8	5.6
26	5	6.2	2.4	2.2	-4.7	3.7	6.1	3.5	0.4	9.6	2.8	3.5	3	6.1	-3.8	-0.7	8.3	2.7	2.5	6.3	3.5
27	10	2.7	3.8	3.9	-1.9	0.3	6.2	3.5	1.7	7.8	5.6	5.1	2.6	5.8	-0	6.8	0.7	3.9	3.2	6.5	8.4

Вар.	Размерность, п. ЛПН	Коэффициенты 1-й строки равенств (дз5.3)										Коэффициенты 2-й строки равенств (дз5.3)									
		ae11	ae12	ae13	ae14	ae15	ae16	ae17	ae18	ae19	ae110	ae21	ae22	ae23	ae24	ae25	ae26	ae27	ae28	ae29	ae210
28	5	0.5	2.8	1.1	-4.9	5.7	6.5	2	0.5	9.3	2	3.2	4.3	-1.5	0.7	1	7	3	2.3	2	7.3
29	9	2.1	3.1	-0.5	-0.7	6.4	1.3	1.9	0.2	9.1	-1.5	2.4	3.6	-0.3	-4.2	2.1	2.3	1.6	2.7	1.7	4.7
30	5	5.7	2.5	-2.2	-5.6	2.8	2.1	3.4	0.9	6.4	9.4	4.4	2.8	3.8	-6.2	-0.8	3.3	1.8	3.7	4.8	6.8
31	6	4	2.8	1.2	-2	2.1	7.1	4	0.3	8.2	10.7	2.7	5	4.4	0.8	6.3	4.6	0.3	2.4	4.9	10.6
32	5	0.8	5.7	1.7	-1.8	0.8	-0.4	3.2	0.3	7.8	6.5	2.9	3.5	1.5	-5.8	1.9	5.2	0.9	2.5	6.7	5.2
33	7	3.6	1.2	0.1	0.2	0.4	2.8	1.3	1.7	2.5	9.6	2.3	4.4	3.1	-3.5	7.7	7.8	4.2	3.8	3.8	8.9
34	8	5.1	2.1	-0.3	-4.6	1.8	0.1	2.1	-0.6	4.3	9.6	2.4	4.5	7	-1.9	1.3	5	2.1	2.2	3.1	5.8
35	5	2.3	5.3	-2.5	0.6	3.9	2.3	3.9	2.5	7	8.7	4.7	3.4	2.6	0.7	5.8	0.4	5.5	3.5	1.9	7.2
36	8	6.2	5	5	-4.5	-1.5	0.1	3.1	0.8	5.7	4.1	0.7	1.5	2	-6.2	4.9	5.3	0.2	3.2	1.7	10.9
37	7	5.4	4.7	-1.2	-1.4	3	0.7	3	2.1	9.2	11.4	2.4	2.8	4.7	-3.1	3	7	1.1	1.6	1.9	7.9
38	5	1.9	1.5	4.2	-1.6	-1.6	3	3.8	0.8	8.8	13.2	5.2	4.7	2.7	1.7	1.2	7.7	1.4	3.6	2	5.6
39	10	2.1	3.5	2.2	-0.5	0.5	1.6	2.5	2.4	5.7	-1.3	4.2	3.3	3.3	1.2	5.5	3.5	4.9	1.7	3.2	8
40	5	3.8	2.4	0.3	-3.6	-2.4	-0.2	1.3	0.2	9.6	-2.5	1.3	4.8	1.1	-0.6	7.1	3	5.6	3.6	4.7	4.4
41	6	3.5	1.2	-2	-4	2.7	5.3	1.9	-1.3	9	7.1	3.7	4.5	-0.2	-3.1	2.6	-1.5	4.4	3.2	6.8	4.3
42	7	6.3	6	2.8	-5.4	5	5.9	1.5	2.1	5.6	9.7	3.7	1.7	1.6	-6.8	6.4	-0.3	0.8	1.4	3.4	11
43	8	6.4	4.4	-0.4	-5.6	3.2	1.6	2.3	1.8	5.8	4.9	1.8	2.6	5.5	1.5	7.6	-0.4	1.2	2.4	5.8	7.9
44	6	1.6	5.6	2.3	0.5	6	4.4	1.1	2.2	2.1	11.7	1.6	3.3	-0.5	-5.8	5.8	2	5.9	2.2	6.2	8.8
45	9	0.6	5.7	1.6	-2.6	-1.6	1.2	2.6	-0.2	9.5	4.3	5.9	4.3	0.3	-3.3	0	8.9	5.6	2.4	1.9	5
46	6	6.4	1.8	4.5	-3	-1.8	0.4	3.6	-1.2	9.6	-2.5	3.7	4.1	4.1	2.9	1.2	3.4	-0.7	3	1.5	10.1
47	5	3.1	5.2	-2.3	-0.2	1.4	5	3.6	2.4	4.2	-0.7	4.5	4.9	1.7	-2.9	6.1	-1.6	0.8	3.2	4.6	4.1
48	6	5.2	3.7	0.7	-1.6	-3	0.1	2.6	-0.9	5.7	1.8	4.6	4.7	4.4	-2.2	7.3	7.7	2.4	1.2	3.8	7.8
49	7	1.6	2.9	2.5	-5.9	3.9	6.7	1.5	0.4	4.4	11.6	2.6	5	1.3	-2.6	5.7	7.6	5.5	2.4	6	9.1
50	5	2.6	5.3	-0.8	-2.1	3.7	2.5	3.5	0.2	7.9	6.8	1.7	5	5.3	-6	7.3	1.3	-0.3	1.2	4.8	10.4
51	6	1.7	1.2	1.7	0.5	0.8	2.4	1.3	1.6	9.6	6.8	4.6	2.7	-1	-6.4	-0	7.6	4	2.3	6.7	3.3
52	10	6.1	5.7	1.1	-1.2	3.6	6.7	1.1	-0.6	8	8.2	5.2	3.9	-1.2	-4	3.8	6.6	-0.2	1.6	6	4.4
53	5	5.8	1.2	-1.6	-1.7	-0.4	2.9	1.6	1.9	8.1	12.5	5.1	4	3.8	-1.2	0.4	4.2	0.3	1	2.8	8.7
54	7	0.7	5.5	-2.3	-5.8	-1.7	5	3.9	1	2.2	-2.9	1.1	3.3	5.4	0.8	6	1.5	4.8	1.1	5.8	8.1
55	9	1.6	2	0.4	-3.1	-2.2	3.2	1.9	-0.2	8	-1.7	1.3	2.6	1.1	-2.1	3.8	1.1	-0.2	2.2	6.9	8.2
56	9	5.1	1.8	0.4	-4.4	-2.3	1.4	2.8	-0.9	2.9	0.2	5.3	3.7	1.3	-3.2	2.9	7.1	0.8	2.7	2	5.2
57	5	3.8	1.1	4.6	-0.4	0	4.7	3.8	2.5	8.6	0.2	0.7	2.7	0.7	-6.7	2.1	-0.7	-0.8	3.8	4.6	3.3
58	7	1.9	4.6	-2.8	-4	-2.6	7.7	1.8	1.9	10	3.4	5.7	4.6	0.1	2.6	0.4	-0.9	4.3	1.4	4.9	10.5
59	5	4.1	1.2	-2.1	-2.4	0.1	0.9	2.9	1.5	5.4	-0.4	3.3	1.9	6	-5.2	0.4	5.8	-0	3.6	3.5	3.2
60	8	2.2	2.8	-1.7	-3.2	6.2	1.6	2.5	1	5.7	2	1.3	3.8	4.1	2.8	6.9	-0.8	5.4	3.1	5.5	8.5
61	7	4.6	5.8	0.5	-5.2	-2.5	1.4	1.7	0.4	3.3	6.4	5.4	1.5	1.2	-4.2	3.6	8	4.9	1.4	4.6	10.6
62	6	4.3	1.2	-0.2	-2.4	-0.6	-0.9	2.1	0.8	4.9	4	5.4	2.2	1	-5.8	3.6	0.5	0.8	3.7	3.6	7.3
63	9	2.3	3.8	2	-4	4	7.2	1.2	1.5	3.2	6.3	2	3.9	-0.8	0.2	7	7.1	0.8	1.2	2.4	7
64	6	2.6	2.9	-1.8	0.8	3.4	-0.7	3.1	1	8.9	8.1	5.8	3.5	0.1	-0.6	7.8	-1.5	1.4	1.5	3.4	6.6
65	8	1.5	5.5	0.3	-5.1	2	2.5	3.9	2.7	2.2	4.7	2.1	2.4	2.3	0.2	5.7	1.9	5.3	2.3	4.5	10.6

Вар.	Размерность, n, ЛПН	Коэффициенты 1-й строки равенств (дз5.3)										Коэффициенты 2-й строки равенств (дз5.3)									
		ae11	ae12	ae13	ae14	ae15	ae16	ae17	ae18	ae19	ae110	ae21	ae22	ae23	ae24	ae25	ae26	ae27	ae28	ae29	ae210
66	9	1.3	2.5	3.1	-5.5	5.9	7.1	3.1	0.6	7.8	3	2.7	1.3	3.4	2.2	4.1	-1.6	5	3.8	1.9	10.3
67	7	0.9	1.5	1.7	-0	-3	1.1	3.6	0.1	3	11.4	3.4	2.3	3.7	0.8	3.6	4.5	4.9	2	6.8	3.4
68	9	1.1	5.7	4.9	-1.5	-0.9	1.5	3.1	-0.9	6.8	11.5	3.7	2.8	0.2	-0.2	5.6	8.4	-0.6	1.4	1.5	8.4
69	6	6.4	4.5	0.9	0.5	-2.1	7.5	2.4	-1.4	2.2	11.9	4.9	1.5	-1.3	1.5	2	1.5	4.4	3.6	6.2	6.5
70	8	3.9	4.1	-2.8	-1.4	-1.8	1.2	1.6	1.8	5.6	3.9	5.5	1.5	-0	-1.7	0.8	-1.3	0.8	2.1	4.1	3.8
71	9	1.4	2.1	-2.1	-3.9	-2.4	0.9	2.5	-0.1	7.7	9.7	2.4	1.6	-0.3	-0.7	1.4	8.5	5.3	1.1	6.1	5.2
72	5	3.3	2.7	3.3	0.6	3.8	3	1.5	0.5	9.5	9.1	2.8	2.9	2.9	-4.4	5	1	4.3	3.6	2	10.9
73	9	5.4	1.3	-0.5	-3.1	6.2	0.5	1.4	1.1	7.5	12.1	4.1	3	6.4	-5	5.7	-0.9	2	1.3	2	3.1
74	6	1.6	2.5	-2.5	0.8	0.4	0.3	3.4	2.5	3.5	12	1	2.9	1.4	-2.1	6.7	-1.2	5.4	1	6.5	9.7
75	8	3.7	5.8	-1.2	-4.9	0.8	7.1	3	-0.5	4.4	12.7	4.9	2.3	1.4	-3.4	0.3	-0.2	2.3	3.7	6	10.3
76	7	4.8	5.7	-1.4	-0.6	-0.7	1.7	1.9	0.9	5.5	-0.7	1.8	4.1	4.7	-5.5	0.4	3	3.6	2.8	6.4	8.6
77	6	0.8	2.8	-1	1	3.3	3	1.7	2.6	6.3	11.4	5.4	2.4	-1.1	1.2	0.2	-1.1	5.6	4	6.5	5.3
78	7	6.2	1.9	3.4	-5.5	-1.7	3.8	2.4	0.8	2.8	12	4	4.1	1.7	-3.9	5.6	6.1	0.8	1.2	4.4	7.7
79	8	6.5	5.7	4.4	-2.1	0.1	4.5	1.5	0.4	7.5	6.5	3.9	3.8	4.8	0.1	5.5	5.9	3.5	1.6	1.6	3.3
80	7	0.9	2	4.6	-2.1	6.2	0.7	2.3	1.9	9.5	8.7	4.5	3.1	6.6	-6.7	1.8	2.1	2.9	1.5	5.1	6.8
81	9	0.7	4.8	3.4	-5.2	6.1	7.3	3.2	0.5	9.2	0.5	2.7	4.6	1	-2.8	4.3	8.4	4.4	1	4.3	6.6
82	7	1.2	3.1	1	-3.3	4.9	2.1	3.5	0.6	6.5	8.3	4.5	2.3	0.7	-1.6	2.9	7.4	2.7	3.2	3.6	11
83	5	3.6	5.6	4.1	0.8	5.6	6.3	1.8	1.7	3.9	4.2	2.5	1.5	-1.3	-2.4	5.5	8.2	0.3	2.1	3.6	10.1
84	6	6.6	2.2	-2.5	-5.8	-3	-0.9	3	2.4	3.2	9.9	5	3.6	0.7	-2	3.6	3.3	2.2	1.6	4.2	8.5
85	8	4.4	2.9	1.8	-3.8	3.4	5	2.2	-1.1	3.9	13.2	2.7	4.5	1.4	-6.1	7.2	4.8	4	1.8	5	5.8
86	10	6.4	5.6	4.1	-4.2	1.8	1.3	2.4	1.5	6.9	7.4	1.8	3.5	5.6	-0.5	2.2	2	1.1	1.1	6.6	3.7
87	5	5.3	1.4	3.4	-1.9	-1.5	4.3	3.9	-0.8	5.8	8.8	2.9	1.4	-0.7	-1.4	-0.3	-0.9	4.1	1.7	2	5.7
88	6	3.1	2.3	3.1	-1.4	6.5	5.4	1	1.1	9.4	2.3	3.8	3.4	4.2	-0.4	4.2	3.1	5.3	1.2	5.5	6.1
89	7	6.2	4.2	-2.8	0.1	3.2	3.8	3.3	-1.2	3.6	9	4.3	2.4	6.1	0.2	-0.5	5.5	2.1	1.3	3.2	5.3
90	9	3.5	3.7	2.8	-4.4	-2.6	7.7	3.4	1.6	2.1	11	2	4.3	4.4	-3.2	4.5	6.6	2.4	3	3.1	3.4

Продолжение 4 табл. дз5.1

Вар.	Размерность, n, ЛПН	Коэффициенты 3-й строки равенств (дз5.3)										Правые части неравенств (дз5.2) и равенств (дз5.3)							
		ae31	ae32	ae13	ae34	ae35	ae36	ae37	ae38	ae39	ae310	b1	b2	b3	b4	b5	be1	be2	be3
1	7	6.2	5.8	6.5	-2.6	3.8	6.8	1.7	-1	10.2	-2.2	0.5	-0.2	14.9	1.1	1.3	4.7	2.6	5.3
2	5	5	2.8	0.4	-3.2	-0.7	2.2	4.4	3.4	5.2	0.4	4.3	4.3	5.7	10.5	0.3	4	6.3	6.2
3	7	3.9	3.9	5	-0.6	2.1	3.8	3.1	0.3	5.6	10.3	0.5	3	10.6	7.3	2.6	3.6	5.3	6.9
4	8	6.5	5.2	3.1	-3.9	-2.3	3.6	7	1.5	5.7	-2.1	1.3	1.1	14.7	9.2	-2.4	5.9	3.2	9.6
5	10	0.6	3.6	-0.2	1.6	7.2	3.2	2.8	-1.6	3.4	-5	3.8	-2.1	9.2	11.4	0.5	6	6.6	9.5

Вар.	Размерность, n, ЛПП	Коэффициенты 3-й строки равенств (дз5.3)										Правые части неравенств (дз5.2) и равенств (дз5.3)							
		ae31	ae32	ae13	ae34	ae35	ae36	ae37	ae38	ae39	ae310	b1	b2	b3	b4	b5	be1	be2	be3
6	7	3.7	5.3	-1.7	3.9	-2.2	2.5	5.6	1.2	9.8	13.1	4.3	-2.9	12.4	1.7	5.5	3.4	2.9	9.5
7	6	3.5	3	0.6	-5.5	5	9.2	7.1	1	6.4	1.9	2.5	3.8	14.4	2.8	-1.1	6	5.8	8.5
8	5	4.5	5.5	6	-2.8	-4.9	7.4	1	0.8	9.8	10.2	2.4	2.6	4.2	3.2	7.4	3.6	4.6	8.4
9	6	1.6	4.6	0.4	-3.2	-0.9	2.6	6.7	2	9.1	6.3	3.4	-2.7	11.2	8.8	6.7	3.2	2.5	7.5
10	8	0.9	2.9	-1	-1.5	6.9	1.7	4.2	4.5	3.4	1	1.2	1.2	4.6	9.9	-0.7	5	6.9	9
11	7	6.5	4.8	0.3	-4.8	5.3	10	4.2	-0.3	8.6	-4.5	4	-2.1	11.3	9.5	5.5	3.1	3.7	7.1
12	9	6.9	4.7	6.1	0.2	4.1	6.2	1.9	4.3	8.7	12	4.2	-1.1	8.1	1.6	-0.5	4.6	2.1	6.6
13	9	7.6	2.9	-1.6	0.5	-2	6.3	7.3	2.5	5.8	-1.1	4.3	-0.2	14.4	4.2	4.7	4.1	1.8	9.7
14	5	3.3	5.1	4.6	2.2	3	2.8	7.5	-1.6	8.6	1.4	3	-1.5	6.4	2.3	-2.4	5.2	1.8	5.4
15	6	5.6	2.1	0.9	3.1	0.3	3.7	5.4	3.2	4.6	10.8	2.3	1.3	12.1	4.9	2.5	5	6	7.4
16	7	1	4.3	4.9	0.1	6	6.4	7	-0.9	6.1	2.2	3.2	-0.2	13.9	5.2	-2.2	4.9	3.2	9.5
17	10	6.5	6	-1.6	-5.7	-4.7	8.9	5.7	4.2	9.8	9.5	2.2	1.4	12.1	3.4	-1.6	5.3	2.8	6.5
18	7	1.4	6	-0.3	2.1	6.4	3.4	1.9	3.1	3.9	-2.2	1.5	2.2	4.1	10.5	8.4	4.2	2.6	5.3
19	9	3.5	3.8	5.7	-2.4	-3.5	6.7	2.9	4.8	7.9	3.4	2	-2.7	15.3	10	8.7	4.2	1.2	5.1
20	6	4	4.4	-0.4	2.5	4.4	5.9	2.6	5	5.2	-3.2	0.4	4.9	4.4	5.6	1.4	4.5	2.3	7.9
21	8	2.6	2.9	6.1	-2.8	4.1	1.2	2.1	5	11	3.7	3.2	-1.3	14	5.5	3.1	5.2	5.7	8.5
22	7	5.1	3.8	1	-0.1	4.3	6.4	5.4	-1.9	6.2	2.7	1.6	-2.4	5.3	6.2	3.6	3	6.1	5.1
23	5	7.5	2.7	0.2	-3.8	4.6	1.8	4	-0.2	3.8	5.7	0.6	3.6	8.3	2.3	-0.4	4.9	3.7	6.6
24	8	3.6	5	4.9	-3.4	4	6.9	4.6	1.1	6.1	3.6	5	2.4	8.8	8.8	-2.6	4.6	6.3	9
25	7	2.3	3.6	0.9	3.4	0	6	7.3	4.3	10.6	13.1	3.7	-1.2	6.1	7.9	-1.7	5	1.2	5.5
26	5	1.9	3.5	3.2	-1.1	-1.2	8.3	7.5	-1.2	6.4	0.7	4	-1.7	15.8	12.3	7.4	4.3	5.7	8.5
27	10	7.2	5.7	2.3	0.5	3.3	7.7	4.3	-0.5	9.3	13.5	0.8	2.5	15.7	2.1	2	3.2	5.8	6.3
28	5	2.1	4.6	2.5	-5.4	-2	7.3	7.9	4.4	7.2	9.7	3.4	4.7	4.7	5.5	4	3.9	2.7	5.7
29	9	5.2	3.6	5.5	-2.6	-0.7	2.7	7.2	1	7	-0.7	2	3.9	9.8	8.2	7.8	3.9	5.8	8
30	5	2.9	4.5	0.5	-3.8	-2	2.9	2.3	4	5.7	-1.4	2.9	4.8	11.1	5.5	-0.8	3.4	4.1	7.5
31	6	1.5	2.1	-0.1	-0.2	2.3	2.5	2.7	-0.7	9.8	-1.8	2	2.5	13.1	12.2	3.3	5.3	4.7	8
32	5	3.2	2.3	5.7	-2.6	1.8	7.3	6.7	-1.3	4.4	-1.2	1.4	0.6	11.4	9.8	1.7	5.1	3.9	8.3
33	7	7.4	4.7	0.9	-5.3	2.6	1.1	4.8	2.9	3.5	-3.6	3	-0.4	5.8	12.2	6.4	4.1	3.8	7.1
34	8	2.5	4.3	5.2	1.4	2.1	8.2	2.7	3.5	9.5	11.1	2	-0.6	6.8	5.9	4.4	5.2	5.1	5.8
35	5	7.4	3.2	4.3	-0.4	-3.5	4.4	5.4	1.5	4.8	7.9	4.5	2.3	4.1	12	1.1	5.5	4.5	8.1
36	8	5.3	6	1.2	-3.1	4.9	3.2	4.6	0.2	3.7	0.7	3.4	1.2	8	8.9	3.9	3.3	5.2	5.4
37	7	1.3	4.7	-2	-0.7	-1.6	4.1	2	-1.4	4.1	-1.8	2.6	4.9	9.1	2.6	3.5	3.8	1.8	7.6
38	5	3.7	3.8	1.7	2.2	-1.9	9	2.4	4.8	9.8	1.7	4.3	2.1	7.8	7.4	1	4.1	3.7	5.9
39	10	6.8	3.5	-0	-2.6	6.2	5.8	5.3	-0.8	10.9	13	2.3	-1.6	16	7.7	-0.4	4.8	2.9	8.3
40	5	5.8	3.2	5.8	3.3	7.6	4.4	7.3	2.3	5.5	-0.9	4.9	2.3	16.9	4.4	2.8	5.3	6.7	6.9
41	6	3.5	2.7	5.5	-3.2	1.4	2.7	6.2	3.7	7.3	10.3	3	-1.4	10	10.4	1.5	3.2	2.4	9
42	7	2	3.1	-2.5	-5.2	0.8	4.5	7	0.2	4.6	5.9	1.6	3.7	4.3	9.7	1	4.2	3.5	9.2
43	8	2.3	5.1	0.4	-5.9	-4.5	3	3.6	0.5	5.3	12	1.9	-1.5	10.9	5.2	1.7	5.2	6.1	5.4
44	6	3.2	2.2	1	0.3	2.1	3.1	5.3	1.4	10.8	4.9	0.6	-0.1	12.5	1.1	6.9	3.3	2.9	5.9
45	9	4.2	2.2	4.2	-5.4	6.8	4.3	6.9	-0.9	3.8	-0.5	3.1	3.7	12.9	2.3	-2.7	3.6	6.1	9.6
46	6	5.6	4.3	-0.1	2.3	-1.2	6.6	4.7	2.5	6.4	1.1	2.7	0	14.8	11.8	8.2	3.4	2.7	8.8

Вар.	Размерность, n, ЛПП	Коэффициенты 3-й строки равенств (дз5.3)										Правые части неравенств (дз5.2) и равенств (дз5.3)							
		ae31	ae32	ae13	ae34	ae35	ae36	ae37	ae38	ae39	ae310	b1	b2	b3	b4	b5	be1	be2	be3
47	5	2.8	4.3	0.4	1.7	6.9	4.3	5.1	0.5	10.7	-4.1	3	0.4	14	3.5	9	4.4	5.4	9.7
48	6	3.5	4	2.7	-4.3	0.1	2.8	4.9	-0.8	8.8	7	1.2	2.5	7.6	3	2.4	4.2	5.5	8.2
49	7	3.9	5.2	7	-2.6	6	3.1	6	0.4	4.4	7.6	0.9	-0.3	6.1	3	3.8	5.6	4.9	9.2
50	5	6	4.6	-0.9	-2.1	-0.9	9.3	6.4	-0.1	8	-3.4	4.6	-1.5	7.3	11.5	6.4	5.3	6.3	6.2
51	6	6.8	4.6	4.6	-2.2	4.9	7.8	5.1	0.4	9.3	1.8	4.1	-1.6	15.9	7.1	4.5	3.4	5.7	6.4
52	10	1.7	4.3	-2.7	-2.3	3.5	7.6	5.3	1.9	7.9	12	1	1.3	4.3	5.7	5.3	4.2	5.1	5.4
53	5	2.9	4.6	5.8	0	-3.8	10	3.6	4.5	7.7	10.5	3.3	0.6	4.5	9.6	7.3	5.8	3.1	9.3
54	7	6.9	3.4	-2	-4.7	2.1	6.9	4	0	8.8	2.8	4.4	-1.3	10.6	12.9	4.1	3.6	1.9	5.2
55	9	4.7	4.7	2.3	-4.5	0.7	2.8	1.1	1.9	5.3	6	2.9	4.3	12.7	9.5	3.8	4.7	3.2	5.4
56	9	5.2	2.6	4.8	-1.4	-4	8	2.7	3.8	9.9	-0.2	0.4	-0.2	6.3	7.3	1.7	3.9	1.5	5.5
57	5	1.9	3	4.2	-4.9	-0.8	7.4	0.8	-1.8	5.9	-1.5	2.6	1.5	9.7	9.4	3.4	5.1	6.6	7
58	7	5.3	4.7	-2.5	-5.1	-2.8	1.7	4.8	0.8	6.5	12.6	3.7	0.9	13.3	5.8	2.1	4.5	3.9	5.1
59	5	1.2	4.2	-2	0.5	3.3	4	3.6	-0.1	4.7	1.4	3.3	0.1	15.4	9.1	4.9	5.6	3.6	8.4
60	8	1.8	5.2	6.9	-5.6	-4.1	8.8	5.5	1.9	3.1	6.3	3	2.3	11.2	3	3.3	4.3	2.3	7.1
61	7	0.9	6	6.6	-1.4	-3.1	8.8	1.8	4.5	10.7	11.3	0.6	2.7	16.6	4.8	5.1	5.7	3.4	6.5
62	6	6.5	5.3	-1.6	-4	6.5	6.6	6.3	1.4	9.5	12.9	2.8	4.7	12.6	12.1	5.2	5.4	5.8	5.6
63	9	4.8	4.5	-0.8	3.3	4.3	1.2	5.8	0.3	8.3	9.2	2.8	0.7	15.3	8	5.9	3.5	1.7	5.3
64	6	7.9	3.1	2.6	-0.4	-1.3	8.5	3.1	2.3	9.7	7.8	1.2	-2.3	13.4	12	2.6	4.5	1.2	6.9
65	8	1.8	2.3	3.7	0.1	4.3	6.7	4.5	1.3	4.8	-4.8	5	-2	15.4	10.3	-1.7	5.1	2.2	8.9
66	9	7	2.2	-0	-3.9	5.4	9.8	3.2	2.7	6.3	8.7	2	3.5	7.6	7.7	4.3	5.8	2.7	6
67	7	7	3.8	-0.5	-1.1	-3	4.3	3.3	0	9.2	9	1.2	1.2	10.2	6	6.4	4.2	6.3	7.8
68	9	7.9	4.3	-0.9	0.1	6.4	1.3	5.7	1.8	9.9	7.7	2.9	-2.7	13	2.8	7.1	5.2	2.3	9.6
69	6	1.7	5.9	3.8	0.7	-4.2	7.2	3.8	-1.6	9.9	13.8	4.2	1	15.2	8.2	0.6	3.3	2.7	5.3
70	8	4.8	5.8	3.8	-4.3	-4.4	6.9	7.8	-1.1	4.5	1.9	1.7	0.5	13.1	8.2	3.3	5.1	5.1	7.1
71	9	3.4	3.1	-1.6	2	4	4.7	2.6	-0.7	3.8	13	2.1	0.8	11.2	3.3	3.9	4.8	3.8	6.6
72	5	4.7	4.4	-0.4	0.8	0.5	4.7	7.6	2.6	6.9	2.5	4.5	-2.5	8.3	6.5	-2.3	3.8	5.3	6.6
73	9	4.8	5.2	1	-4.3	7.9	7	6	-0.8	8.5	-4.4	1.1	3.6	10.2	8.7	7.9	3.1	5.1	8.3
74	6	5.5	4	5	-0.4	7	3.8	7.9	1.1	9.8	0.6	4.3	-1.5	16.5	5.9	4.6	5.6	2.2	6.1
75	8	5.3	5.7	-2.5	-1.6	-4.7	4.6	5.3	4.9	7.4	-0.1	4.9	0.3	7.7	9.1	-1.7	5.5	2.8	6.5
76	7	6	3.9	-2.9	0	-0.4	9.6	6.2	3.1	8.6	-2.8	4.5	3	8.5	10	1.9	4.3	1.6	7.7
77	6	2.5	2.1	-0.8	-4.7	3	4	5	2.8	5.6	7.1	3.4	3.7	6.1	8.9	7.9	3.6	6.6	5.4
78	7	5.2	4.7	-1.5	-5.6	4.8	5.8	3.4	2.1	11	2.5	2.2	0.9	7.9	10.5	2.6	4.2	1.7	9.7
79	8	2.1	4.4	2.2	3	-3.2	7.1	4.8	2.2	3.2	12.9	3.3	4.6	16.4	11.4	3.1	4.7	4.5	9.5
80	7	1.1	4.3	4	0.4	-2.9	9.7	4	0.2	7.3	13.5	0.9	-0.6	16.1	11.9	7.1	4.3	3.2	5.3
81	9	2.4	2.3	2.1	-0.4	1.5	2.9	5.8	2.8	8.3	4.7	3.4	-1.7	15.3	7.5	2.3	3.8	6	7.1

Вар.	Размерность, п. ЛПП	Коэффициенты 3-й строки равенств (дз5.3)										Правые части неравенств (дз5.2) и равенств (дз5.3)							
		ae31	ae32	ae13	ae34	ae35	ae36	ae37	ae38	ae39	ae310	b1	b2	b3	b4	b5	be1	be2	be3
82	7	5.7	5	3.1	2.2	-1.2	5.2	2.4	2.2	8.9	9.8	3.7	3.5	5.2	12.9	8.1	5.8	6.9	5.6
83	5	5.2	2.4	0.9	3.8	-0.8	3.2	0.9	1.9	7.9	4.1	4.8	-2.5	10.3	12.4	5.1	3.4	1.6	6.9
84	6	2.4	5.7	-0.9	-4.2	-3.8	8.6	7.8	2.7	11	-3.8	1.3	3.1	16.9	2.5	1.2	3.2	3.8	7.8
85	8	2.5	3.3	0.4	1.5	1.1	1.1	2.5	4	10.2	-1.7	1	4.8	9.6	3	-1.3	5.6	1.6	7.1
86	10	6.9	2.4	-0.2	1.1	5.6	8.2	1.6	3.5	3.7	-2.4	3.4	-1.8	10	4.6	-2.9	3.6	3.4	6.4
87	5	4.8	5	-1	2.5	4.6	9.4	5.6	-0.3	6.5	12.8	3.7	2.5	10.3	1.3	2.6	3.5	3.6	7.8
88	6	0.8	4.4	-0.5	-5.8	2.2	5.6	3.2	0.2	8.9	-2.7	3.1	3.3	4.8	11	8.5	4.2	3.8	5.4
89	7	7.4	5.4	4.3	-5.1	-4.8	5.8	3.2	3.2	4.3	-1.3	3.5	0.3	8.9	11.2	-0.1	4	3.5	5.8
90	9	1.6	5.2	-1.5	1	5.6	6.9	6.3	2.5	9.6	9.6	0.6	-1.8	9.3	5.5	-1.7	5	3.9	7.9

Продолжение 5 табл. дз5.1

Вар.	Размерность, п. ЛПП	Правые части границ ц-области (дз5.4)									
		g1	g2	g3	g4	g5	g6	g7	g8	g9	g10
1	7	6.1	5.8	7.7	4.8	7.9	2.8	6.7	4.5	8.2	7.9
2	5	6.2	8	9.3	3.1	6.1	2	9.4	3.1	3.2	14
3	7	4.2	6.7	8.2	4.4	5.9	1.6	6.5	3.9	3.3	6.3
4	8	4.1	4.7	9.1	4.7	7.5	3.2	7.4	1.8	5.4	12.7
5	10	6.8	3.2	8.4	3.8	5	3.1	8.8	4.3	4.9	11.2
6	7	6.3	7.4	7.1	3.6	6.2	3.7	5.1	1.4	8.1	14.8
7	6	5.8	4	8.5	4.1	6.6	5.9	4.6	1.3	6.5	8.7
8	5	4.1	6.5	9.4	3.4	5.4	1.8	2.1	3.4	6.9	6.8
9	6	4.7	6.1	7.8	4.4	6.7	4.8	3.9	1.5	3.8	16.4
10	8	7.9	4.1	9.3	4.3	6.2	4	11.4	3.4	9.6	6.2
11	7	4.1	4.1	9	4	6	4.6	11.7	3.3	7.3	5.3
12	9	5.3	5.6	9.3	4.1	5	1.8	3.1	3.2	8.9	7.7
13	9	6.5	8.3	8.9	3	6.8	4	11.6	4.9	10.6	11
14	5	5.4	7.3	9.7	3.7	6.3	3.4	11.8	3.7	8.5	15.7
15	6	7.9	3.3	9.1	3.4	7.9	3.8	5.8	4.9	10.7	5.7
16	7	7.5	5.2	7.2	3.9	5	1.1	8.8	3.1	6.5	12.2
17	10	6.1	5.7	8.4	3.6	7	4.8	3.9	3.4	10.2	8.6
18	7	6.5	4.4	9.9	5	7.3	2.4	2.7	3.4	6.3	16.8
19	9	4.5	8.8	8.9	3.6	5.7	5.6	8.3	4.2	6.3	5.9
20	6	4.5	4.2	10	4.7	6	3.4	11.6	2.3	4.9	5.3
21	8	6.8	8.6	7.1	3.5	6.2	5.9	9.1	2.8	3.3	13.5
22	7	5.6	5.2	8	4.9	6.8	4.5	9.2	2.5	4.1	11.6
23	5	5.4	7.4	7.7	4.7	7.9	3.9	6.3	5	3.6	10.7
24	8	6.9	7.7	7.9	4.7	7	1.7	7.5	1.4	5	10.1
25	7	4.9	3.2	7.6	3.5	5	5.4	2.1	2.2	7.5	5.3

Вар.	Размерность, п.ДПП	Правые части границ ц-области (дз5.4)									
		g1	g2	g3	g4	g5	g6	g7	g8	g9	g10
26	5	7.2	5.4	8.3	4.1	5.4	1.7	9.2	4.2	5.5	13
27	10	5.8	4.7	9.7	3.6	5.4	2.4	11	3.6	7.5	5.2
28	5	6.2	3.9	8.8	3.1	6.7	1.6	2.6	4.5	9.1	14.1
29	9	5.3	8.2	8.3	4.5	5.5	3	11	2.4	7.8	15.6
30	5	7.7	7.9	7.7	4.4	6	4.2	2.1	4.5	9.7	10.3
31	6	6.3	7.4	8.7	4	7.2	1.7	2.6	3.7	7.3	15.6
32	5	6.8	6	7.6	4.3	6.8	1.5	5.8	2.3	8	9.7
33	7	6.6	7.3	10	3.6	6.9	1.2	10.1	4.9	7.6	7.4
34	8	4.3	6.3	8.4	3.3	5.6	1.8	5.1	1.5	8.6	14
35	5	6	4.1	9.2	4.4	5.3	4.4	11.2	4.2	10.1	11.3
36	8	7.1	6.7	9.2	5	5.1	1.6	7.1	4.9	5	7.8
37	7	5	7	7.9	3	6.7	4	9	3.1	5.2	13.9
38	5	5.6	7.2	8.8	4.1	8	5.1	4.5	3.8	5.7	12.6
39	10	5.6	8.5	8.5	4.8	7.9	2.4	3.8	3.8	8.2	5.7
40	5	4.1	9	8	4.5	5.3	5.7	3.3	3.6	5	9.8
41	6	4.7	7.6	8.6	3.4	5.7	4.1	7.6	4.4	7.1	9.9
42	7	5.6	3.5	8.2	4.8	6.9	3.2	4.4	4.4	7.3	8.4
43	8	4.9	6.4	7.4	4.9	5.7	3.8	8.3	4.3	6.2	8.5
44	6	4.3	7.1	8.5	3	7.8	6	8.6	4.3	8.5	11.2
45	9	6.8	7.3	9.9	3	6.5	2.8	8.2	3.5	6	5.3
46	6	7.2	4.4	9.5	4.6	5.2	5.9	10.7	4.3	3.3	13.2
47	5	6.9	6.4	7.3	3.7	5.2	1.4	2.7	1.6	6.4	9.7
48	6	4.7	6.4	8.2	4.3	5.8	1.7	2.8	1.3	10.6	13.1
49	7	4.8	5.6	9.9	3.1	5.3	3.8	3.1	2.1	4.3	15.2
50	5	5	3.5	7.1	3.7	5.4	3	2.2	2.5	8.7	15.4
51	6	4.6	5.4	7.1	4.3	7.8	5.6	11.2	4.9	6	7
52	10	4.5	5.6	7.3	3.7	6.1	5	4.6	5	5.1	17
53	5	6.2	8.6	8.8	4.3	5.2	4	8.5	1.5	10.5	12.9
54	7	6.5	5.1	8.7	3.8	5.4	2.2	7	2	5.7	14.5
55	9	6.1	5.9	9.9	3.5	5.5	4.9	6.8	2.3	8.4	16.7
56	9	7.7	6.4	7.4	4.3	5.5	5.5	11.7	4.6	9.6	15
57	5	5.2	3.4	9.6	3.8	6.1	3.7	5.3	4.6	8.2	10.6
58	7	7	5.4	8.5	4.6	7.5	5.3	5.9	3.8	4.1	14.9
59	5	6.2	7	8.3	3.1	7.8	3.9	9.3	3.4	6.6	8.1
60	8	7.7	3	9.9	3	6.3	2.5	7.4	4.7	7.1	11.8
61	7	5.5	8.1	8.4	4.6	7.1	3.6	5.5	2.4	10.5	15.6
62	6	4.4	8.1	7	4.8	5.8	2	2.1	3.7	8.8	15.9
63	9	4.4	4.4	7.6	4.6	5.8	5.7	10.2	2.1	4.1	13.7
64	6	6.3	5.7	9.1	3.2	5.1	1.4	3	3.3	5.7	5.6

Вар.	Размерность, n, ЛПН	Правые части границ ц-области (дз5.4)									
		g1	g2	g3	g4	g5	g6	g7	g8	g9	g10
65	8	5.6	4.4	8.3	3.3	5.4	1.3	4.7	2.3	7.4	5.9
66	9	7	7.8	8.8	3.2	5.3	2.1	6.4	5	10.3	8.4
67	7	5.8	7.1	8.3	4.6	6	1.6	3.8	4.9	8.9	6.1
68	9	4.1	7.4	9.7	3.2	6.4	1.5	6.1	3.8	6.1	13.3
69	6	6.6	7.7	7.1	3.7	6.1	4.6	11.9	4.7	5.9	9.5
70	8	5.1	6.2	10	4.2	7.3	1.1	9.3	1.6	10.4	15.6
71	9	4.3	3.6	9	3.8	6.4	3.9	4.1	2.9	7.9	11.4
72	5	4.2	8.7	8.8	3.7	5.8	3.5	6.6	1.4	6.9	6.8
73	9	7.1	7.8	9	3.1	6.5	3.9	4.2	2.7	3.4	10.8
74	6	8	3.6	9.2	3.1	7.8	3.4	9.3	3.2	6.4	8.9
75	8	7.8	3.5	8.2	3.5	7.8	2.2	8.8	4.1	10.9	13.5
76	7	6.9	4.6	7.6	4.6	8	4.5	7	2.6	6.4	12.3
77	6	7.5	3	7.1	4.2	6.9	2.5	9	3.9	5.9	9.2
78	7	4.5	4.6	7.5	4.5	7.2	4	7.2	3.7	3.1	11.7
79	8	6.8	5.2	7.2	4.4	5.1	2.3	8.5	2.6	3.5	9.7
80	7	7.6	4.2	9.7	4.9	7	5.9	10.7	2.7	10	14.1
81	9	5.5	3.7	8	4.1	7.4	4.5	11	1.6	3.3	12.5
82	7	5	4.7	7.7	3.8	7.5	3.4	6.4	4.1	6.7	6.3
83	5	4.7	3.1	7.3	4.2	5.5	4.2	3	4.5	8.6	14.6
84	6	4	7.3	9.1	4.3	5.2	4.7	5.4	1.5	9.9	9.3
85	8	6.3	7.5	9	4.8	7.4	1.3	7.4	4.4	5.7	13.8
86	10	6.8	4	9.8	4.3	7.3	4.8	10.8	4.1	7.2	16.2
87	5	7.2	4.9	7.2	4.5	6.1	2	3.4	2	9.4	10.8
88	6	6.6	7.9	7.8	4.7	6.4	3.2	3.1	4.2	10.2	10
89	7	6.8	3.8	8	4.8	6.8	1.6	7.4	1.8	4.8	12.8
90	9	5.6	6.8	7.8	3.2	5.5	5.8	6.7	3.5	6	16.7

Таблица дз5.2
Данные для линпроганизации с размерность 1000...8000

Вариант	Размерность, n (1000...8000)	Число нену- левых строк в Ao, n1 (80...500)	Число ненуле- вых столбцов в Ao, n2 (30...200)	Число нену- левых строк в Aeqo, n3 (200...600)	Число нену- левых столб- цов в Aeqo, n4 (400...800)
1	4.0*1000	397	54	246	455
2	3.3*1000	393	52	546	591
3	7.4*1000	438	72	514	671
4	6.8*1000	150	70	258	634
5	2.1*1000	256	142	580	457

Вариант	Размерность, n (1000...8000)	Число нену- левых строк в Ao, n1 (80...500)	Число ненуле- вых столбцов в Ao, n2 (30...200)	Число нену- левых строк в Aeqo, n3 (200...600)	Число нену- левых столб- цов в Aeqo, n4 (400...800)
6	4.0*1000	348	138	267	798
7	3.4*1000	242	176	473	735
8	5.7*1000	397	193	207	713
9	4.7*1000	238	118	274	519
10	6.9*1000	425	57	271	595
11	1.5*1000	486	189	382	705
12	2.1*1000	195	183	458	626
13	2.7*1000	348	186	453	591
14	3.9*1000	350	105	329	610
15	3.8*1000	402	164	529	469
16	7.6*1000	338	194	494	509
17	3.4*1000	243	167	477	566
18	5.9*1000	167	199	424	681
19	1.8*1000	95	184	438	475
20	4.0*1000	332	163	220	425
21	7.9*1000	267	177	407	573
22	2.1*1000	346	98	532	452
23	4.3*1000	238	166	458	742
24	2.8*1000	311	86	264	689
25	6.4*1000	235	67	515	712
26	3.8*1000	472	79	248	629
27	6.6*1000	362	106	401	526
28	5.1*1000	280	63	530	445
29	6.8*1000	156	145	251	649
30	1.0*1000	295	177	260	476
31	6.5*1000	334	100	432	633
32	6.7*1000	459	106	600	494
33	1.9*1000	377	144	255	647
34	2.5*1000	204	52	478	773
35	5.1*1000	428	127	210	785
36	1.7*1000	201	56	381	767
37	3.9*1000	314	163	225	464
38	4.7*1000	288	144	411	611
39	5.6*1000	393	105	273	432
40	3.7*1000	128	112	213	456
41	7.3*1000	382	186	225	484
42	3.0*1000	488	53	590	450
43	2.1*1000	467	60	479	616
44	1.8*1000	320	113	416	786
45	3.6*1000	409	38	256	711
46	1.1*1000	185	73	251	615
47	7.7*1000	129	53	298	488
48	4.6*1000	443	111	293	448
49	2.0*1000	409	145	463	407
50	6.6*1000	141	130	454	452

Вариант	Размерность, n (1000...8000)	Число нену- левых строк в Ao, n1 (80...500)	Число ненуле- вых столбцов в Ao, n2 (30...200)	Число нену- левых строк в Aeqo, n3 (200...600)	Число нену- левых столб- цов в Aeqo, n4 (400...800)
51	3.6*1000	479	54	544	468
52	5.0*1000	150	199	584	466
53	4.4*1000	366	172	406	781
54	5.3*1000	87	163	263	526
55	2.4*1000	157	81	573	461
56	5.1*1000	155	113	561	534
57	4.9*1000	271	159	568	725
58	1.4*1000	351	126	377	674
59	1.7*1000	345	111	514	723
60	3.2*1000	215	119	587	785
61	5.0*1000	425	59	522	410
62	8.0*1000	388	76	597	709
63	5.6*1000	371	101	563	572
64	7.5*1000	315	149	513	622
65	5.8*1000	330	62	547	718
66	5.2*1000	101	41	360	566
67	6.6*1000	298	103	520	455
68	2.9*1000	429	52	332	591
69	2.1*1000	351	175	287	671
70	2.0*1000	147	153	315	634
71	4.5*1000	411	163	379	457
72	1.2*1000	130	127	278	798
73	7.8*1000	291	84	470	735
74	3.7*1000	427	49	546	713
75	2.7*1000	134	136	365	519
76	4.2*1000	143	62	379	595
77	1.1*1000	324	129	467	705
78	4.5*1000	500	70	252	626
79	7.8*1000	138	135	429	591
80	5.0*1000	372	188	261	610
81	7.6*1000	90	194	514	469
82	6.9*1000	270	186	469	509
83	6.7*1000	106	57	377	566
84	2.6*1000	301	120	393	681
85	5.4*1000	156	43	566	475
86	1.5*1000	93	54	253	425
87	3.8*1000	107	66	270	573
88	5.2*1000	490	51	396	452
89	3.1*1000	373	122	219	742
90	1.7*1000	307	194	302	689

Таблица дз5.3
Данные для ЦФ $y_{42}(x,p_1)$ и $y_{43}(x,p_1,p_2)$

Вар.	Размерность, n ЛПН	y42(x,p1)				y43(x,p1,p2)			
		p1 и p2 вставляются в f, A и Aeq во все члены, где присутствует компонент с p1 или с p2. Конечные значения параметров p1k и p2k не должны превосходить величин 5...9. Шаг параметра shp=(pk-pn)/(mp-1)							
		Кол. значений p1, mp1	Нач. значение p1, p1n	Конечное, p1k	№ комп. x для умнож. на p1	мин – 0 макс – 1	Кол. значений p2, mp2	Начальное значение p2, p2n	№ комп. x для умнож. на p2
1	7	7	0.5	8	5	0	6	0.6	6
2	5	8	0.3	7.5	1	1	6	1	4
3	7	8	0.7	8.5	3	0	3	0.6	7
4	8	9	0.2	7.9	3	0	4	0.5	8
5	10	5	0.2	8.1	5	1	5	0.8	9
6	7	5	0.9	3.5	4	0	6	0.9	6
7	6	4	0.7	7.6	1	1	5	0.7	6
8	5	6	0.3	8.5	1	0	5	0.8	4
9	6	4	0.6	6.2	2	1	6	1	6
10	8	7	0.4	7.9	5	0	4	0.7	6
11	7	9	0.3	8.8	1	0	3	0.7	5
12	9	8	0.8	6.4	2	1	6	1	7
13	9	6	0.5	6.4	2	0	5	0.8	6
14	5	8	0.5	8.7	3	1	5	0.6	4
15	6	5	0.6	6.8	3	0	5	0.5	6
16	7	7	0.4	7.4	1	1	4	1	5
17	10	6	0.8	7.9	2	1	5	0.6	5
18	7	8	0.6	7.2	2	1	5	0.7	7
19	9	6	0.9	4.7	4	0	4	0.6	6
20	6	9	0.3	6.2	1	1	4	0.6	5
21	8	6	0.2	4.8	3	0	5	0.6	4
22	7	4	0.1	5.2	3	1	5	0.6	7
23	5	5	0.5	6.6	3	1	5	0.7	2
24	8	6	0.1	6.4	4	0	5	0.9	1
25	7	7	0.8	3	4	0	5	0.6	7
26	5	4	0.3	8.5	3	0	6	0.7	5
27	10	7	0.5	3.5	2	1	6	0.9	6

Вар.	Размерность, n ЛПН	y42(x,p1)				y43(x,p1,p2)			
		p1 и p2 вставляются в f, A и Aeq во все члены, где присутствует компонент с p1 или с p2. Конечные значения параметров p1k и p2k не должны превосходить величин 5...9. Шаг параметра shp=(pk-pn)/(mp-1)							
		Кол. значений p1, mp1	Нач. значение p1, p1n	Конечное, p1k	№ комп. x для умнож. на p1	мин – 0 макс – 1	Кол. значений p2, mp2	Начальное значение p2, p2n	№ комп. x для умнож. на p2
28	5	6	0.3	6.6	3	0	4	0.8	5
29	9	6	0.4	3.9	4	1	5	0.9	9
30	5	4	0.1	3.1	3	0	3	0.8	4
31	6	7	0.6	4.1	2	1	6	0.8	6
32	5	4	0.2	3.5	5	0	5	0.6	2
33	7	4	0.5	4.6	5	0	5	0.6	1
34	8	9	0.1	8.1	4	1	4	0.6	7
35	5	3	0.2	3.1	4	0	4	0.7	5
36	8	8	0.7	4.2	2	0	3	0.7	6
37	7	3	0.8	6.9	3	1	5	0.6	7
38	5	9	0.8	5.8	2	1	4	0.7	4
39	10	6	0.6	8.4	3	1	3	0.8	7
40	5	7	0.2	6.5	3	0	4	0.6	2
41	6	7	0.7	7.5	1	1	6	1	4
42	7	4	0.2	7.9	3	0	4	1	7
43	8	8	0.2	4.5	1	0	5	0.9	6
44	6	7	0.8	6.4	4	0	3	1	6
45	9	8	0.5	8.3	4	1	5	1	8
46	6	9	0.3	4.9	1	1	4	0.6	5
47	5	4	0.4	5	2	1	5	0.7	5
48	6	4	0.5	4.1	5	0	6	0.9	3
49	7	6	0.5	7.7	4	1	4	0.7	1
50	5	6	0.5	4.1	1	0	6	1	4
51	6	8	0.5	6.3	1	0	5	0.8	3
52	10	5	0.3	7.3	3	0	4	0.6	8
53	5	8	0.7	6.7	1	1	5	0.8	4
54	7	8	0.7	4.7	3	0	5	0.8	6
55	9	4	0.5	4.7	1	1	5	0.6	8
56	9	8	0.5	8.5	2	0	4	0.5	7
57	5	4	0.6	8.1	2	0	5	0.8	5
58	7	8	0.5	6.4	2	0	4	0.5	6
59	5	5	0.4	6.1	3	0	5	0.9	4
60	8	4	0.9	7.3	1	0	4	1	6

Вар.	Размерность, n ЛПД	y42(x,p1)				y43(x,p1,p2)			
		p1 и p2 вставляются в f, A и Aeq во все члены, где присутствует компонент с p1 или с p2. Конечные значения параметров p1k и p2k не должны превосходить величин 5...9. Шаг параметра shp=(pk-pn)/(mp-1)							
		Кол. значений p1, mp1	Нач. значение p1, p1n	Конечное, p1k	№ комп. x для умнож. на p1	мин – 0 макс – 1	Кол. значений p2, mp2	Начальное значение p2, p2n	№ комп. x для умнож. на p2
61	7	4	0.6	5.6	2	0	4	0.5	5
62	6	7	0.4	5	2	0	4	0.6	6
63	9	9	0.2	8.4	3	1	4	1	8
64	6	5	0.5	7.9	2	1	5	0.5	6
65	8	4	0.8	8.2	3	1	5	0.5	8
66	9	5	0.7	8.1	4	1	5	0.7	9
67	7	8	0.1	6.8	5	1	4	0.6	2
68	9	9	0.6	7	4	0	4	0.8	8
69	6	6	0.8	7.9	4	0	4	0.7	1
70	8	6	0.5	8.1	5	1	5	0.7	6
71	9	4	0.8	7.7	3	1	5	0.8	6
72	5	5	0.1	3.9	4	0	5	0.9	3
73	9	7	0.8	4.5	3	1	4	0.8	8
74	6	6	0.6	5.5	5	0	5	0.9	1
75	8	6	0.3	5.4	4	0	5	0.8	8
76	7	6	0.7	8.7	3	0	3	0.6	5
77	6	7	0.1	5.1	2	0	6	0.8	5
78	7	5	0.9	7.2	5	1	4	0.8	7
79	8	4	0.6	3.7	5	0	4	0.7	7
80	7	5	0.2	5.6	2	0	4	0.9	6
81	9	7	0.8	8.9	4	0	5	0.7	6
82	7	3	0.6	3.9	5	0	3	0.8	4
83	5	9	0.1	5.8	2	1	5	0.9	5
84	6	9	0.7	4.6	3	1	3	0.7	6
85	8	6	0.6	7.6	2	0	5	0.7	7
86	10	9	0.5	5.4	1	0	5	0.6	5
87	5	7	0.1	7.8	2	1	3	0.6	5
88	6	4	0.9	6.5	3	0	4	0.6	6
89	7	5	0.4	8	2	1	3	0.7	6
90	9	7	0.3	7.3	4	0	5	0.9	9

6. Самостоятельное решение тематической линпрогзадачи из разд.7

К тематическим относятся задачи планирования (производственного, агротехнического, строительного, торгового, инвестиционного и т.д.) и транспортные. Задачи сформулированы на проблемном языке, включающем специфическую терминологию, наборы численных данных и условий, критерии эффективности искомого плана или организации дела.

При решении линпрогзадачи необходимо выполнить следующее.

1. Создать папку «LinprogZadFamilija(vasha)N» (N – номер варианта задачи), в которую следует помещать все файлы, связанные с выполнением задачи. Создайте в папке мат-файл «WS_ZadFamilija(vasha)N.mat», Уорд-файл, присвоив ему имя: «ЛинпрогзадачаФамилияN».

Для этого откройте Матлаб. В поле «Current Directory» вставьте полное имя папки (путь к ней), используя кнопку поиска папки , сделав её Действующей. Создайте объекты среды для решения линпрогзадачи.

Файл «WS_ZadFamilija(vasha)N.mat» (файл Вычпространства задачи): после открытия папки откройте меню «Desktop» и поставьте галочку в строке «Workspace». В открывшемся окне Вычпространства (Workspace) командами меню «File_Save Workspace As» сохраните пустой файл с указанным именем. В нём сохраняются имена и значения всех вычислительных объектов расчёта, за исключением файлов.

Уорд-файл создаётся в дополнительно открытом Уорд-редакторе.

2. Скопировать в Уорд-файл текст Приложения 1 как шаблон Отчёта решения задачи, вставив данные Исполнителя и текст его задачи на место имеющихся, не удаляя рубрик после текста задачи.

3. На основе текста линпрогзадачи сформулировать её математическую постановку в тексте скопированного Приложения 1 в виде набора выражений (2.8)...(2.11) с наименованием в терминах линейного программирования отдельных объектов задачи, вставляя элементы набора в готовые рубрики А)...Д) шаблона Приложения 1 на места имеющихся объектов. Отсутствующие линпрогобъекты следует также именовать, присоединив к имени примечание «нет».

Такая запись помогает уяснить сущность задачи оптимизации и фиксирует её трактовку Исполнителем. Дело в том, что для успешного решения конкретной задачи может потребоваться изменить неравенства на равенства или часть неравенств включить в число равенств (и наоборот), входя в кажущееся противоречие с исходной постановкой линпрогзадачи. Фиксация в тексте этапов таких манипуляций с данными позволяет избежать случайных ошибок и ошибок неправильного распоряжения данными.

4. Сформировать в рубрике Е) шаблона на основе рубрик А)...Д) в терминах Матлаба «Условия для вставки в линпрогуказание решения задачи».

5. Скопировать в папку из электронной версии Монографии тексты линпрогуказания LinProgUnivUkz_Tab (рис. 2.12) и линпрогооптимизатора LinProgUniversAvtomatMU_Tab (рис. 2.13).

6. Создать шаблон итогтаблиц линпрогминимума/ линпрогмаксимума (в зависимости от требований Задачи) ItogTabLPrmi.rpt/ ItogTabLPrma.rpt согласно разд. 2.4.4.2, но не меняя условий оптимизации в скопированном указании LinProgUnivUkz_Tab.

7. В указании LinProgUnivUkz_Tab закомментировать весь разд.2 «Аргументы ...» и вставить в него объекты из п. Е) Отчёта, за исключением команд Lmi_Lma. Раскомментируйте в указании требуемый тип оптимизации согласно п. Е) и закомментируйте противоположный. Запустите линпрогуказание: Матлаб выдаст html-итогтаблицу вида той, что показана в п. Ж) Отчёта.

8. Проанализируйте полученную таблицу. Должно быть (см. таблицу в п. Ж) Отчёта): $efm=1$. Сопоставьте полученный результат с условиями текста задачи на предмет неучёта какого либо условия или неправильного его включения в тот или иной разряд ограничений. Чтобы не вышло так, что решение найдено, но оно формально и не учитывает полностью условия задачи, т.е. является неправильным. Если решение правильно, то можно приступать к дальнейшему составлению Отчёта.

9. Если $efm \neq 1$, то следует вернуться к составлению условий задачи, к пп. А)...Д). Рассмотреть ещё раз: правильность назначения неизвестных в п.А), справедливость и полноту ограничений-неравенств и равенств (в пп. Б)...Г)), правильность назначения границ ц-области в п. Д). Может попробовать изменить неравенства на равенства или часть неравенств включить в число равенств (и наоборот).

10. Если решение Матлаба правильно, то следует скопировать полученную итогтаблицу в п. Ж), заменив

ею имеющуюся в шаблоне Отчёта, и отформатировать её так, чтобы она имела вид помещённой в шаблоне таблицы.

11. Скопировать 1-й и 2-й столбцы таблицы в п. 3) и вставить названия компонент вектора x и величин y в соответствии с терминологией Задачи. В последних 2 строках п.3) следует заменить названия ограничений согласно решаемой задаче.

12. По итогам решения сформулировать Выводы в п. И), оформив их согласно примеру из Шаблона Отчёта.

13. В рубрике К) Приложения следует откорректировать названия файлов в пп. 3...6 согласно именам файлов в действующей папке Задачи.

7.3 А Д А Ч И

1. Машиностроительное предприятие.

Для изготовления четырех видов продукции использует токарное, фрезерное, сверлильное, расточное и шлифовальное оборудование, а также комплектующие изделия (И.Л. Акулич) [5, с.6]. Кроме того, сборка изделий требует выполнения определенных сборочно-наладочных работ. Нормы затрат всех видов ресурсов на изготовление каждого из изделий приведены в табл. 1. В этой же таблице указаны наличный фонд каждого из ресурсов, прибыль от реализации единицы продукции данного вида, а также ограничения на возможный выпуск продукции 2-го и 3-вида.

Таблица 1

Ресурсы	Нормы затрат на изготовление одного изделия каждого вида				Общий объем ресурсов
	I	2	3	4	
Станочные работы, человеко-ч:					
токарного	550	–	620	–	64 270
фрезерного	40	30	20	20	4 800
сверлильного	86	110	150	52	22 360
расточного	160	92	158	128	26 240
шлифовального	–	158	30	50	7 900
Комплектующие изделия (шт)	3	4	3	3	520
Сборочно-наладочные работы (человеко-ч)	4.5	4.5	4.5	4.5	720
Прибыль от реализации одного изде- лия (руб.)	315	278	573	370	–
Выпуск (шт.);					
минимальный	–	40	–	–	–
максимальный	–	–	120	–	–

Найти план выпуска продукции, при котором прибыль от ее реализации максимальна. Величина прибыли?

2. Предприятие выпускает радиоприемники трех различных моделей.

Модель *A*, модель *B* и модель *C* (Е.Н. Бендерская и др.) [1, с.25]. Каждое изделие указанных моделей приносит доход в размере 80, 150, 250 руб соответственно. Необходимо, чтобы предприятие выпускало не менее 100 приемников модели *A*, 150 приемников модели *B* и 75 приемников модели *C*.

Каждая модель характеризуется определенным временем, необходимым для изготовления соответствующих деталей, сборки изделия, сдаточных испытаний и упаковки. Так, в частности, в расчете на 10 приемников модели *A* требуется 3 ч. для изготовления соответствующих деталей, 4 ч. на сборку и испытания и 1 ч. на упаковку. Соответствующие показатели на 10 приемников модели *B* равняются 3.5, 5 и 1.5 ч., а на 10 приемников модели *C* – 5, 8 и 3 ч.

В течение ближайшей недели фирма может израсходовать на производство – 155 ч., на сборку и испытания – 230 ч. и на упаковку – 70 ч.

Составить производственный план на неделю, обеспечивающий максимальный доход предприятию. Величина дохода?

3. Автозавод выпускает две модели автомобилей.

Модель *A* и более дешевую модель *B*. На заводе работает 1000 неквалифицированных и 800 квалифицированных рабочих, каждому из которых оплачивается 40 часов в неделю (Е.Н. Бендерская и др.) [1, с.29]. Для выпуска модели *A* требуется 30 ч. неквалифицированного и 50 ч. квалифицированного труда. Для модели *B* – 40ч. и 20ч. соответственно. Каждая из моделей требует затрат на сырье в размере 500 руб и 1500 руб. Суммарные затраты не должны превышать 900 000руб. в неделю. Цех отгрузки продукции работает 5 дней в неделю и

может отгружать с завода не более 210 машин в день. Каждая модель A (B) приносит прибыль 1000 (500) руб. Каким должен быть объем выпуска каждой модели, максимизирующий прибыль автозавода?

4. Цех выпускает 5 деталей.

Обработка их может быть выполнена на различных станках (см. табл. 4). Требуется определить план выпуска деталей на станках такой, чтобы получать в час максимальное число деталей при соблюдении следующих условий, налагаемых на соотношения между количествами деталей:

Таблица 4

Типы станков	Число станков (шт.)	Производительность станка (дет/час)				
		деталь 1	деталь 2	деталь 3	деталь 4	деталь 5
1.Фрезерный	3	10	20	40	10	35
2.Револьверный	3	20	30	25	18	52
3.Автоматический револьверный	1	30	80	60	32	78
4.Обрабатывающий центр	2	50	70	90	45	94

$нд1 = нд2, нд3 = 3нд4, нд5 = нд4 + нд3 + 0,1нд1$ ($ндL$ – число деталей L).

5. Для производства комплексной продукции требуется изготовить 4 вида изделий.

Их изготовление может быть поставлено на каждом из пяти типов предприятий «Е.Н. Бендерская и др.» [1, с. 28]. Производственная мощность предприятия и количество предприятий каждого типа даны в табл. 5.

Таблица 5

Тип предприятия	Число предприятий	Производственная мощность, шт/нед			
		По изделиям № 1	По изделиям № 2	По изделиям № 3	По изделиям № 4
№1	5	100	15	200	450
№2	3	400	200	150	170
№3	11	20	3	300	390
№4	9	200	50	20	160
№5	2	600	250	180	240

Определить, сколько предприятий каждого типа надо поставить на производство каждого из изделий, чтобы обеспечить максимальный выпуск комплектов, если в каждый комплект должно входить два изделия первого вида, одно второго, три третьего и пять четвертого.

6. На мебельной фабрике требуется раскроить.

5000 прямоугольных листов фанеры размером 4 x 5 м каждый с тем, чтобы получить 3 вида прямоугольных деталей: деталь А должна иметь размер 2 x 2 м, деталь Б - размер 1 x 3 м, деталь В – 1 x 0,5 м (Е.Н. Бендерская и др.) [1,с.28]. Необходимо, чтобы деталей А оказалось не меньше, чем деталей Б, а деталей В должно равняться сумме деталей А и Б. Каким образом следует произвести раскрой, чтобы получить максимальное число деталей?

7. Фирма – булочно-кондитерский комбинат (БКК).

Выпускает следующие виды продукции: табл. 7.1 [13].

Таблица 7.1

Номер продукции j	1	2	3	4	5
Наименование продукции	булки	пирожные	ватрушки	коржики	слойки
Отпускная цена ед. продукции C_j , руб	0,84	3,2	1,6	1,5	2,1

Для выпуска этих видов продукции необходимо сырье, перечисленное в табл. 7.2; здесь же указано количество каждого вида сырья, имеющегося на складе БКК.

Таблица 7.2

Номер сырья i	1	2	3	4	5
Наименование сырья	мука	сахар	масло	творог	яйца
Количество сырья	200 кг	50кг	50 кг	50 кг	500 шт.

В табл. 7.3 приведена рецептура, т.е. необходимое количество каждого вида сырья для приготовления каждого вида продукции.

Таблица 7.3

Продукция j Сырье i	1 Булка	2 Пирожное	3 Ватрушка	4 Коржик	5 Слойка
1 Мука, кг	0,1	0,04	0,08	0,06	0,05
2 Сахар, кг	0,01	0,05	0,02	0,04	0,03
3 Масло, кг	0	0,05	0,01	0,02	0,02
4 Творог, кг	0	0	0,05	0,02	0,03
5 Яйца, шт.	0,1	0,2	0,2	0,2	0,3

Фирме необходимо определить оптимальный план выпуска продукции (что и в каком количестве произвести), чтобы при имеющемся в БКК сырье получить максимальный доход.

8. Для строительства домов на 100 строительных площадках.

Выбраны 5 типовых проектов (Е.Н. Бендерская и др.) [1,с.29]. По каждому из проектов известны длительность закладки фундаментов, возведения стен и крыш, отделка, а также жилая площадь дома. Возможно, исходя из трудовых, сырьевых и технических ресурсов одновременное ведение закладки 10 фундаментов и строительства и отделки 15 зданий (см. табл. 8).

Таблица 8

Вид работ	Длительность ведения работ (сут)				
	I	II	III	IV	V
Закладка фундамента	20	30	35	30	40
Строительство	30	15	40	25	15
Отделка	10	5	20	10	10
Жилая площадь м ²	2000	2000	5000	4000	6000

Составить план строительства, максимизирующий ввод жилой площади в течении года (300 рабочих суток) при условии, что домов II типа должно быть построено не меньше 10

9. Участник экспедиции укладывает рюкзак

Ему требуется решить, какие положить продукты (С. А. Ашманов) [2, с. 119]. В его распоряжении имеются мясо, мука, сгущенное молоко и сахар. В рюкзаке для продуктов осталось лишь 45 дм³ объема, и нужно, чтобы суммарная масса продуктов не превосходила 35 кг. Врач экспедиции рекомендовал, чтобы мяса (по массе) было больше муки по крайней мере в два раза, муки не меньше молока сгущенного, а молока по крайней мере в восемь раз больше, чем сахара. Характеристики продуктов приведены в таблице 1.3. При этом, запас каждого из продуктов не может быть меньше количества, указанного в 3-й строке табл. 9.

Таблица 9

Характеристики	Продукты			
	Мясо-нежир. говяд	Мука выпш. сорта	Молоко сгущ. с сах.	Сахар
Объем (дм ³ / кг)	1	1.57	0,6	1
Калорийность 1кг, Ккал	1710	3540	3450	4000
Минимальный запас, кг	0,8	1,5	0,8	0

Сколько и каких продуктов нужно положить в рюкзак, с тем, чтобы суммарная калорийность продуктов была наибольшей?

10. Прядильная фабрика для производства двух видов пряжи.

Использует три типа сырья — чистую шерсть, капрон и акрил (С. А. Ашманов) [2, с.105]. В табл. 10.1

указаны нормы расхода сырья, его общее количество, которое может быть попользовано фабрикой в течение года, и прибыль от реализации тонны пряжи каждого вида.

Таблица 10.1			
Тип сырья	Нормы расхода сырья на		Количество сырья (т)
	Вид 1	Вид 2	
Шерсть	0,5	0,2	600
Капрон	a	0,6	b
Акрил	$0,5-a$	0,2	c
Прибыль от реализации 1 т пряжи (руб)	1100	900	

Требуется составить годовой план производства пряжи для одного из производственных вариантов табл. 10.2 с целью максимизации суммарной прибыли:

Таблица 10.2

Вар	a	b	c	Вар	a	b	c	Вар	a	b	c	Вар	a	b	c
1	0,1	620	500	6	0,1	870	510	11	0,2	710	400	16	0,3	690	300
2	0,1	730	500	7	0,1	790	520	12	0,2	880	410	17	0,3	720	300
3	0,1	840	500	8	0,2	920	400	13	0,2	810	410	18	0,3	750	300
4	0,1	650	510	9	0,2	850	400	14	0,2	740	410	19	0,3	780	300
5	0,1	760	510	10	0,2	780	400	15	0,3	660	300	20	0,3	800	300

11. Нефтеперерабатывающий завод может использовать.

Две различные технологии перегонки нефти для производства бензина, керосина и солярового масла (С. А. Ашманов) [2, с. 106].

Таблица 11.1

Наименование продукции	Выход продукции (т)		Стоимость 1 т готового продукта (р)	Суточный объем госзаказа (т)
	Технология 1	Технология 2		
Бензин	0,6	0,3	100	117
Керосин	0,1	0,3	50	54
Соляровое масло	—	0,3	20	
Отходы	0,3	0,1		
Издержки производства (р.)	a	b		
Загрузка оборудования (маш.-ч)	0,2	0,05		

В табл. 11.1 и 11.2 приведены данные, показывающие выход продукции, отходы, издержки производства (стоимость нефти, заработная плата, амортизация и т. п.) и загрузку оборудования в расчете на 1 т переработанной нефти. Кроме того, указаны стоимость 1 т готовой продукции и суточный объем государственного заказа, который необходимо удовлетворить.

Ресурс оборудования составляет 75 маш.-ч в сутки. Все отходы должны пройти через очистные сооружения, производительность которых составляет с т/сут. Поставки нефти и опрос на всю продукцию завода неограниченны. Требуется составить суточный план производства для одного из вариантов табл. 20.2 с целью максимизации прибыли.

Таблица 11.2

Вар	a	b	c	Вар	a	b	c	Вар	a	b	c	Вар	a	b	c
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Вар	a	b	c	Вар	a	b	c	Вар	a	b	c	Вар	a	b	c
1	13	37	130	6	21	39	135	11	37	43	140	16	39	45	145
2	15	37	135	7	23	39	140	12	39	45	145	17	31	45	130
3	17	37	140	8	25	39	145	13	37	45	130	18	37	45	135
4	19	37	145	9	29	41	130	14	35	45	135	19	35	45	140
5	21	37	130	10	31	41	135	15	33	45	140	20	33	45	145

12. Рацион кормления коров на молочной ферме.

Может состоять из трех продуктов — вена, силоса и концентратов (С. Л. Ашманов) [2, с. 116]. Эти продукты содержат питательные вещества — белок, кальций и витамины. Численные данные представлены в табл. 12. В расчете на одну корову уточные нормы потребления белка и кальция составляют не менее 2000 и 210 г соответственно. Потребление витаминов строго дозировано и должно быть равно 87 мг в сутки.

Продукты	Таблица 12 Питательные вещества		
	Белок (г/кг)	Кальций (г/кг)	Витамины (мг/кг)
Сено	50	10	2
Силос	70	6	3
Концентраты	180	3	1

Составить самый дешевый рацион, если стоимость 1 кг сена, силоса и концентрата равна соответственно 15, 2 и 6 к.

13. В области имеются 3 цементных завода.

И 5 потребителей их продукции – домостроительные комбинаты (С.А.Ашманов) [2,с. 117]. В табл. 13 указаны суточные объемы производства цемента, суточные потребности в нем комбинатов и стоимость перевозки 1 т цемента от каждого завода к каждому комбинату.

Таблица 13

Заводы	Производство цемента (т/сут)	Стоимость перевозки 1 т цемента (р.)				
		Комбинат 1	Комбинат 2	Комбинат 3	Комбинат 4	Комбинат 5
1	40	10	15	25	27	35
2	60	20	30	30	30	30
3	80	15	45	22	34	18
	Потребности в цементе (т/сут)	50	20	30	45	35

Требуется составить план суточных перевозок цемента с целью минимизации транспортных расходов.

14. Колхоз «Куликово поле» может засеять свои поля пшеницей пяти сортов.

Урожайность которых зависит от того, будет ли лето дождливым, умеренным, сухим, жарким или прохладным. Соответствующие данные приведены в табл. 36 (С.А.Ашманов) [2,с. 120].

1) Какие сорта пшеницы и в какой пропорции следует сеять, чтобы гарантированный (не зависящий от погоды) максимизированный урожай на площади 150 га был не менее 3500 ц? По результату вычисления уточните величину гарантированного урожая.

Гарантированный урожай – это наименьший урожай из группы погодных урожаев табл. 14.

Таблица 36

	Урожайность (ц /га)
--	---------------------

	Сорт 1	Сорт 2	Сорт 3	Сорт 4	Сорт 5
Дождливая	21	20	30	15	32
Умеренная	24	20	23	30	26
Сухая	25	40	15	34	22
Жаркая	28	35	48	36	18
Прохладная	31	22	19	27	24

Подсказка. Для формирования α -вектора f следует использовать строку данных по урожайности для любой погоды с одновременной проверкой соответствия урожая для каждой погоды заданному неравенству.

2) Повторите расчёт для набора урожайностей другой погоды. В выводах по решению сопоставьте результаты обоих расчётов.

15. Имеются 2 склада и 6 цехов обработки деталей.

Для которых известны объёмы запасов деталей на складах $A_1 = 36$ шт и $A_2 = 23$ шт и потребности цехов $B_1=5, B_2=8, B_3=11, B_4=7, B_5=16, B_6=12$, причем, запасы деталей на складах и потребности цехов являются сбалансированными, т.е.

$$A_1 + A_2 = B_1 + B_2 + \dots + B_6$$

и заданы длины транспортных коммуникаций между складами и цехами в следующей таблице:

Склад	Расстояния Склад–Цех, м					
	B_1	B_2	B_3	B_4	B_5	B_6
A_1	180	70	300	450	600	1200
A_2	240	300	460	700	900	2400

Требуется сформулировать математическую модель выбора оптимального плана транспортировки деталей от складов к цехам, минимизирующую суммарную длину перевозок, если за каждый рейс транспортного средства можно доставить только одну деталь.

16. Фирма производит комплект деталей.

Комплект состоящий из детали №1 – 1 шт, детали №2 – 1 шт и детали №3 – 2 шт [12, с.344]. Производительность станков при изготовлении деталей комплекта представлена в табл.16.

Таблица 16

Производительность станков, дет./смена

Тип станка	Число станков	Производительность одного станка			Производительность всех станков		
		Деталь №1	Деталь №2	Деталь №3	Деталь №1	Деталь №2	Деталь №3
Фрезерный	4	10	20	50	40	80	200
Обрабатывающий центр	2	25	35	60	50	70	120
Револьверный	3	20	30	40	60	90	120
Автоматический револьверный	1	30	80	125	30	80	125

Требуется распределить изготовление деталей между станками так, чтобы получить максимальное количество изготавливаемых комплектов за смену.

17. Фирма продаёт консервы.

Фирма распределяет поступающие партии консервов от 4 промышленных предприятий на 7 своих складов, которые снабжают консервами торговые предприятия [13, с.352]. Объем произведенной предприятиями продукции равен объему продукции, поступающей со складов в торговлю (см. табл. 17.1).

Таблица 17.1

Объемы производимой и потребляемой консервной продукции, партии/нед.

Предприятия				Склады						
1	2	3	4	1	2	3	4	5	6	7

Предприятия				Склады						
190	100	150	300	100	150	85	135	110	70	90

Стоимость перевозки продукции от предприятий на склады – согласно табл. 17.2.

Таблица 17.2

Стоимость перевозки продукции Предприятие – Склад, руб.

		Склады						
		1	2	3	4	5	6	7
Предприятия	1	200	280	320	550	700	390	420
	2	480	360	400	440	180	250	370
	3	350	540	220	470	360	170	240
	4	170	380	430	120	570	245	630

Необходимо спланировать перевозки от предприятий на склады так, чтобы склады получили продукцию согласно табл. 17.1, и стоимость всех перевозок была бы минимальной.

18. Определение плана компании по закупкам и продажам бананов.

Компания имеет склад [12.с.375], емкость которого $B=520$ ящиков бананов. На начало года на складе был запас $A = 95$ ящиков. Пусть x_j – количество ящиков бананов, приобретаемых компанией в j – месяц 1-го полугодия по цене c_j руб/ящик (см. табл.18), а y_j – количество ящиков, проданных компанией в j -м месяце по реализационной цене p_j руб/ящик.

Таблица 18

Закупочные и продажные цены ящика бананов, руб.

Месяцы	Янв	Фев	Март	Апр	Май	Июнь
Цены, руб						
Закупочная	150	200	190	160	140	120
Реализационная	250	270	230	200	180	150

Продавать в рассматриваемом j -м месяце можно только те бананы, которые закуплены до 1-го числа данного месяца, т.е. на объем продаваемых ящиков налагается следующее ограничение:

$$\sum_{k=1}^j y_k \leq A + \sum_{k=1}^{j-1} x_k \quad (18.1)$$

Накупаемые бананы также налагается ограничение из-за конечной емкости склада:

$$\sum_{k=1}^j x_k \leq B - A + \sum_{k=1}^j y_k \quad (18.2)$$

Требуется так спланировать закупки x_j и продажи y_j , чтобы максимизировать прибыль компании за рассматриваемый период.

19. В механическом цехе имеется три металлорежущих станка.

M_1 , M_2 и M_3 , на которых могут быть изготовлены детали $D_1, \dots, D_5$ (Алексинская Т.В.) [14, с.6]. Время изготовления деталей каждым из станков в часах, ресурс времени станков и плановые задания указаны в табл. 19.

Таблица 19

Станки \ Детали	Время изготовления детали, час					Ресурс времени станка, час
	D_1	D_2	D_3	D_4	D_5	
M_1	1	3	2.5	1.8	3	1800
M_2	2	4	3	1.4	4	1500
M_3	3	1	2	2.7	5	2200
Требуемое количество деталей	100	280	350	600	490	—

Распределить задания между станками так, чтобы общее время работы всех станков было минимальным, и время работы каждого не превосходило его ресурс.

20. В амбаре 4 мышиных норы.

В первой проживает 15 мышей [15], во второй – 20, в третьей – 10 мышей, а в четвертой – 25 мышей, а также 5 источников пищи, у которых и кормятся все мыши: у окорока – 5 мышей, у мешка крупы – 18 мышей, у мешка муки – 17 мышей, у мешка картошки – 22 мыши и у ведра с пищевыми отходами – 8 мышей. На пути от нор к источникам пищи мышей подстерегает кот: потери мышей представлены в табл.20.

Таблица 20

Количество мышей, схваченных котом, на пути от норы к источнику пищи,
отн. ед. от количества, вышедшего к данному источнику

Пища Норы	окорок	мешок крупы	мешок муки	мешок картошки	ведро с отходами
нора 1	0,25	0,3	0,4	0,15	0,1
нора 2	0,2	0,18	0,05	0,27	0,35
нора 3	0,15	0,25	0,33	0,16	0,03
нора 4	0,1	0,07	0,28	0,12	0,38

Определить оптимальную стратегию прокормления мышей, минимизирующую их потери на пути к источникам пищи.

21. На трёх морских линиях используются суда пяти типов.

Объемы перевозок, производительности судов, их эксплуатационные расходы приведены в табл. 21 (Вепринская Т.) [16].

Таблица 21

Типы судов	Производительность судов, млн. тонно-миль в сутки			Эксплуатационные расходы, тыс. руб. в сутки			Эксплуатационный период, сут.
	Линия 1	Линия 2	Линия 3	Линия 1	Линия 2	Линия 3	
1	12	10	5	6	8	4	280
2	15	12	8	7	7	4	280
3	12	14	12	8	7	7	280
4	10	12	14	6	6	6	280
5	16	20	18	9	8	9	280
Объём перевозок, млн. т-миль	4360	7840	6200	—	—	—	—

Определить план расстановки судов по линиям, обеспечивающий минимум эксплуатационных расходов при условии выполнения заданного объёма перевозок на линии.

22. Порт обеспечивается вагонами 4-х типов для погрузки 3-х видов различных грузов.

В табл.22 дано количество, грузоподъёмность вагонов, количество каждого вида груза для отправки и нормы времени погрузки (Вепринская Т.) [16].

Таблица 22

№ варианта	Тип вагонов	Грузоподъёмность по видам груза, т			Нормы времени при погрузке по видам груза, часы.			Число вагонов
		1 вид	2 вид	3 вид	1 вид	2 вид	3 вид	

3.1	1	22	30	-	3	4	-	40
	2	45	54	62	5	6	7	60
	3	32	-	60	3	-	9	30
	4	35	62	67	3	8	7	30
	кол-во груза, т	1750	2220	3200	—	—	—	—

Составить план погрузки всего груза в вагоны, минимизирующий суммарное время погрузки.

23. Хозяйство имеет 850 гектаров земли.

А также 15 тысяч тонн органических удобрений [17] и трудовой ресурс 50 тысяч человеко-дней. Имеются семена картофеля, капусты и многолетних трав. Затраты труда на возделывание одного гектара капусты, картофеля и многолетних трав равны, соответственно, 50, 30 и 15 человеко-дней. Расходы органических удобрений на те же культуры равны 20, 15 и 10 тоннам на гектар.

Необходимо найти оптимальное распределение имеющейся земли под эти культуры с учетом того, что каждый гектар земли под капустой принесет 1000 рублей дохода, картофеля 800, а многолетних трав 450.

24. На морском бассейне имеются 3 судоремонтных завода.

На которые поступают 5 типов судов (Вепринская Т.) [16]. В табл. 24 дано количество судов по каждому типу, время, необходимое для ремонта каждого типа судна каждым заводом, а также количество суток каждому заводу для ремонта судов в течение планируемого периода.

Таблица 24

Тип судна	Время ремонта судна, сут.			Расходы на ремонт судов, тыс. руб.			Число судов бассейна, шт.
	Завод 1	Завод 2	Завод 3	Завод 1	Завод 2	Завод 3	
1	30	18	35	300	450	250	14
2	36	50	44	400	350	350	20
3	90	120	110	600	800	700	15
4	100	90	120	600	600	700	10
Кол. сут. для ремонта, сут.	2450	1340	930	—	—	—	—

Составить распределение ремонтируемых судов по заводам при условии, что все суда должны быть отремонтированы в планируемый период, а суммарные расходы на ремонт – минимизированы.

25. Хозяйство «Курская дуга» специализируется в полеводстве.

На производстве зерна, сахарной свеклы и подсолнечника [18]. В с.-х. предприятии имеются 3200 га пашни, трудовые ресурсы в объеме 7000 чел.-дней и минеральные удобрения в объеме 15000 ц. Техноэкономические показатели производства рассматриваемых культур представлены в табл. 25. Следует также учесть, что:

- площадь посева технических культур (сахарной свеклы и подсолнечника) не должна превышать 25% общей площади пашни;
- хозяйством заключен договор на продажу зерна в объеме 65000 ц.

Таблица 25

Показатели	Сельскохозяйственные культуры		
	зерновые	сахарная свекла	подсолнечник
Урожайность, ц/га	26	275	18
Цена реализации 1 ц продукции, руб./ц.	215	75	374
Стоимость товарной продукции с 1 га, тыс. руб.	5,59	20,62	6,73
Затраты на 1 га: труда, чел.-дней. минеральных удобрений, ц	1,5 2	4,5 15	1,5 2,3

Прибыль с 1 га, руб.	2,89	7,93	3,63
----------------------	------	------	------

Требуется найти такое сочетание посевных площадей, которое обеспечило бы получение максимума прибыли.

26. Для перегрузки 5 типов грузов выделено 4 козловых крана.

Нормы выработки для каждого крана, количества предъявленных грузов и матрица прибылей по каждому крану представлены в табл. 26 (Вепринская Т.) [16]. Составить план организации работы кранов по перегрузке всех грузов, обеспечивающий получение максимальной прибыли при условии работы каждого крана в объеме, не превышающем недельной выработки.

Таблица 26

№ крана	Недельная прибыль от работы кранов на каждом типе груза, руб.					Недельная норма выработки, тыс. т
	1 тип	2 тип	3 тип	4 тип	5 тип	
1	580	430	480	340	270	150
2	200	570	370	490	400	350
3	600	530	440	380	470	100
4	590	710	420	450	300	150
Кол-во груза, тыс.т	95	85	120	145	125	—

27. На фабрике эксплуатируются 4 типа ткацких станков С1... С3.

На них можно выпускать четыре вида тканей Т1...Т4 [19]. Известны следующие данные о производственном процессе – табл. 27.

Таблица 27

Показ. производства Станки	Производительность станков при выпуске тканей, м/час				Себестоимость тканей, руб/м				Фонд рабочего времени станка, час
	Т1	Т2	Т3	Т4	Т1	Т2	Т3	Т4	
С1	24	30	18	42	2	1	3	1	90
С2	12	15	9	21	3	2	4	1	220
С3	8	10	6	14	6	3	5	2	180
Объем выпуска тканей, м	1200	900	1800	840	—	—	—	—	—

Требуется распределить выпуск тканей по станкам, минимизирующий общую себестоимость производства.

28. Для перевозок пассажиров на 3 линиях используются суда 3 типов.

В табл. 28 даны провозные способности каждого типа судов линии, число судов каждого типа и среднесуточное число пассажиров по всем линиям, а также соответствующие эксплуатационные расходы (Вепринская Т.) [16].

Таблица 28

Типы судов	Провозная способность судна, тыс. пассажиров сутки			Эксплуатационные расходы, тыс. руб. в сутки			Число судов
	1 линия	2 линия	3 линия	1 линия	2 линия	3 линия	
1	1,2	1,6	1,6	1,1	1,3	1,2	30
2	3,9	4,0	4,7	2,7	2,8	3,1	25
3	1,5	1,7	1,1	1,1	1,3	0,9	22
4	3,6	4,3	2,6	2,2	2,2	1,8	16

Суточная интенсивность перевозок, тыс. пассажиров в сутки	90	140	80	—	—	—	—
---	----	-----	----	---	---	---	---

Найти расстановку судов по линиям, при которой достигается минимум эксплуатационных расходов и обеспечивается заданная суточная интенсивность перевозок на линиях.

29. Три грузопотока обрабатываются 5 типами перегрузочного оборудования.

Производительности каждого из этих видов перегрузочного оборудования (ПО) на каждом из грузопотоков указаны в табл. 29 (Вепринская Т.) [16]. Там же приведены количества выделенных машино-смен по каждому виду ПО и мощности грузопотоков.

Таблица 29

Вид перегрузочного оборудования	Производительность перегрузочного оборудования при освоении грузопотоков, т/смену			Выделенный ресурс перегрузочного оборудования, машино-смены
	1-й грузопоток	2-й грузопоток	3-й грузопоток	
1	140	190	250	270
2	200	-	200	360
3	150	180	220	360
4	180	160	200	480
5	120	210	-	410
Объем грузопотока, т	108000	66800	117500	—

Составить план загрузки ПО, обеспечивающий максимальную суммарную производительность оборудования и освоение объемов всех грузопотоков, при условии неперевышения ресурса оборудования.

30. Семейная фирма производит три широко популярных напитка.

Напитки: «Pink Fizz», «Mint Pop» и «Gloria» (Аксенова Р.Н.) [6, с.5]. Фирма может продать всю продукцию, которая будет произведена, однако объем производства ограничен количеством специального ингредиента и производственной мощностью имеющегося оборудования. Для производства 1 л «Pink Fizz», «Mint Pop» и «Gloria» оборудованию требуется соответственно: 0,02 ч, 0,04 ч и 0,05 ч. Расход специального ингредиента составляет 0,01, 0,04 и 0,06 кг на 1 л «Pink Fizz», «Mint Pop» и «Gloria» соответственно. Оборудование работает 23 ч в сутки; 1 ч – профилактика. На фирму в сутки поставляется 22 кг специального ингредиента.

Доход фирмы составляет 0,10 ф. ст. за 1 л «Pink Fizz», 0,30 ф. ст. за 1 л «Mint Pop» и 0,45 ф. ст. за 1 л «Gloria». Каков объем ежесуточного производства каждого напитка для максимизации суточного дохода? Величина суточного дохода? При какой величине суточных поставок спецингредиента доход фирмы будет наибольшим? Величина наибольшего дохода?

31. Завод-производитель высокоточных элементов для автомобилей.

Выпускает два различных типа деталей: X и Y (Аксенова Р.Н.) [6, с. 5]. Завод располагает фондом рабочего времени в 4000 чел.-ч. в неделю. Для производства одной детали типа X требуется 1 чел.-ч, а для производства одной детали типа Y — 2 чел.-ч. Производственные мощности завода позволяют выпускать максимум 2250 деталей типа X и 1750 деталей типа Y в неделю. Каждая деталь типа X требует 2 кг металлических стержней и 5 кг листового металла, а для производства одной детали типа Y необходимо 5 кг металлических стержней и 2 кг листового металла. Уровень запасов каждого вида металла составляет 10000 кг в неделю. Кроме того, еженедельно завод поставляет 600 деталей типа X своему постоянному заказчику. Существует также профсоюзное соглашение, в соответствии с которым общее число производимых в течение одной недели деталей должно составлять не менее 1500 штук.

Сколько деталей каждого типа следует производить, чтобы максимизировать общий доход за неделю,

если доход от производства одной детали типа X составляет 30 ф. ст., а от производства одной детали типа Y – 40 ф. ст.? Величина максимального дохода?

32. Завод по производству компьютеров и системы подготовки текстов.

В настоящее время на нём освоены четыре модели (Аксенова Р.Н.) [6, с. 7]:

- а) «Юпитер» – ОЗУ 256 Мбайт, HDD 40 Гб, CD-RW, видеокарта 128 Мб;
- б) «Венера» – ОЗУ 256 Мбайт, HDD 80 Гб, CD RW, видеокарта 256 Мб;
- в) «Марс» – ОЗУ 512 Мбайт, HDD 120 Гб, DVD RW, видеокарта 256 Мб;
- г) «Сатурн» – ОЗУ 1024 Мбайт, HDD 250 Гб, DVD RW, видеокарта 256 Мб.

В производственный процесс вовлечены три цеха завода — цех узловой сборки, сборочный и испытательный. Распределение времени, требуемого для обработки каждой модели в каждом цехе, а также максимальные производственные мощности цехов приведены в табл. 57.1. Отдел исследований рынка производит периодическую оценку потребительского спроса на каждую модель. Максимальные прогнозные значения спроса и доходы от реализации единицы продукции каждой модели также содержатся в табл. 32.

Таблица 32.
Время, требуемое на обработку каждой модели в каждом цехе

Цех	Время на единицу продукции				Максимальная производительность, ч/мес.
	“Юпитер”	“Венера”	“Марс”	“Сатурн”	
Узловой сборки	5	8	20	25	800
Сборочный	2	3	8	14	420
Испытательный	0,1	0,2	2	4	150
Максимальное прогнозное значение спроса за месяц	100	45	25	20	—
Доход, ф.ст.	120	180	220	280	—

Составить месячный план цеха с целью максимизации дохода. Величина максимального дохода?

33. Менеджер по ценным бумагам намерен разместить 100 000 ф. ст. капитала.

Разместить таким образом, чтобы получать максимальные годовые проценты с дохода (Аксенова Р.Н.) [6, с. 8]. Его выбор ограничен четырьмя возможными объектами инвестиций: А, В, С и D. Объект А позволяет получать 6 % годовых, объект В – 8 % годовых, объект С – 10 %, а объект D – 9 % годовых. Для всех четырех объектов степень риска и условия размещения капитала различны. Чтобы не подвергать риску имеющийся капитал, менеджер принял решение, что не менее половины инвестиций необходимо вложить в объекты А и В. Чтобы обеспечить ликвидность, не менее 25% общей суммы капитала нужно поместить в объект D. Учитывая возможные изменения в политике правительства, предусматривается, что в объект С следует вкладывать не более 20% инвестиций, тогда как особенности налоговой политики требуют, чтобы в объект А было вложено не менее 30% капитала.

Вычислить конкретное размещение капитала, обеспечивающее максимизацию общей суммы годовых процентов с дохода, и величину дохода.

34. Оливер А. Петерс скоро выйдет на пенсию.

Ему предстоит решить, как поступить с единовременным пособием, которое в соответствии с пенсионной программой будет предоставлено ему фирмой (Аксенова Р.Н.) [6, с. 42]. М-р Петерс и его супруга намерены предпринять длительный визит в Австралию к своей дочери сроком на два года, поэтому любые сделанные в настоящий момент инвестиции будут свободны для использования на данный период. Очевидно, цель м-ра Петерса состоит в максимизации общего дохода от вложений, полученного за двухлетний период.

Мистера Петерса проконсультировали, что наилучшим вариантом вложения инвестиций был бы инвестиционный фонд, и в настоящее время он рассматривает возможность помещения инвестиций в один из таких фондов, состоящий из инвестиций трех типов – А, В и С. Сумма единовременного пособия составит 25000 ф. ст., однако, мистер Петерс считает, что нет необходимости вкладывать в данный инвестиционный фонд все деньги; часть из них он намерен перевести на свой счет жилищно-строительного кооператива, который гарантирует ему 9 % годовых.

По мнению бухгалтера фирмы, мистеру Петерсу следует попытаться распределить свои инвестиции та-

ким образом, чтобы обеспечить как получение дохода, так и рост капитала. Поэтому ему посоветовали не менее 40% от общей суммы вложить в вариант А и перевести на свой счет. Для обеспечения значительного роста капитала не менее 25 % общей суммы денежных средств, вложенных в инвестиционный фонд, необходимо поместить в проект В, однако, вложения в В не должны превышать 35 % общего объема вложений в инвестиционный фонд ввиду высокой вероятности риска, соответствующей проекту В. Кроме того, для сохранности капитала в проекты А и С следует вложить не менее 50 % средств, помещаемых в инвестиционный фонд.

В настоящее время проект А позволяет получать 10 % годовых и обеспечивает 1 % роста капитала; проект В предполагает рост капитала в 15 %; проект С дает 4 % годовых и 5 %-ный рост капитала.

Как следует распределить сумму единовременного пособия между различными проектами инвестиций, чтобы максимизировать сумму годовых дохода и роста капитала. Вычислить максимальное значение суммы, а также её годовые составляющие: доход и рост капитала.

35. Фабрика «GRM plc» выпускает три вида каш для завтрака.

Это: «Crunchy», «Chewy» и «Riceblue» (Аксенова Р.Н.) [6, с. 42]. Используемые для производства обоих продуктов ингредиенты в основном одинаковы и, как правило, не являются дефицитными. Основным ограничением, накладываемым на объем выпуска, является наличие фонда рабочего времени в каждом из трех цехов фабрики.

Управляющему производством Джою Дисону необходимо разработать план производства на месяц. В табл. 35 указаны общий фонд рабочего времени и число человеко-часов, требуемое для производства 1 т продукта.

Таблица 35

Цех	Необходимый фонд рабочего времени, чел.-ч/т			Общий фонд рабочего времени, чел.-ч. в месяц
	«Crunchy»	«Chewy»	«Rice-blue»	
А. Производство	10	4	6	1400
В. Добавка приправ	3	2	3	390
С. Упаковка	2	5	4	650

Доход от производства 1 т «Crunchy» составляет 150 ф. ст., от производства «Chewy» – 75 ф. ст., от производства «Riceblue» – 110 ф. ст. Нет никаких ограничений на возможные объемы продаж: продать можно всю произведенную продукцию. Составить месячный план производства, максимизирующий доход фабрики. Вычислить величину максимального дохода.

36. Порт обеспечивается вагонами 4 типов для погрузки 3 видов грузов.

В табл. 36 дано количество, грузоподъемность вагонов, количество каждого вида груза для отправки и нормы времени погрузки (Вепринская Т.) [16].

Таблица 36

№ варианта	Тип вагонов	Грузоподъемность по видам груза, т			Нормы времени при погрузке по видам груза, часы.			Число вагонов
		1 вид	2 вид	3 вид	1 вид	2 вид	3 вид	
3.1	1	22	30	-	3	4	-	40
	2	45	54	62	5	6	7	60
	3	32	-	60	3	-	9	30
	4	35	62	67	3	8	7	30
	кол-во груза, т	1750	2220	3200	—	—	—	—

Составить план погрузки всего груза в вагоны, минимизирующий суммарное время погрузки

37. Для перевозок пассажиров на 3 линиях используются суда 4 типов.

В табл. 37 (Вепринская Т.) [16] даны провозные способности каждого типа судов линии, число судов каждого типа и среднесуточное число пассажиров по всем линиям, а также соответствующие эксплуатационные

расходы.

Таблица 37

Типы судов	Провозная способность судна, тыс. пассажиров сутки			Эксплуатационные расходы, тыс. руб. в сутки			Число судов
	1 линия	2 линия	3 линия	1 линия	2 линия	3 линия	
1	1,2	1,6	1,6	1,1	1,3	1,2	30
2	3,9	4,0	4,7	2,7	2,8	3,1	25
3	1,5	1,7	1,1	1,1	1,3	0,9	22
4	3,6	4,3	2,6	2,2	2,2	1,8	16
Суточная интенсивность перевозок, тыс. пассажиров в сутки	90	140	80	—	—	—	—

Найти расстановку судов по линиям, при которой достигается минимум эксплуатационных расходов и обеспечивается заданная суточная интенсивность перевозок на линиях.

38. Нефтяная компания "РТ" улучшает выпускаемое топливо.

Для улучшения эксплуатационных качеств и снижения точки замораживания дизельного топлива, которое она производит, добавляет в него определенные химикаты (Аксенова Р.Н.) [6, с. 43]. В каждом бензобаке объемом 1000 л должно содержаться не менее 40 мг химической добавки X, не менее 14 мг химической добавки Y и не менее 18 мг химической добавки Z. Необходимые химические добавки в форме готовых смесей поставляют "РТ" две химические компании А и В. В табл. 38 приведено содержание химических добавок в каждом продукте, поставляемом указанными компаниями.

Таблица 38

Продукт	Химические добавки в продукте, мг/л		
	X	Y	Z
A	4	2	3
B	5	1	1

Стоимость продукта А – 1,50 ф. ст. за 1 л, а продукта В – 3,00 ф. ст. за 1 л.

Требуется: найти ассортиментный набор продуктов А и В, минимизирующий общую стоимость добавленных в топливо химикатов, и величину минимальной стоимости.

39. Завод производит порошок для изготовления напитков трех видов.

Один из них продается в качестве напитка здоровья, поскольку имеет низкое содержание сахара; другой напиток поставляется в медицинские учреждения в качестве продукции для больных, поскольку он содержит витаминные добавки; наконец, третий является стандартным товаром (Аксенова Р.Н.) [6, с. 43].

В табл. 39 для каждого напитка указаны основные ингредиенты, их стоимость и размер недельного запаса, а также оценки максимального спроса на соответствующие товары за неделю.

Таблица 39

	Расход ингредиентов на 1 кг. продукта, кг			Оценка максимального спроса за неделю, кг	Цена продажи 1 кг. напитка, ф.ст.
	Сахар	Солодовый экстракт	Сухие сливки		
Стандартный напиток	0,30	0,30	0,35	2000	1
Напиток здоровья	0,15	0,25	0,55	1800	1,2
Напиток для больных	0,15	0,30	0,25	1200	1,5
Стоимость 1 кг ингредиента, пенсы	20	60	50	-	-
Размер недельного запаса ингредиентов, кг	1000	1250	2200	-	-

Запас витаминных добавок неограничен. Издержки производства остальных переменных имеют сле-

дующие значения: 10 пенсов за 1 кг стандартного напитка, 9 пенсов за 1 кг напитка здоровья и 12 пенсов за 1 кг напитка для больных.

– Определить недельный план производства напитков, максимизирующий прибыль завода, и величину максимальной прибыли.

– Как изменится план производства, если максимальный спрос на напиток здоровья увеличится до 2500 кг/нед? Вычислите величину измененной максимальной прибыли.

40. "Princetown Paints Ltd" выпускает три основных типа румян.

Жидкие, перламутровые и матовые (Аксенова Р.Н.) [6, с.45] – с использованием одинаковых смесеобразующих машин и видов работ. Главному бухгалтеру фирмы было поручено разработать для компании план производства на неделю. Информация о ценах продаж и стоимости 100 л товара приведена в табл. 40 (ф. ст.).

Таблица 40

–	Румяна, ф. ст.		
	Жидкие	Перламутровые	Матовые
Цена продажи на 100 л	120	126	110
Издержки производства товаров на 100 л:			
Стоимость сырья	11	25	20
Стоимость трудозатрат	30	36	24
Стоимость приготовления смеси	32	20	36
Другие издержки	12	15	10

Стоимость 1 чел.-ч составляет 3-ф.ст. а стоимость 1 ч приготовления смеси – 4 ф. ст. Фонд рабочего времени ограничен 8000 чел.-ч. в неделю, а ограничение на фонд работы смесеобразующих машин равно 5900 ч в неделю.

В соответствии с контрактными соглашениями компания должна производить 25000 л матовых румян в неделю. Максимальный спрос на жидкие румяна равен 35000 л в неделю, а на перламутровые румяна – 29000 л в неделю.

– Определить объемы производства жидких и перламутровых румян в неделю, при которых достигается максимальное значение получаемой за неделю прибыли. Вычислить величину максимальной прибыли.

– Рабочие предложили выполнять сверхурочную работу за дополнительное вознаграждение в 1 ф. ст. за каждый сверхурочно отработанный час. Будет ли целесообразным введение сверхурочной работы на таких условиях? Если это так, то каковы рекомендации по поводу количества часов сверхурочной работы, которое следует ввести, и какова будет дополнительная прибыль от применения сверхурочной работы?

41. Администрация компании "Nemesis Company" приняла решение.

О слиянии двух своих заводов в Аббатсфилде и Берчвуде (Аксенова Р.Н.) [6, с. 45]. Предусматривается закрытие завода в Аббатсфилде и за счет этого — расширение производственных мощностей предприятия в Берчвуде. На настоящий момент распределение рабочих высокой и низкой квалификации, занятых на обоих заводах, является следующим: табл. 41.

Таблица 41

Квалификация рабочих	Аббатсфилд, чел	Берчвуд, чел
Высокая	200	100
Низкая	300	200
Итого	500	300

В то же время после слияния завод в Берчвуде должен насчитывать 240 рабочих высокой и 320 рабочих низкой квалификации.

После проведения всесторонних переговоров с привлечением руководителей профсоюзов были выработаны следующие финансовые соглашения.

1. Все рабочие, которые попали под сокращение штатов, получают выходные пособия следующих разме-

ров:

Квалифицированные рабочие	- 2000 ф. ст.;
Неквалифицированные рабочие	- 1500 ф. ст.

2. Рабочие завода в Аббатсфилде, которые должны будут переехать, получают пособие по переезду в размере 2000 ф. ст.

3. Во избежание каких-либо преимуществ для рабочих Берчвудского завода доля бывших рабочих завода в Аббатсфилде на новом предприятии должна совпадать с долей бывших рабочих Берчвудского завода.

Как осуществить выбор работников нового предприятия из числа рабочих двух бывших заводов таким образом, чтобы минимизировать общие издержки, связанные с увольнением и переменой места жительства части рабочих?

42. Менеджер международной банковской организации по инвестициям.

Располагает 550 000 ф. ст., находящимися на счете банка, которые необходимо инвестировать, и рассматривает четыре общих типа инвестиций (Аксенова Р.Н.) [6, с.47].

Тип 1: государственные ценные бумаги;

Тип 2: ценные бумаги корпораций;

Тип 3:- обыкновенные акции отраслей сферы обслуживания;

Тип 4: обыкновенные акции отраслей производственной сферы.

Целью менеджера по инвестициям является максимизация нормы отдачи вложений, причем размер годовых процентов от инвестиций равен 8, 9, 10 и 12% для типов 1, 2, 3 и 4 соответственно. Денежные средства, не инвестированные ни по одному из указанных выше типов, остаются на банковском счете и приносят 4% годовых.

Менеджер по инвестициям принял решение, что не менее 50 000 ф. ст. следует поместить в ценные бумаги корпораций, а в инвестиционные проекты с элементами риска (т.е. ценные бумаги корпораций и все виды обыкновенных акций) следует вложить не более 300 000 ф.ст. Кроме того, он считает, что, по крайней мере, половину общей суммы денежных средств, инвестированных в соответствии с указанными выше типами инвестиций, следует вложить в обыкновенные акции, но в акции отраслей производственной сферы следует поместить не более одной четверти общей суммы инвестиций.

Определить оптимальный план инвестиций, сумму денежных средств, оставленных на банковском счете, и ежегодный доход от реализации данного плана, выраженный в процентах.

43. Компания "Bermuda Paint" – частная промышленная фирма.

Специализируется на производстве технических лаков (Аксенова Р.Н.) [6, с. 48]. Табл. 43 содержит информацию о ценах продажи и соответствующих издержках производства единицы полировочного и матового лаков.

Таблица 43

Лак	Цена продажи 1 галлона, ф. ст.	Издержки производства 1 галлона, ф. ст
Матовый, стандартный	12,0	8,0
Матовый, супер	14,0	9,0
Полировочный, стандартный	17,0	10,5
Полировочный, супер	19,0	12

Для производства 1 галлона матового лака каждого вида необходимо затратить 6 мин трудозатрат, а для производства одного галлона полировочного лака каждого вида – 12 мин. Резерв фонда рабочего времени составляет 400 чел.-ч. в день. Размер ежедневного запаса необходимой для производства лаков химической смеси равен 100 унциям, тогда как ее расход на один галлон матового и полировочного лаков составляет соответственно 0,04; 0,05; 0,01 и 0,02 унции. Технологические возможности завода позволяют выпускать не более 3000 галлонов лака в день.

В соответствии с соглашением с основным оптовым покупателем компания должна поставлять ему 5000 галлонов матового лака и 2500 Галлонов полировочного лака за каждую рабочую неделю (состоящую из 5 дней). Кроме того, существует профсоюзное соглашение, в котором оговаривается минимальный объем произ-

водства в день, равный 2000 галлонов.

– Администрации компании необходимо определить ежедневные объемы производства каждого вида лаков, которые позволяют получать максимальную прибыль. Вычислить величину максимальной прибыли.

– Профсоюз компании требует увеличения оплаты 1 ч. сверхурочных работ на 20 ф. ст. Обосновать, считает ли администрация компании целесообразным такое предложение? Если указанный размер оплаты сверхурочных работ является выгодным, какое количество часов сверхурочных работ в день целесообразно использовать?

44. Компания имеет четыре сбытовые базы и четыре заказа.

Заказы необходимо доставить различным потребителям (Аксенова Р.Н.) [6, с. 71]. Складские помещения каждой базы вполне достаточны для того, чтобы вместить один из этих заказов. В табл. 44 содержится информация о расстоянии между каждой базой и каждым потребителем.

Таблица 44
Расстояние от сбытовых баз до потребителей

Сбытовая база	Расстояние до потребителей, км			
	Потребители			
	I	II	III	IV
A	68	72	75	83
B	56	60	58	63
C	38	40	35	45
D	47	42	40	45

Как следует распределить заказы по сбытовым базам, чтобы суммарная дальность транспортировки была минимальной? Величина минимальной суммарной дальности транспортировки?

45. В распоряжении некоторой компании имеется 6 торговых точек и 6 продавцов.

Из прошлого опыта известно, что эффективность работы продавцов в различных торговых точках неодинакова (Аксенова Р.Н.) [6, с. 71]. Коммерческий директор компании произвел оценку деятельности каждого продавца в каждой торговой точке. Результаты этой оценки представлены в табл. 45.

Таблица 45
Объемы продаж в различных торговых точках для различных продавцов

Продавец	Объемы продаж, ф. ст./тыс. шт.					
	Торговые точки					
	I	II	III	IV	V	VI
A	68	72	75	83	75	69
B	56	60	58	63	61	59
C	35	38	40	45	25	27
D	40	42	47	45	53	36
E	62	70	68	67	69	70
F	65	63	69	70	72	68

Как коммерческий директор должен осуществить назначение продавцов по торговым точкам, чтобы достичь максимального объема продаж? Величина максимального объема продаж?

46. Три завода поставляют разновидность стали на пять торговых складов.

Спрос каждого торгового склада в декабре, наличие стали на заводах, а также значения стоимости транспортировки 1 т стали, приведены в табл.46 (Аксенова Р.Н.) [6].

Таблица 46

Завод	Транспортные издержки, ф. ст. за единицу					Предложение, т
	Торговый склад					
	1	2	3	4	5	

Завод	Транспортные издержки, ф. ст. за единицу					Предложение, т
	Торговый склад					
	1	2	3	4	5	
A	20	27	33	25	34	200
B	22	36	34	28	26	250
C	26	29	27	26	28	300
Потребность, т	100	150	200	100	200	

1) Требуется определить план складов транспортировки стали на декабрь, минимизирующий транспортные издержки, и величину минимальной стоимости транспортировки.

2) Требуется определить план транспортной компании транспортировки стали на декабрь, максимизирующий транспортную выручку, и величину этой выручки.

47. Компании "Zeit pie" принадлежат три фермы, где выращиваются овощи.

Предназначенные для последующей обработки на двух холодильных заводах компании (Аксенова Р.Н.) [6, с. 78]. Одним из выращиваемых овощей являются бобы, которые холодильные заводы продают по 200 ф. ст. за 1 т. Прогнозные значения спроса на следующий сезон равны 2750 т для завода "Craft" и 3250 т для завода "Liver". Ниже приведены (табл. 47.1 и 47.2) издержки производства для каждой фермы и каждого холодильного завода, а также максимальные значения урожая для каждой фермы.

Таблица 47

Ферма, завод	Издержки производства, ф. ст. за 1 т	Максимальный урожай, т
"Ascent Hill"	90	2000
"Midrow Top"	95	3000
"Alum Up"	87	1500
Завод "Craft"	20	2750
"Liver"	23	3250

Таблица 47.2. Стоимость транспортировок.

Ферма	Холодильный завод, ф. ст./ т	
	"Craft"	"Liver"
"Ascent Hill"	10	15
"Midrow Top"	12	12
"Alum Up"	18	9

Требуется.

1. Для ферм и холодильных заводов найти производственный план на следующий сезон, позволяющий получить максимальную прибыль.

2. Администрация компании "Zeit" планирует превратить ферму "Midrow Top" в центр производства бобов высокого качества, вследствие чего она обратила внимание на то, что издержки производства на данной ферме являются самыми высокими и составляют 95 ф. ст. за 1 т. На сколько вы порекомендовали бы снизить эти издержки, прежде чем изменение оптимального распределения перевозок будет целесообразным?

48. Дирекция деревообрабатывающего предприятия "Vibra" приняла на работу.

Пять человек (Аксенова Р.Н.) [6, с.78]. Каждый из них имеет различные способности и навыки и затрачивает различное время на выполнение определенной работы. В настоящее время необходимо выполнить пять видов работ. Время выполнения работы каждым работником приведено в табл. 48:

Таблица 48

Работник	Время выполнения, ч				
	Работы 1	Работы 2	Работы 3	Работы 4	Работы 5

M1	25	16	15	14	13
M2	25	17	18	23	15
M3	30	15	20	19	14
M4	27	20	22	25	12
M5	29	19	17	32	10

Требуется назначить на каждый вид работы одного из работников, чтобы общее время, необходимое для завершения всех видов работ, было минимальным? Величина минимального общего времени?

49. В Kingdom of the Republik of Jdion имеется пять угольных шахт.

Показатели объемов выпуска продукции и издержек производства которых приведены в табл. 49.1 (Аксенова Р.Н.) [6, с.79]:

Таблица 49.1

Шахта	Выпуск продукции, т/день	Издержки производства, ф. ст. за 1 т
1	120	25
2	150	29
3	80	34
4	160	26
5	140	28

До того как уголь будет готов к продаже, его необходимо "очистить" и отсортировать на одном из трех углеперерабатывающих заводов. В табл. 49.2 приведены значения производственных возможностей и эксплуатационных расходов по каждому заводу:

Таблица 49.2

Завод	Выпуск продукции, т/день	Эксплуатационные расходы, ф. ст. за 1 т
A	300	2
B	200	3
C	200	3

Перевозка угля производится по железной дороге, ее стоимость равна 0,5 ф. ст. за 1 т-км. Расстояние от каждой шахты до каждого углеперерабатывающего завода в табл. 49.3 (км):

Таблица 49.3

Углеперерабатывающий завод	Расстояние до шахты, км				
	1	2	3	4	5
A	22	44	26	52	24
B	18	16	24	42	48
C	44	32	16	16	22

1. Построив транспортную модель, определите, как следует распределить перевозки добытого угля с шахт на каждый из трех перерабатывающих заводов.

2. Ввиду установки нового оборудования на шахте 3 ее издержки производства, как ожидается, снизятся до 30 ф. ст. за 1 т. Окажет ли это изменение воздействие, и если да, то какое на распределение перевозок угля на перерабатывающие заводы?

3. Планируется увеличение объема добычи на шахте 5 до 180 т в день, причем его можно достичь, не увеличивая издержки производства 1 т угля. Как это повлияет на распределение перевозок угля к перерабатывающим заводам?

50. Компания "Braintree Electronics Company" выпускает 100-метровые ленты.

К видеокассетам, предназначенные для продажи населению (Аксенова Р.Н.) [6, с.80]. В табл. 50 приведены значения спроса и производственных возможностей за IV квартал.

Таблица 50

Месяц	Спрос, шт	Выпуск продукции в урочное время, шт	Выпуск продукции в сверхурочное время, шт
Октябрь	300	400	150
Ноябрь	450	400	150
Декабрь	800	400	150

Отметим, что производственные возможности позволяют производить ленты к видеокассетам как в течение урочного, так и сверхурочного времени работ, причем если показатели производственных возможностей постоянны, то значение спроса возрастает перед Рождеством. Компания не обладает каким-либо запасом продукции на данный момент и не намерена создавать его после декабря.

Издержки производства 100-метровой ленты к видеокассетам равны 150 ф. ст. в урочное время и 180 ф. ст. в сверхурочное время работы. Было установлено, что стоимость хранения запасов составляет 20 ф. ст. за одну 100-метровую ленту в месяц. При ответе на вопросы примите предпосылку о том, что все заказы удовлетворяются точно в срок, а спрос и предложение возникают в середине каждого месяца.

Найти оптимальный план производства на указанный период. Определить общую стоимость, соответствующую найденному решению.

51. Компания "Royal Wedgetoun Pottery" получила заказы на 3 вида изделий.

Изделия: бокалы, чашки и вазы (Аксенова Р.Н.) [6, с.81]. Заказы необходимо удовлетворить в течение следующей недели. Размеры заказов следующие – табл. 51.1:

Таблица 51.1

ПРОДУКТ	Размер заказа, шт
Бокалы	4000
Чашки	2400
Вазы	1000

В распоряжении компании имеются три станка, на каждом из которых можно производить любой из указанных видов продукции с одинаковой производительностью. Однако единичные затраты по каждому виду продукции варьируют в зависимости от используемого станка. В табл. 51.2 приведены единичные издержки (ф. ст./ изделие) по каждому станку:

Таблица 51.2

Станок	Издержки, ф. ст./изделие		
	Бокалы	Чашки	Вазы
А	1,20	1,30	1,10
В	1,40	1,30	1,50
С	1,10	1,00	1,30

Кроме того, известно, что производственные мощности станков В и С на следующую неделю составят 3000 единиц, а станка А — 2000 единиц.

Найти план производства для видов продукции и станков, минимизирующий общие издержки производства. Определить значение минимальных издержек.

52. В комбинате "Сосенка" приняты на работу 6 человек.

Каждый из них имеет различные способности и навыки и затрачивает различное время на выполнение определенной работы. В настоящее время необходимо выполнить пять видов работ. Время выполнения работы каждым работником приведено в табл. 52 (Аксенова Р.Н.) [6]:

Таблица 52

Работник	Время выполнения, ч				
	Работы 1	Работы 2	Работы 3	Работы 4	Работы 5
P1	32	18	13	16	12
P2	26	17	18	23	15
P3	30	15	20	19	14
P4	27	22	21	25	14
P5	24	19	17	32	9
P6	28	16	19	16	15

Требуется назначить работника на каждый из шести видов работы так, чтобы общее время, необходимое

для завершения всех видов работ, было минимальным? Величина минимального общего времени?

Подсказка. Использовать оптимизатор bintprog. Ввести 30-размерный вектор распределения работ.

53. Компания "Orange Computer" производит матричные печатающие устройства.

Они в настоящее время являются дефицитом (Аксенова Р.Н.) [6, с.81]. Четыре основных покупателя — это крупные специализированные компьютерные универмаги, расположенные в Аббатстауне, Бесвиче, Карлике и Денстоуне, уже подали заявки, общий размер которых превышает общие производственные мощности трех заводов компании в Рексфорде, Сидоне и Тристроне. Компания должна принять решение о том, как распределить производственные мощности, чтобы получить максимальную прибыль.

После того, как каждый принтер тщательно упакован в мягкую упаковку, предохраняющую его от каких-либо повреждений, его помещают в отдельную коробку. В нижеследующей таблице приведены значения стоимости транспортировки одной единицы от каждого завода-производителя в каждый специализированный универмаг (ф. ст.) – табл. 53.

Таблица 53

Город– Универмаг Город– завод	Стоимость транспортировки, ф.ст./ 1 МПУ			
	"Аббатстаун"	"Бесвич"	"Карлик"	"Денстоун"
Рексфорд	22	24	22	30
Сидон	24	20	18	28
Тристрон	26	20	26	24

Поскольку все четыре специализированных универмага расположены в различных частях страны и, следовательно, стоимость транспортировки продукции между заводами-производителями и универмагами различна, а также ввиду некоторых различий и в издержках производства каждого из четырех заводов, существующая структура цен предусматривает возможность установления различных цен для каждого из четырех универмагов. В настоящее время установлены следующие цены за единицу продукции: 230 ф. ст. в Аббатстауне, 235 ф. ст. в Бесвиче, 225 ф. ст. в Карлике и 240 ф. ст. в Денстоуне. Издержки производства на единицу продукции составляют 150 ф. ст. на заводах в Рексфорде и Тристроне, и 155 ф. ст. на заводе в Сидоне.

Значения спроса в Аббатстауне, Бесвиче, Карлике и Денстоуне равны 850, 640, 380 и 230 единицам соответственно. Производственные мощности позволяют производить на заводе в Рексфорде 625, в Сидоне — 825, а в Тристроне — 450 принтеров.

Определить оптимальное распределение перевозок, обеспечивающее максимальную прибыль компании, и величину этой прибыли.

54. Членов Ассоциации ученых Мидленда недавно уведомили.

Их ассоциация получит государственные гранты на проведение исследований в соответствии с четырьмя основными исследовательскими проектами (Аксенова Р.Н.) [6, с.82]. Исполнительный директор ассоциации должен по каждому проекту назначить научного руководителя. В настоящее время эти обязанности можно возложить на одного из пяти исследователей — Адамс, Браун, Карр, Дэй и Иванс. Время, требуемое для завершения каждого из исследовательских проектов, зависит от опыта и способностей исследователя, которому будет поручено руководство выполнением проекта. Исполнительному директору были представлены оценки времени выполнения проекта каждым из ученых – табл. 54.

Таблица 54

Ученый-исследователь	Время выполнения проекта, сутки			
	1	2	3	4
1. Адамс	80	120	60	104
2. Браун	72	144	48	110
3. Карр	96	148	72	120
4. Дэй	60	108	52	92
5. Иванс	64	140	60	96

Поскольку все четыре проекта обладают равным приоритетом в выполнении, исполнительный директор заинтересован в таком назначении научных руководителей, которое бы позволило свести к минимуму общее время (в сутках), требуемое для завершения всех четырех проектов.

Требуется определить оптимальный вариант назначения научных руководителей проектов и, следовательно, общее число суток, необходимое для завершения четырех проектов.

Примечание. Назначение руководителя следует представить как X_{mk} (m – номер работы, k – номер учёного). Причём, $X_{mk} = 0$ (назначения нет) или 1 (назначение состоялось). При этом

$$\sum_m X_{mk} = 1 \text{ (} k=\text{const) и } \sum_k X_{mk} = 1 \text{ (} m=\text{const) и } \sum_{m,k} X_{mk} = 4$$

55. Комбинат выпускает книжные полки А из натурального дерева со стеклом.

Полки В1 из полированной ДСП (древесностружечной плиты) без стекла и полки В2 из полированной ДСП со стеклом (Т.В. Алексинская Т.В.) [14, с. 29]. Габариты полок А, В1 и В2 следующие: длина 1100 (d) мм, ширина 250 (w) мм, высота 300(h) мм (рис. 55). Размер листа ДСП 2×3 м.

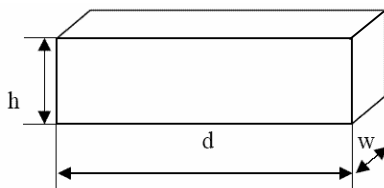


Рис. 55. Габариты полок, выпускаемых мебельным комбинатом

При изготовлении полок А выполняются следующие работы: столярные, покрытие лаком, сушка, резка стекла, упаковка. Все операции, производимые входе столярных работ и упаковки, выполняются вручную. Полки В1 и В2 поставляются в торговую сеть в разобранном виде. За исключением операции упаковки, все остальные операции (производство комплектующих полки, резка стекла) при изготовлении полок В1 и В2, выполняются на специализированных автоматах. Трудоемкость столярных работ по выпуску одной полки А составляет 4(Тр1) ч. Производительность автомата, покрывающего полки А лаком – 10(Пр1) полок в час, автомата, режущего стекло – 100 (Пр2) стекол в час. Сменный фонд времени автомата для покрытия лаком – 7 (ФВ1) ч, автомата для резки стекла – 7,5 (ФВ2) ч. Сушка полок, покрытых лаком, происходит в течение суток в специальных сушилках, вмещающих 50 (V1) полок. На упаковку полки А требуется 4 (Тр2) минуты. В производстве полок заняты 40(Р1) столяров и 14 (Р2) упаковщиков. Производительность автомата, производящего комплектующие полки В1и В2, равна 3 (Пр3) полки в час, а его сменный фонд времени равен 7,4 (ФВ3) ч, трудоемкость упаковочных работ составляет 8 (Тр3) мин для полки В1 и 10(Тр4) мин для полки В2.

От поставщиков комбинат получает в месяц 400 (Z1) листов полированной ДСП, 230 (Z2) листов ДВП (древесноволокнистой плиты), а также 260 (Z3) листов стекла. Из каждого листа ДВП можно выкроить 14 (К1)задних стенок полок В1 и В2, а из каждого листа стекла – 10 (К2) стекол для полок А и В2.Склад готовой продукции может разместить не более 350 (V2) полок и комплектов полок, причем ежедневно в торговую сеть вывозится в среднем 40(N) полок и комплектов. На начало текущего месяца на складе осталось 100(Ост) полок, произведенных ранее. Себестоимость полки А равна 205 (С1) руб., полки В без стекла – 142 (С2) руб., со стеклом – 160 (С3) руб. Маркетинговые исследования показали, что доля продаж полок обоих видов со стеклом составляет не менее 60% (Д) в общем объеме продаж, а емкость рынка полок производимого типа составляет около 5300 (V3) штук в месяц. Мебельный комбинат заключил договор на поставку заказчику 50 (З) полок типа В2 в текущем месяце.

Известны цены реализации полок: полка А – 295 (Ц1) руб., полка В без стекла – 182 (Ц2) руб., полка В со стеклом – 220 (Ц3) руб.

Составьте план производства полок на текущий месяц, обеспечивающий получение комбинатом максимальной прибыли. Вычислите величину максприбыли.

56. На приобретение оборудования для нового участка выделено 108 тыс. руб.

Оборудование должно быть размещено на площади, не превышающей 92 м² [15]. Предприятие может заказать оборудование 4 видов – см. табл. 56.

Таблица 56

Тип машины	Стоимость, тыс. руб	Занимаемая площадь с учётом проходов, м ²	Производительность, ед. продукции/смена	Обслуживающий персонал, чел/маш	Зарплата обслуж.перс., тыс. руб./чел.
А	18	6	9000	1	40000
Б	16	8	7000	1	30000
В	14	11	5500	2	21000
Г	12	14	3700	2	16000

Машин типа А можно заказать не более 4 ед. Фонд зарплаты участка составляет 280 тыс. руб.

Найти оптимальный вариант приобретения оборудования, обеспечивающий максимум общей производительности нового участка.

57. Рацион кормления коров на молочной ферме может состоять из 6 продуктов.

Сена, силоса, соломы, сенажа, концентратов и корнеплодов (моркови, свёклы и др.). Эти продукты содержат питательные вещества: белок, кальций, витамины – см. табл. 57 [15].

Таблица 57

Продукты	Питательные вещества			Стоимость 1 кг, руб	Минимальная суточная потребность, кг	Максимальная суточная потребность, кг
	Белок, г/кг	Кальций, г/кг	Витамины, мг/кг			
Сено	54	10	2	6	1,5	10
Силос	112	6	3	8	–	15
Солома	40	–	–	2	–	5
Сенаж	32,7	3,34	8	2	–	8
Концентраты	180	3	1	24	–	2
Морковь красная	10	0,7	153	8 (опт)	–	–

В расчете на одну корову суточные нормы потребления белка и кальция составляют не менее 1630 и 109 г соответственно. Потребление витаминов строго дозировано и равно 87 мг в сутки.

Требуется составить самый дешёвый суточный рацион, вычислить его стоимость.

58. Отдел кадров предприятия устроил конкурсный набор.

Набор специалистов на две вакантные должности. На эти новые места (НМ) претендуют 3 прежних сотрудника (ПС), уже работающие в других отделах, и 4 новых сотрудника (НС). Номера новых сотрудников, новых и прежних мест выбираются из табл. 58.1 (Алексинская Т.В.) [14]. Номера прежних мест являются номерами прежних сотрудников.

Отдел кадров оценил по десятибалльной шкале компетентность новых сотрудников (табл. 58.2) и прежних сотрудников (табл. 58.3) для работы и на новых местах, и на прежних местах (ПМ), то есть занимаемых прежними сотрудниками. Необходимо учесть, что руководство предприятия, во-первых, предпочитает, чтобы прежние сотрудники не претендовали на места друг друга, и, во-вторых, не намерено увольнять прежних сотрудников.

Необходимо распределить сотрудников по должностям наилучшим образом.

Таблица 58.1

Номера сотрудников и мест их работы для конкретного варианта

Новые сотрудники (НС)	Места работы прежних сотрудников (ПМ)	Новые места (НМ)
5, 6, 7, 8	1, 2, 5	3, 4

Таблица 58.2
Компетентность новых сотрудников

	НМЗ	НМ4	ПМ1	ПМ2	ПМ5
НС 5	6	5	5	4	4
НС6	9	7	6	7	9
НС 7	9	9	8	7	8
НС 8	8	9	7	8	7

Таблица 58.3
Компетентность прежних сотрудников

	НМЗ	НМ4	Занимаемое место
ПС1	6	7	7
ПС2	7	7	8
ПС 5	8	8	7

59. На складах хранится мука, которую необходимо завезти в хлебопекарни.

Номера складов и номера хлебопекарен выбираются в соответствии с табл. 60.1. Текущие тарифы перевозки муки [руб./т], ежемесячные запасы муки [т/мес.] на складах и потребности хлебопекарен в муке [т/мес.] указаны в табл. 60.2 (Вепринская Т) [16].

При этом необходимо учитывать, что из-за ремонтных работ временно нет возможности перевозить муку с некоторых складов в некоторые хлебопекарни. В табл. 59.1 это показано в графе "Запрет перевозки" в формате № склада х № хлебопекарни. Например, «2х5» обозначает, что нельзя перевозить муку со склада № 2 в хлебопекарню № 5. Кроме того, необходимо учесть, что некоторые хлебопекарни имеют договоры на гарантированную поставку муки с определенных складов. В табл. 59.1 это показано в графе "Гарантированная поставка" в формате № склада х № хлебопекарни = объем поставки: «4х3=45» обозначает, что между складом № 4 и хлебопекарней № 3 заключен договор на обязательную поставку 45 т муки.

Необходимо организовать поставки наилучшим образом по данным табл. 59.2, учитывая, что мука хранится и транспортируется в мешках весом по 50 кг.

Таблица 59.1
Номера складов, хлебопекарен, запрещенные и гарантированные поставки

№ Складов	№ Хлебопекарен	Запрет перевозки	Гарантированная поставка, т/мес.
2, 3, 4	2, 3, 4, 5	3х3, 2х5	4х3=45

Таблица 59.2
Запасы, потребности и тарифы перевозок (руб/т)

Склады	Хлебопекарни				
	2	3	4	5	Запас, т/мес.
2	100, руб/т	500	600	500	60
3	200	100	600	300	60
4	700	200	400	900	55
Спрос, т/мес.	56,78	58,88	62,44	73,92	—

60. По заказу пяти потребителей А, Б, В, Г, Д.

На 4 предприятиях-изготовителях производится штучная продукция. В процессе доставки к потребителям продукция может храниться на 3 оптовых базах. Существует следующая система организации снабжения потребителей продукцией (Алексинская Т.В.) [14, с.61]:

ИЗГОТОВИТЕЛЬ → ОПТОВАЯ БАЗА → ПОТРЕБИТЕЛЬ,

то есть вся продукция, произведенная изготовителями, сначала складывается на оптовых базах и только потом развозится потребителям.

Ежемесячный спрос на продукцию [шт.], емкость оптовых баз [шт.] и тарифы [руб./шт.] за доставку продукции с оптовых баз к потребителям приведены в табл. 60.1. Ежемесячные объемы производства [шт.], емкость оптовых баз [шт.] и суммарные затраты [руб./шт.] на производство и доставку продукции от изготовителей к оптовым базам приведены в табл. 60.2.

Таблица 60.1
Параметры перевозок из оптовых баз к потребителям

Оптовая база	Потр-ль А	Потр-ль Б	Потр-ль В	Потр-ль Г	Потр-ль Д	Запас, шт
1	12, руб/шт	32	14	22	36	540
2	16	20	31	20	17	380
3	16	26	32	34	10	690
Спрос на товар, шт	600	550	420	780	400	—

Таблица 60.2
Параметры перевозок от изготовителей к оптовым базам

Изготовитель	Оптовая база 1	Оптовая база 2	Оптовая база 3	Производство, шт/мес
1	18	14	27	510
2	19	14	36	620
3	7	18	30	660
4	10	15	15	420
Запас, шт	540	380	690	—

Необходимо выбрать оптимальный способ организации снабжения потребителей продукцией предприятий-изготовителей, обеспечивающий минимальные затраты в системе снабжения.

61. По заказу пяти потребителей А, Б, В, Г, Д на 4 предприятиях-изготовителях.

Производится штучная продукция. Существует следующая система организации снабжения потребителей продукцией (Алексинская Т.В.) [14, с.62]:

ИЗГОТОВИТЕЛЬ → ПОТРЕБИТЕЛЬ,

то есть вся продукция, произведенная изготовителями, напрямую доставляется потребителям, минуя оптовые базы.

Ежемесячные объемы производства [шт.], спрос на продукцию [шт.] и суммарные затраты [руб./шт.] на производство и доставку продукции от изготовителей к потребителям приведены в табл. 61.1.

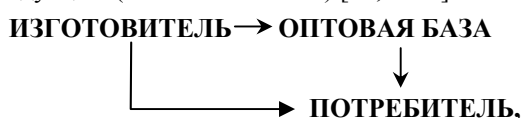
Таблица 61.1
Параметры перевозок от изготовителей к потребителям

Изготовитель	Потр-ль А	Потр-ль Б	Потр-ль В	Потр-ль Г	Потр-ль Д	Произ-во, шт/мес
1	10, руб/шт	2	1	10	20	510
2	20	48	30	25	15	620
3	10	60	20	38	44	660
4	28	40	18	28	18	420
Спрос на товар, шт	600	550	420	780	400	—

Необходимо выбрать оптимальный способ организации снабжения потребителей продукцией предприятий-изготовителей, обеспечивающий минимальные затраты в системе снабжения.

62. По заказу пяти потребителей А, Б, В, Г, Д на 4 предприятиях-изготовителях.

Производится штучная продукция. Существует следующая система организации снабжения потребителей продукцией (Алексинская Т.В.) [14, с. 62]:



то есть продукция, произведенная изготовителем, доставляется к потребителям частично напрямую, а частично транзитом через оптовые базы.

Ежемесячные объемы производства [шт.], спрос на продукцию [шт.] и суммарные затраты [руб./шт.] на производство и доставку продукции от изготовителей к оптовым базам, от оптовых баз к потребителям и от изготовителей к потребителям приведены соответственно в табл. 62.1... 62.3.

Таблица 62.1
Параметры перевозок от изготовителей к оптовым базам

Изготовитель	Оптовая база 1	Оптовая база 2	Оптовая база 3	Производство, шт/мес
1	18	14	27	510
2	19	14	36	620
3	7	18	30	660
4	10	15	15	420
Запас, шт	540	380	690	–

Таблица 62.2
Параметры перевозок из оптовых баз к потребителям

Оптовая база	Потр-ль А	Потр-ль Б	Потр-ль В	Потр-ль Г	Потр-ль Д	Запас, шт
1	12, руб/шт	32	14	22	36	540
2	16	20	31	20	17	380
3	16	26	32	34	10	690
Спрос на товар, шт	600	550	420	780	400	–

Таблица 62.3
Параметры перевозок от изготовителей к потребителям

Изготовитель	Потр-ль А	Потр-ль Б	Потр-ль В	Потр-ль Г	Потр-ль Д	Произ-во, шт/мес
1	10, руб/шт	2	1	10	20	510
2	20	48	30	25	15	620
3	10	60	20	38	44	660
4	28	40	18	28	18	420
Спрос на товар, шт	600	550	420	780	400	–

Необходимо выбрать оптимальный способ организации снабжения потребителей продукцией предприятий-изготовителей, обеспечивающий минимальные затраты в системе снабжения.

63. На АО “Светлана” подготовлены к серийному производству 5 новых изделий.

Изделия: И1, И2, И3, И4, И5, оптовые цены O_{Cj} которых равны соответственно (46, 27, 40, 35, 23) [руб/шт]. Производство может быть развернуто в четырех сборочных цехах: Ц1, Ц2, Ц3, Ц4. Затраты в рублях на изготовление в i -м цехе j -го изделия задаются матрицей $C=(c_{ij})$. Фонды времени F_i работы цехов в плановом периоде равны соответственно 550, 870, 620, 790 часов, план выпуска продукции P_j в штуках составляет соответственно 6 400, 8 700, 16 400, 4 800, 4 600, а трудоёмкость в минутах изготовления t_{ij} в i -м цехе одной единицы изделия I_j задается матрицей (t_{ij}) . Численные значения упомянутых величин представлены в табл. 63 (Алексинская Т.В.) [14, с. 69].

Таблица 63
Планирование серийного производства изделий И1...И5

Цех	Изготовление изделий: трудоёмкость, мин/шт себестоимость, руб/шт					Фонд времени работы цеха, ч
	И1	И2	И3	И4	И5	
1	3, мин/шт	0,5	2	4	6	550
	8, руб/шт	19	7	21	9	
2	3,6	0,6	2,4	4,8	7,2	870
	43	12	40	26	15	
3	6	1	4	8	12	620
	9	18	23	27	20	
4	7,2	1,2	4,8	9,6	14,4	790
	21	16	22	13	21	
План, шт	6400	8700	16 400	4800	4600	—
Оптовая цена, руб/шт	46	27	40	35	23	—

Составьте план производства изделий, обеспечивающий максимизацию прибыли АО Светлана. Вычислите величину прибыли.

64. На АО “Позитрон” подготовлены к серийному производству 5 новых изделий.

Изделия: И1 , И2, И3, И4, И5 , оптовые цены ОЦ_ј которых равны соответственно 46, 27, 40, 35, 23 руб/шт. Производство может быть развёрнуто в четырёх сборочных цехах: Ц1 , Ц2 , Ц3 , Ц4 . Затраты в рублях на изготовление в i-м цехе j-го изделия задаются матрицей C=(c_{ij}). Фонды времени F_i работы цехов в месячном плановом периоде равны соответственно 550, 870, 620, 790 часов, план выпуска продукции P_j в штуках составляет соответственно 6 400, 8 700, 16 400, 4 800, 4 600, а трудоёмкость в минутах изготовления t_{ij} в i-м цехе одной единицы изделия И_j задается матрицей (t_{ij}). Кроме того, планируется осуществить модернизацию оборудования для производства одного изделия в одном из цехов. Себестоимость производства изделия на модернизированном оборудовании снизится на 25%. Затраты на модернизацию (тыс. руб), а также все упомянутые данные представлены в табл. 64.

Таблица 64
Планирование серийного производства изделий И1...И5

Цех	Изготовление изделий: трудоёмкость, мин/шт// себестоимость, руб/шт//затраты на модернизацию цеха по изделию, тыс. руб					Фонд времени работы цеха за месяц, ч
	И1	И2	И3	И4	И5	
1	3, мин/шт//	0,5//	2//	4//	6//	550
	8, руб/шт//	19//	7//	21//	9//	
	72 т.руб	90	134	162	110	
2	3,6//	0,6//	2,4//	4,8//	7,2//	870
	43//	12//	40//	26//	15//	
	62	80	115	64	55	
3	6//	1//	4//	8//	12//	620
	9//	18//	23//	27//	20//	
	77	82	151	78	42	
4	7,2//	1,2//	4,8//	9,6//	14,4//	790
	21//	16//	22//	13//	21//	
	122	103	52	65	74	
План, шт	6400	8700	16 400	4800	4600	—
Оптовая цена, руб/шт	46	27	40	35	23	—

Необходимо рассчитать:

1) план загрузки цехов без проведения модернизации из условия получения максимальной прибыли АО Позитрон;

2) план загрузки цехов с проведением модернизации (цех и изделие на выбор решающего) из условия получения максимальной прибыли АО Позитрон с включение всех затрат на модернизацию в планируемый месяц;

3) если окажется, что прибыль в п. 2) меньше прибыли из п.1), определить месяц, когда будет обеспечено неравенство: $\text{Приб2} \geq \text{Приб1}$.

Пояснение. Для выполнения п.3 (если $\text{Приб2} < \text{Приб1}$) следует начиная с 2 месяцев с шагом 1 увеличивать количество месяцев, на которые равномерно распределяются затраты на модернизацию с вычислением плана загрузки для полученной уменьшенной величины затрат. Расчет можно производить «вручную» или с использованием программы ОптЦФ(P1) из рис. 4.5 разд. 4 Методуказаний. Параметром P1 будет вектор модернизационных затрат с компонентами, уменьшающимися с увеличением их номера j по закону: $\text{НачМодернЗатр}/j$, где НачМодернЗатр – величина из табл. 64.

65. Промышленное предприятие «Белподшип» может выпускать подшипники.

Четырёх видов (П1, П2, П3, П4). На их изготовление необходимо использовать три вида производственных ресурсов: кольца, шарики, сепараторы. Наличие ресурсов в плановом периоде и норма их расхода на единицу продукции указаны в таблице 65.1.

Таблица 65.1

Вид ресурса	Наличие ресурса, шт.	Норма расхода на единицу продукции, шт./ед.			
		П1	П2	П3	П4
Кольца	600	2	2	2	2
Шарики	2000	6	8	10	6
Сепараторы	550	2	1	1	2

Для изготовления данных видов подшипников необходимо оборудование трех видов: токарное, шлифовальное, измерительная техника. Наличие оборудования и потребляемая им мощность, требуемая на изготовление подшипников каждого вида, указаны в таб. 65.2.

Таблица 66.2

Вид оборудования	Наличие оборудования, станко-ч	Норма расхода на единицу продукции, станко-ч/ед.			
		П1	П2	П3	П4
Токарное	65	0,2	0,3	0,1	0,2
Шлифовальное	30	0,1	0,1	0,2	0,2
Измерительная техника	85	0,3	0,2	0,1	0,1

Маркетинговый отдел данного предприятия выявил, что в экономическом регионе есть два крупных потребителя подшипников данных видов: Россия и Беларусь. Общая потребность в подшипниках по экспертным оценкам в настоящее время составляет порядка тех объёмов, которые представлены в таблице 65.3.

Таблица 65.3

Подшипник	Рынки	Планируемый объём продаж, шт.	Планируемая прибыль от единицы продукции, руб.
П1	Россия	100	700
	Беларусь	200	880
П2	Россия	150	500
П3	Россия	100	486
П4	Беларусь	Не ограничен	600

Определить план производства подшипников, обеспечивающий максимальную прибыль предприятию «Белподшип», и величину этой прибыли.

66. Продукция городского молочного завода.

Это молоко, кефир и сметана, расфасованные в пакеты (И.Л. Акулич) [5, с.7]. На производство 1 т молока, кефира и сметаны требуется соответственно 1010, 1010 и 9450 кг молока. При этом затраты рабочего време-

ни при разливе 1 т молока и кефира составляют 0,18 и 0,19 машино-ч. На расфасовке 1 т сметаны заняты специальные автоматы в течение 3,25 ч. Всего для производства цельномолочной продукции завод может использовать 136 000 кг молока. Основное оборудование может быть занято в течение 21,4 машино-ч, а автоматы по расфасовке сметаны — в течение 16,25 ч. Прибыль от реализации 1 т молока, кефира и сметаны соответственно равна 7500, 6 250 и 32 000 руб. Завод должен ежедневно производить не менее 60 т молока, 20 т кефира и 3 т сметаны, расфасованных в пакеты. На производство дополнительной продукции нет никаких ограничений.

Требуется определить, какую продукцию и в каком количестве следует ежедневно изготавливать заводу, чтобы прибыль от ее реализации была максимальной. Величина этой прибыли?

67. Для поддержания нормальной жизнедеятельности среднему человеку...

Ежедневно необходимо потреблять не менее 75 г белков, 72 г жиров, 11 г полиненасыщенных жирных кислот, 365 г углеводов, 8 г минеральных солей и пищевой энергии 2500 ккал. Количество питательных веществ, содержащихся в 1 кг каждого вида заданного набора продуктов, а также цена 1 кг каждого из этих продуктов приведены в табл. 67 (И.Л. Акулич) [5, с. 89].

Таблица 67

Содержание (г) питательных веществ 1 кг продуктов

Питательные вещества	мясо	рыба	молоко	масло сливочное	масло постное (подсолнечное)	сыр	крупы (гречневая)	картофель
Белки	180	190	30	10	—	260	126	21
Жиры	20	3	25	865	351	310	33	2
Углеводы	—	—	50	6	—	—	621	200
Минеральные соли	9	10	7	12	—	60	20	10
Полиненасыщенные жирные кислоты	—	—	—	—	598	—	—	—
Пищевая энергия, ккал/кг	2180	1270	580	7480	8990	3520	3350	800
Цена 1 кг продуктов (руб.)	250	74	28	120	45	215	18	14

Составить дневной рацион из заданного набора продуктов, содержащий не менее минимальной суточной нормы потребности человека в необходимых питательных веществах и энергии при минимальной общей стоимости потребляемых продуктов. Величина этой стоимости?

68. Диспетчерская служба имеет следующие минимальные потребности.

В количестве диспетчеров в различное время суток (табл. 68) (Е.Н. Бендерская) [1, с.24]:

Таблица 68

Порядковый номер периода	Время суток, час	Минимальное число диспетчеров, требуемое в указанный период
1	2-6	20
2	6-10	50
3	10-14	80
4	14-18	100
5	18-22	40
6	22-2	30

При этом нужно иметь в виду, что период 1 следует сразу же за периодом 6. Каждый диспетчер ежедневно приступает к работе в начале определенного периода и работает восемь часов без перерыва. Требуется составить расписание выхода групп диспетчеров в каждый период суток таким образом, чтобы обойтись минимальным числом диспетчеров, не нарушая сформулированных выше требований.

69. Завод должен изготовить 3 типа стальных сплавов.

Сплавы состоят из чистой стали и переработанного металлолома (Е.Н. Бендерская) [1, с.24]. Отношение

веса металлолома к весу чистой стали в сплавах не должно превышать: 85, 72 и 56 процентов для 1-го, 2-го и 3-го сплавов. Производственные затраты на 1 т стали составляют 5000 руб., а затраты на 1 т металлолома 3000 руб.

Заказ предусматривает поставку не менее: 5, 4 и 6 т литья 1-го, 2-го и 3-го сплавов. Предназначенные для литья запасы чистой стали составляют 8 т, а металлолома – 12 т.

Производственно-технологические условия таковы, что на процессы плавки и литья может быть отведено не более 36 час., при этом на 1 т стали расходуется от 2,5 до 3 час., а на 1 т металлолома – от 1,5 до 2 час.

Цель завода - выполнить заказ с минимальными производственными затратами.

70. Компания по производству гусеничных механизмов.

Выпускает пять сходных друг с другом товаров — А, В, С, D, и Е (Аксенова Р.Н.) [6, с.43]. В табл. 70 представлены расходы ресурсов, необходимых для выпуска единицы каждого товара, а также недельные запасы каждого ресурса и цены продажи единицы каждого продукта.

Таблица 70

Ресурсы	Товар					Недельный запас ресурсов
	А	В	С	Д	Е	
Сырье, кг	6,00	6,50	6,10	6,10	6,40	35000
Сборка, ч	1,00	0,75	1,25	1,00	1,00	6000
Обжиг, ч	3	4,50	6	6	4,50	30000
Упаковка, ч	0,50	0,50	0,50	0,75	1,00	4000
Цена продажи, ф.ст.	40	42	44	48	52	

Известны также издержки, связанные с использованием каждого вида ресурсов:

сырье — 2,10 ф. ст. за 1 кг;

сборка — 3,00 ф. ст. за 1 ч;

обжиг — 1,30 ф. ст. за 1 ч;

упаковка — 8,00 ф. ст. за 1 ч.

Требуется определить недельный план выпуска изделий, максимизирующий недельный доход компании.

71. В отделе технического контроля фирмы работают контролеры.

Разряды контролёров: 1 и 2 [13]. Норма выработки ОТК за 8 - часовой рабочий день составляет не менее 1800 изделий. Контролер разряда 1 проверяет 25 изделий в час, причем не ошибается в 98% случаев. Контролер разряда 2 проверяет 15 изделий в час; его точность составляет 95%.

Заработная плата контролера разряда 1 равна 200 руб. в час, контролер разряда 2 получает 150 руб. в час. При каждой ошибке контролера фирма несет убыток в размере 100 руб. Фирма может использовать 8 контролеров разряда 1 и 10 контролеров разряда 2. Руководство фирмы хочет определить оптимальный состав ОТК, при котором общие затраты на контроль (зарплата плюс убытки) будут минимальны.

72. Для обогрева помещений используют четыре агрегата.

Каждый из агрегатов может работать на любом из пяти сортов топлива, имеющемся в количествах 90, 110, 70, 80 и 150 т (Аксенова Р.Н.) [6, с.85]. Потребность в топливе каждого из агрегатов соответственно равна 80, 120, 140 и 160 т. Теплотворная способность (10^{-4} *ккал/кг) i -го сорта топлива при использовании его на j -м агрегате задается матрицей c_{ij} .

$$(c_{ij}) = \begin{array}{|c|c|c|c|c|} \hline 8 & 7 & 9 & 11 & 8 \\ \hline 6 & 5 & 8 & 7 & 6 \\ \hline 7 & 11 & 5 & 8 & 7 \\ \hline 9 & 8 & 7 & 9 & 11 \\ \hline \end{array}$$

Найти такое распределение топлива между агрегатами, при котором получается максимальное количество теплоты (в Мкал) от использования всего топлива.

73 . При производстве четырех типов кабеля выполняется пять групп операций.

Нормы затрат на 1 км кабеля данного типа на каждой из групп операций, прибыль от реализации 1 км каждого типа кабеля, а также общий фонд рабочего времени, в течение которого могут выполняться эти операции, указаны в табл. 73 (Аксенова Р.Н.) [6, с.85]. В планируемом квартале предприятие должно выпустить не менее 3 км кабеля типа 2. Кроме того, у предприятия есть долг по кабелю типа 4 в объеме 2,6 км, оплаченный заказчиком и который нужно погасить в планируемом квартале. Последнее означает, что прибыль за этот долг предприятие не получит.

Таблица 73

Технологическая операция	Нормы затрат времени (час) на производство 1 км кабеля типа				Общий фонд рабочего времени (час)
	1	2	3	4	
Волочение	1,2	1,8	1,6	2,4	7 200
Наложение изоляций	1,0	0,4	0,8	0,7	5 600
Скручивание элементов в кабель	6,4	5,6	6,0	8,0	11 176
Освинцование	3,0	-	1,8	2,4	3 600
Испытание и контроль	2,1	1,5	0,8	3,0	4 200
Прибыль от реализации 1 км кабеля (тыс. руб.)	40	27	33	45	-

Определить план выпуска кабелей, при котором общая прибыль от реализации изготавливаемой продукции является максимальной.

74. Цех выпускает комплекты из 3 деталей.

Они изготавливаются на двух станках (С.А.Ашманов) [2, с. 107]. На рис. 74 показана технологическая схема изготовления детали каждого вида с указанием времени ее обработки на станках.



Рис. 74

Задан суточный ресурс рабочего времени каждого станка: b мин для станка 1, c мин для станка 2, 710 и 830 мин – для станков 3 и 4. Цена одной детали вида 1, 2 и 3 составляет 3, 1 и 2 руб. соответственно. В комплект входят: 2 дет. Вида 1, 4 – вида 2, 7 – вида 3. Требуется составить суточный план производства деталей с целью:

А) максимизации количества комплектов для одного из вариантов табл. 74 с оценкой их суммарной стоимости;

Б) максимизации стоимости выпущенных деталей для одного из вариантов табл. 74, считая, что они не составляют комплект, с оценкой по результату фактического количества комплектов и количества деталей по видам, не вошедших в комплект.

Таблица 74

Вар	a	b	c	Вар	a	b	c	Вар	a	b	c	Вар	a	b	c
1	3	600	900	6	5	840	450	11	5	870	900	16	4	870	450
2	3	960	600	7	3	660	900	12	5	930	450	17	5	630	270
3	3	510	750	8	3	960	510	13	3	720	900	18	3	600	750
4	4	690	450	9	4	600	900	14	3	960	420	19	4	720	900
5	5	660	900	10	4	780	450	15	4	660	900	20	4	750	360

75. Ресторан работает 7 дней в неделю.

Повара работают 6 часов в день, 5 дней подряд и затем 2 дня отдыхают (Е. В. Алексеева) [10, с. 92]. У всех поваров одинаковая зарплата. Приготовление каждого блюда занимает определенное время, и для каждого дня недели установлено общее необходимое количество часов для приготовления пищи. Данные приведены в табл. 75. Администратору нужно решить, какое количество поваров нанять и в какие дни они должны работать, чтобы нужное количество часов было отработано, а затраты на оплату труда были минимальными.

Таблица 75

Ежедневное количество рабочих часов

День недели	Требуемое количество часов
понедельник	150
вторник	200
среда	400
четверг	300
пятница	700
суббота	800
воскресенье	300

Постройте математическую модель. Найдите оптимальное решение.

76. Авиакомпания «Альфа» составляет расписание вылетов.

Из Чикаго по следующим направлениям: Колумбия, Денвер, Лос-Анджелес и Нью-Йорк (Е. В. Алексеева) [10, с. 92]. В каждый город должен состояться ровно один вылет. Вылеты могут быть в 8:00, 10:00 и 12:00. Авиакомпания оплачивает вылет каждого самолета по каждому направлению. Эти затраты составляют 5 тыс. у.е., если вылет совершается до 10:00 включительно, и 3 тыс. у.е. — после 10:00. В каждый момент времени выполняется не более двух рейсов. Кроме того, если в определенное время есть вылет в Нью-Йорк, то в это же время должен быть вылет в Лос-Анджелес. Ожидаемый доход (в тыс. у.е.) от полетов приводится в следующей табл. 76.

Таблица 76. Ожидаемый доход от полетов, тыс. у.е.

Направление	Время вылета		
	8:00	10:00	12:00
Колумбия	10	6	6
Денвер	9	10	9
Лос-Анджелес	14	11	10
Нью-Йорк	18	15	10

Требуется составить расписание, доставляющее максимальную прибыль авиакомпании. Постройте математическую модель. Найдите оптимальное решение.

77. В лесничестве.

Нужно принять решение, какие участки леса будут вырублены под застройку. Лес разделен на 16 участков прямоугольной формы, как показано в табл. 77.1 (Е. В. Алексеева) [10, с. 96].

Таблица 77.1. Разделение леса на участки

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Соседние участки нельзя выбирать под застройку, чтобы избежать больших вырубленных площадей. Соседними для шестого участка являются, например, участки с номерами 2, 5, 7 и 10. Польза от застройки каждого участка в баллах приводится в табл. 77.2.

Таблица 77.2. Полезность застройки каждого участка, баллы

5	6	2	5
4	3	5	8
8	2	4	9
10	1	6	7

Определите, какие участки выбрать, чтобы общая польза от их застройки была максимальной. Постройте математическую модель, найдите оптимальное решение задачи.

78. Импресарио готовит выставку старинных автомобилей.

Среди которых могут быть Buggati, Cadillac, Cobra, Corvette, Pierce Arrow, Studebaker (Е. В. Алексеева) [10, с. 97]. Опрос показал, что посмотреть именно Buggati придут 58 специально приглашенных гостей, Cadillac — 37, Cobra — 42, Corvette — 40, Pierce Arrow — 55 и Studebaker — 33. Бюджет организации выставки составляет 15 млн у.е. Стоимость доставки автомобиля на выставку и обеспечение его сохранности составляют 6, 4, 3.8, 4.2, 5.5 и 3.2 млн у.е. соответственно. Задача импресарио в том, чтобы привлечь как можно больше специально приглашенных гостей, не превышая бюджет на организацию. Кроме того, на выставке должно быть не менее трех старинных автомобилей. Если Corvette будет выбран для выставки, то и Cobra должен там быть. Если же Buggati отсутствует, то обязательно нужно включить в показ Cadillac.

Постройте математическую модель и найдите оптимальное решение задачи. Определите, каким может быть минимальный и максимальный бюджет, чтобы выставка состоялась.

79. Электроэнергетическая сеть.

На рис. 79 изображена электроэнергетическая сеть, соединяющая электрогенераторы G1... G4 с источниками потребления C1... C4 (Е. В. Алексеева) [10, с. 99]. Потоки энергии могут идти в любом направлении по ребрам сети, пропускные способности ребер неограниченны, стоимость передачи по одному ребру составляет 11 у.е. за 1000 кВт/час. Мощность каждого электрогенератора и стоимость выработки электроэнергии приводятся в табл. 79.1.

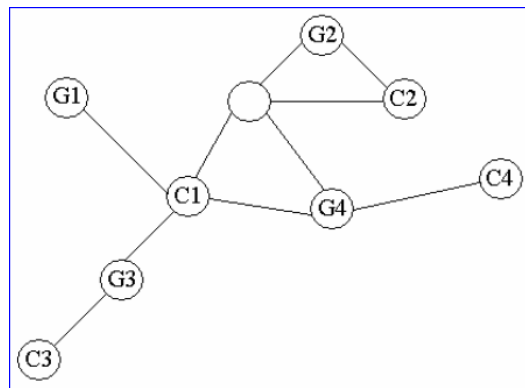


Рис. 79. Электроэнергетическая сеть.

Таблица 79.1. Данные о выработке электроэнергии

	Электрогенераторы G1 G 2 G3 G4			
Мощность (тыс. кВт/час)	100	60	80	150
Стоимость выработки (за 1000 кВт/час), у.е.	15	13,5	21	23,5

Энергопотребление в C1...C4 составляет 35, 50, 60 и 40 тыс. кВт соответственно. Какие электрогенераторы из сети использовать, чтобы суммарные затраты были минимальные, а потребность в электроэнергии была удовлетворена? Постройте математическую модель и найдите оптимальное решение задачи.

80. Винодел смешивает четыре сорта винограда.

Получает три сорта вина. Количество виноградного сока, имеющегося в наличии (литров), пропорции смешивания и стоимость каждого сорта вина указаны в табл. 80 (Е. В. Алексеева) [10, с. 102].

Таблица 80. Данные о компонентах для производства вина

Сорт вина	Алиготе	Сорт винограда		Шардоне	Стоимость вина (за литр)
		Изабелла	Рислинг		
Каберне	*	> 75% в любой пропорции		>4%	80
Шардоне	*	> 10%	*	<35%	50
Мускат	*	> 35%		*	35
наличие (в литрах)	130	200	150	350	

* — в любом количестве.

Какие сорта винограда и в каком объеме необходимо смешивать, чтобы получить максимальный доход от продажи вина? Постройте математическую модель и найдите оптимальное решение задачи.

Пример оформления отчета по решению линпрогзадачи из разд. 7

(Методика решения – в разд. 6)

Отчёт о решении линпрогзадачи

Гр. ?. Иванов И.И.

вар. 32

Объединение «Комфорт» производит холодильники, газовые плиты, морозильные шкафы и электропечь по цене 200, 180, 250 и 100 р. соответственно. Постоянным фактором, ограничивающим объемы производств является фиксированная величина трудовых ресурсов - 12 000 человеко-часов в месяц. Выяснилось, однако, что в ближайший месяц дефицитной будет и листовая сталь для корпусов указанных изделий, поскольку поставщики смогут обеспечить лишь 7000 м² этого материала.

Требуется составить план производства на данный месяц, максимизирующий стоимость выпущенной продукции (доход объединения). Известно, что для изготовления холодильника требуется 2 м² листовой стали и 3 чел.-ч рабочего времени, для газовой плиты — соответственно 1,5 м² и 3 чел.-ч, для морозильного шкафа — 3 м² и 2,7 чел.-ч, для электропечи — 1 м² и 2 чел.-ч.

Составление математического описания задачи

А) Неизвестные

x1-количество холодильников;

x2-количество газовых плит;

x3-количество морозильных шкафов;

x4-количество электропечей;

Б) Целевая функция:

$F_c = 200x_1 + 180x_2 + 250x_3 + 100x_4$;

В) Ограничения-неравенства:

$3x_1 + 3x_2 + 2.7x_3 + 2x_4 \leq 12000$ чел-час (ограничения трудовых ресурсов).

Г) Ограничения-равенства:

$2x_1 + 1.5x_2 + 3x_3 + x_4 = 7000$ м² (ограничение по запасу материала изделий).

Д) Область определения ц-функции F_c

Нижние пределы: $x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0$. Верхние: не определены.

Е) Условия задачи для вставки в универсальное линпрогуказание LinProgUnivUkz_Tab Матлаба

Lmi_Lma=-1; % Поиск линпрогмакса. Удалите эту строку при поиске линпрогмина.

Lmi_Lma=1; % Поиск линпрогмина. Удалите эту строку при поиске линпрогмакса.

fo=[200 180 250 100]; % Коэффициенты ц-функции F_c

Ao=[3 3 2.7 2; 0 0 0 0; 0 0 0 0; 0 0 0 0]; bo=[12000 0 0 0]; % Ограничения-неравенства

Aeqo=[2 1.5 3 1; 0 0 0 0; 0 0 0 0]; beqo=[7000 0 0 0]; % Ограничения-равенства

LBo=zeros(4,1); UBo=[inf inf inf inf]; % Область определения ц-функции F_c

Примечание: серым цветом отмечена удалённая строка, т.к. в задаче требуется найти максимум.

Ж) Решение задачи

Таблица 1. TabLPrma_23_Aug_2016 23h44m10 N1_98901

xma_1_98901	yua_1_98901_23_Aug_2016	efma_1_98901	nrma_1_98901_Ax	nrma_1_98901_b	rvma_1_98901_Aeqx	rvma_1_98901_beq
1.1158e-006	7.7333e+005	1	1.2000e+004	12000	7.0000e+003	7000
3.4545e+003			0	0	0	0
606.0606			0	0	0	0
9.5056e-006			0	0	0	0

З) Анализ результата

xma_1_98901

1.1158e-006, холодильники

3.4545e+003, газовые плиты

606.0606, морозильные шкафы

9.5056e-006, электропечи

yта_1_98901_23_Aug_2016

7.7333e+005, доход Объединения

$A \cdot x \leq b$, ограничения по трудовым ресурсам – см. столбцы 4 и 5

$Aeq = beq$, ограничения по запасу материала изделий – см. столбцы 6 и 7

И) Выводы

1. Следует производить в месяц: 3454 газовых плит и 606 морозильных шкафов, а холодильников и электропечей – не производить.
2. Максимальная стоимость выпущенной продукции составит 773 330 руб.
3. Ограничения по трудовым ресурсам (12 000 чел-час) и запасам материала изделий (7000 м²) удовлетворены.

К) Приложения

- 1) Файл линпрогуказания LinProgUnivUkz_Tab с вставленными условиями решения задачи, 2 КБ.
- 2) Файл линпрогоптимизатора LinProgUniversAvtomatMU_Tab с настроенным шаблоном html-итогаблиц для максимума.
- 3) Мат-файл решения WS_ZadIvanov32.mat, 1 КБ.
- 4) Файл шаблона итогтаблиц для максимума ItogTabLPrma.rpt, 1 КБ.
- 5) Файл конструкции итогтаблиц для максимума buildItogTabLPrma.m, 1 КБ.
- 6) Файл итогтаблицы решения задачи TabLPrma_23_Aug_2016 23h44m10 N1_98901.html.

Приложение 2

Программа linprog

№ строки	Программа linprog_Matlab 2009b
1	function [x,fval,exitflag,output,lambda]=linprog(f,A,B,Aeq,Beq,lb,ub,x0,options)
2	%LINPROG Linear programming.
3	% X = LINPROG(f,A,b) attempts to solve the linear programming problem:
4	%
5	% min f'*x subject to: A*x <= b
6	% x
7	%
8	% X = LINPROG(f,A,b,Aeq,beq) solves the problem above while additionally
9	% satisfying the equality constraints Aeq*x = beq.
10	%
11	% X = LINPROG(f,A,b,Aeq,beq,LB,UB) defines a set of lower and upper
12	% bounds on the design variables, X, so that the solution is in
13	% the range LB <= X <= UB. Use empty matrices for LB and UB
14	% if no bounds exist. Set LB(i) = -Inf if X(i) is unbounded below;
15	% set UB(i) = Inf if X(i) is unbounded above.
16	%
17	% X = LINPROG(f,A,b,Aeq,beq,LB,UB,X0) sets the starting point to X0. This
18	% option is only available with the active-set algorithm. The default
19	% interior point algorithm will ignore any non-empty starting point.
20	%
21	% X = LINPROG(f,A,b,Aeq,beq,LB,UB,X0,OPTIONS) minimizes with the default
22	% optimization parameters replaced by values in the structure OPTIONS, an
23	% argument created with the OPTIMSET function. See OPTIMSET for details.
24	% Options are Display, Diagnostics, TolFun, LargeScale, MaxIter.
25	% Currently, only 'final' and 'off' are valid values for the parameter
26	% Display when LargeScale is 'off' ('iter' is valid when LargeScale
27	% is 'on').
28	%
29	% X = LINPROG(PROBLEM) finds the minimum for PROBLEM. PROBLEM is a
30	% structure with the vector 'f' in PROBLEM.f, the linear inequality
31	% constraints in PROBLEM.Aineq and PROBLEM.bineq, the linear equality
32	% constraints in PROBLEM.Aeq and PROBLEM.beq, the lower bounds in
33	% PROBLEM.lb, the upper bounds in PROBLEM.ub, the start point
34	% in PROBLEM.x0, the options structure in PROBLEM.options, and solver
35	% name 'linprog' in PROBLEM.solver. Use this syntax to solve at the
36	% command line a problem exported from OPTIMTOOL. The structure PROBLEM
37	% must have all the fields.
38	%
39	% [X,FVAL] = LINPROG(f,A,b) returns the value of the objective function
40	% at X: FVAL = f'*X.
41	%
42	% [X,FVAL,EXITFLAG] = LINPROG(f,A,b) returns an EXITFLAG that describes
43	% the exit condition of LINPROG. Possible values of EXITFLAG and the
44	% corresponding exit conditions are

№ строки	Программа linprog_Matlab 2009b
45	% 1 LINPROG converged to a solution X.
46	% 0 Maximum number of iterations reached.
47	% -2 No feasible point found.
48	% -3 Problem is unbounded.
49	% -4 NaN value encountered during execution of algorithm.
50	% -5 Both primal and dual problems are infeasible.
51	% -7 Magnitude of search direction became too small; no further
52	% progress can be made. The problem is ill-posed or badly
53	% conditioned.
54	%
55	% [X,FVAL,EXITFLAG,OUTPUT] = LINPROG(f,A,b) returns a structure OUTPUT
56	% with the number of iterations taken in OUTPUT.iterations, maximum of
57	% constraint violations in OUTPUT.constrviolation, the type of
58	% algorithm used in OUTPUT.algorithm, the number of conjugate gradient
59	% iterations in OUTPUT.cgiterations (= 0, included for backward
60	% compatibility), and the exit message in OUTPUT.message.
61	%
62	% [X,FVAL,EXITFLAG,OUTPUT,LAMBDA] = LINPROG(f,A,b) returns the set of
63	% Lagrangian multipliers LAMBDA, at the solution: LAMBDA.ineqlin for the
64	% linear inequalities A, LAMBDA.eqlin for the linear equalities Aeq,
65	% LAMBDA.lower for LB, and LAMBDA.upper for UB.
66	%
67	% NOTE: the LargeScale (the default) version of LINPROG uses a
68	% primal-dual method. Both the primal problem and the dual problem
69	% must be feasible for convergence. Infeasibility messages of
70	% either the primal or dual, or both, are given as appropriate. The
71	% primal problem in standard form is
72	% min f'*x such that A*x = b, x >= 0.
73	% The dual problem is
74	% max b'*y such that A'*y + s = f, s >= 0.
75	%
76	% See also QUADPROG.
77	%
78	% Copyright 1990-2009 The MathWorks, Inc.
79	% \$Revision:2.2\$ \$Date: 2009/05/07 18:25:03\$
80	%
81	% If just 'defaults' passed in, return the default options in X
82	% Default MaxIter is set to [] because its value depends on the algorithm.
83	defaultopt = struct('Display','final',...
84	'TolFun',[],'Diagnostics','off',...
85	'LargeScale','on','MaxIter',[], ...
86	'Simplex','off');
87	
88	if nargin==1 && nargout <= 1 && isequal(f,'defaults')
89	x = defaultopt;
90	return
91	end

№ строки	Программа linprog_Matlab 2009b
92	
93	% Handle missing arguments
94	if nargin < 9, options = [];
95	if nargin < 8, x0 = [];
96	if nargin < 7, ub = [];
97	if nargin < 6, lb = [];
98	if nargin < 5, Beq = [];
99	if nargin < 4, Aeq = [];
100	end, end, end, end, end, end
101	
102	% Detect problem structure input
103	problemInput = false;
104	if nargin == 1
105	if isa(f,'struct')
106	problemInput = true;
107	[f,A,B,Aeq,Beq,lb,ub,x0,options] = separateOptimStruct(f);
108	else % Single input and non-structure.
109	error('optim:linprog:InputArg','The input to LINPROG should be either a structure with valid fields or consist of at least three arguments.');
110	end
111	end
112	
113	if nargin < 3 && ~problemInput
114	error('optim:linprog:NotEnoughInputs','LINPROG requires at least three input arguments.')
115	end
116	
117	% Check for non-double inputs
118	% SUPERIORFLOAT errors when superior input is neither single nor double;
119	% We use try-catch to override SUPERIORFLOAT's error message when input
120	% data type is integer.
121	try
122	dataType = superiorfloat(f,A,B,Aeq,Beq,lb,ub,x0);
123	catch ME
124	if strcmp(ME.identifier,'MATLAB:datatypes:superiorfloat')
125	dataType = 'notDouble';
126	end
127	end
128	
129	if ~strcmp(dataType,'double')
130	error('optim:linprog:NonDoubleInput', ...
131	'LINPROG only accepts inputs of data type double.')
132	end
133	
134	if nargout > 4
135	computeLambda = true;
136	else
137	computeLambda = false;
138	end

№ строки	Программа linprog_Matlab 2009b
139	<code>if nargout > 3</code>
140	<code> computeConstrViolation = true;</code>
141	<code>else</code>
142	<code> computeConstrViolation = false;</code>
143	<code>end</code>
144	
145	<code>% Options setup</code>
146	<code>if isfield(options,'Simplex')</code>
147	<code> useSimplex = isequal(optimget(options,'Simplex',defaultopt,'fast'), 'on');</code>
148	<code>else</code>
149	<code> useSimplex = isequal(defaultopt.Simplex, 'on');</code>
150	<code>end</code>
151	
152	<code>largescale = isequal(optimget(options,'LargeScale',defaultopt,'fast'),'on');</code>
153	<code>diagnostics = isequal(optimget(options,'Diagnostics',defaultopt,'fast'),'on');</code>
154	<code>switch optimget(options,'Display',defaultopt,'fast')</code>
155	<code>case {'off','none'}</code>
156	<code> verbosity = 0;</code>
157	<code>case {'iter','iter-detailed'}</code>
158	<code> verbosity = 2;</code>
159	<code>case {'final','final-detailed'}</code>
160	<code> verbosity = 1;</code>
161	<code>otherwise</code>
162	<code> verbosity = 1;</code>
163	<code>end</code>
164	<code>% Set the constraints up: defaults and check size</code>
165	<code>[nineqcstr,nvarsineq]=size(A);</code>
166	<code>[neqcstr, nvarseq]=size(Aeq);</code>
167	<code>nvars = max([length(f),nvarsineq,nvarseq]); % In case A is empty</code>
168	<code>ncstr = nineqcstr + neqcstr;</code>
169	
170	<code>if isempty(f), f=zeros(nvars,1); end</code>
171	<code>if isempty(A), A=zeros(0,nvars); end</code>
172	<code>if isempty(B), B=zeros(0,1); end</code>
173	<code>if isempty(Aeq), Aeq=zeros(0,nvars); end</code>
174	<code>if isempty(Beq), Beq=zeros(0,1); end</code>
175	
176	<code>% Set to column vectors</code>
177	<code>f = f(:);</code>
178	<code>B = B(:);</code>
179	<code>Beq = Beq(:);</code>
180	
181	<code>if ~isequal(length(B),nineqcstr)</code>
182	<code> error('optim:linprog:SizeMismatchRowsOfA', ...</code>
183	<code> 'The number of rows in A must be the same as the length of b.');</code>
184	<code>elseif ~isequal(length(Beq),neqcstr)</code>
185	

№ строки	Программа linprog_Matlab 2009b
186	error('optim:linprog:SizeMismatchRowsOfAeq', ...
187	'The number of rows in Aeq must be the same as the length of beq.');
188	elseif ~isequal(length(f),nvarsineq) && ~isempty(A)
189	error('optim:linprog:SizeMismatchColsOfA', ...
190	'The number of columns in A must be the same as the length of f.');
191	elseif ~isequal(length(f),nvarseq) && ~isempty(Aeq)
192	error('optim:linprog:SizeMismatchColsOfAeq', ...
193	'The number of columns in Aeq must be the same as the length of f.');
194	end
195	
196	[x0,lb,ub,msg] = checkbounds(x0,lb,ub,nvars);
197	if ~isempty(msg)
198	exitflag = -2;
199	x=x0; fval = []; lambda = [];
200	output.iterations = 0;
201	output.constrviolation = [];
202	output.algorithm = ''; % not known at this stage
203	output.cgiterations = [];
204	output.message = msg;
205	if verbosity > 0
206	disp(msg)
207	end
208	return
209	end
210	caller = 'linprog';
211	ncstr = nineqcstr + neqcstr;
212	
213	if largescale
214	OUTPUT.algorithm = 'large-scale: interior point';
215	elseif useSimplex
216	OUTPUT.algorithm = 'medium-scale: simplex';
217	else
218	OUTPUT.algorithm = 'medium-scale: active-set';
219	end
220	
221	if diagnostics
222	% Do diagnostics on information so far
223	gradflag = []; hessflag = []; line_search=[];
224	constflag = 0; gradconstflag = 0; non_eq=0;non_ineq=0;
225	lin_eq=size(Aeq,1); lin_ineq=size(A,1); XOUT=ones(nvars,1);
226	funfcn{1} = [];ff=[]; GRAD=[];HESS=[];
227	confcn{1}=[];c=[];ceq=[];cGRAD=[];ceqGRAD=[];
228	msg = diagnose('linprog',OUTPUT,gradflag,hessflag,constflag,gradconstflag,...
229	line_search,options,defaultopt,XOUT,non_eq,...
230	non_ineq,lin_eq,lin_ineq,lb,ub,funfcn,confcn,ff,GRAD,HESS,c,ceq,cGRAD,ceqGRAD);
231	end
232	

№ СТОКИ	Программа linprog_Matlab 2009b
233	<code>if (largescale)</code>
234	<code>if (useSimplex)</code>
235	<code>warning('optim:linprog:IgnoreSimplexOn', ...</code>
236	<code>['Simplex algorithm does not currently solve large-scale problems;\n' ...</code>
237	<code>'calling large-scale interior point algorithm. (To run simplex, set\n' ...</code>
238	<code>'OPTIONS.LargeScale to "off".'])</code>
239	<code>end</code>
240	<code>if ~isempty(x0) && verbosity > 0</code>
241	<code>warning('optim:linprog:IgnoreStartPoint', ...</code>
242	<code>['Large scale (interior point) algorithm uses a built-in starting point;\n' ...</code>
243	<code>'ignoring user-supplied X0.'])</code>
244	<code>end</code>
245	<code>% Set the default values of TolFun and MaxIter for this algorithm</code>
246	<code>defaultopt.TolFun = 1e-8;</code>
247	<code>defaultopt.MaxIter = 85;</code>
248	<code>[x,fval,lambda,exitflag,output] = lipsol(f,A,B,Aeq,Beq,lb,ub,options,defaultopt,computeLambda);</code>
249	<code>elseif (useSimplex)</code>
250	<code>if ~isempty(x0) && verbosity > 0</code>
251	<code>warning('optim:linprog:IgnoreStartPoint', ...</code>
252	<code>['Simplex algorithm uses a built-in starting point; ignoring user-supplied X0.'])</code>
253	<code>end</code>
254	<code>% Set the default values of TolFun and MaxIter for this algorithm</code>
255	<code>defaultopt.TolFun = 1e-6;</code>
256	<code>defaultopt.MaxIter = '100*NumberOfVariables';</code>
257	<code>[x,fval,lambda,exitflag,output] = simplex(f,A,B,Aeq,Beq,lb,ub,options,defaultopt,computeLambda);</code>
258	<code>% Remap exitflags if necessary</code>
259	<code>if exitflag == -1</code>
260	<code>exitflag = -2;</code>
261	<code>elseif exitflag == -2</code>
262	<code>exitflag = -3;</code>
263	<code>end</code>
264	<code>else</code>
265	<code>if ~largescale && (issparse(A) issparse(Aeq)) % asked for medium-scale but sparse</code>
266	<code>if verbosity > 0</code>
267	<code>disp('The medium-scale (active-set) algorithm does not currently handle sparse matrices.')</code>
268	<code>disp('Converting to full matrices to solve.')</code>
269	<code>end</code>
270	<code>end</code>
271	<code>if isempty(x0), x0=zeros(nvars,1); end</code>
272	<code>% Set the default value of MaxIter for this algorithm</code>
273	<code>defaultopt.MaxIter = '10*max(NumberOfVariables,NumberOfInequalities+NumberOfBounds)';</code>
274	<code>% Create qpsub options structure</code>
275	<code>lpmaxiter = optimget(options,'MaxIter',defaultopt,'fast');</code>
276	<code>if ischar(lpmaxiter)</code>
277	<code>if ise-</code>
278	<code>qual(lower(lpmaxiter),'10*max(numberofvariables,numberofinequalities+numberofbounds)')</code>
279	<code>lpmaxiter = 10*max(nvars,ncstr-neqcstr);</code>

№ строки	Программа linprog_Matlab 2009b
280	else
281	error('optim:linprog:InvalidMaxIter', ...
282	'Option "MaxIter" must be an integer value if not the default.')
283	end
284	end
285	lpoptions.MaxIter = lpmxiter;
286	% A fixed constraint tolerance (eps) is used for constraint
287	% satisfaction; no need to specify any value
288	lpoptions.TolCon = [];
289	
290	[x,lambdaqp,exitflag,output,dum1,dum2,msg]= ...
291	qpsub([],full(f),full([Aeq;A]),full([Beq;B]),lb,ub,x0,neqcstr,verbosity,caller,ncstr, ...
292	nvars,lpoptions);
293	output.algorithm = 'medium-scale: active-set';
294	end
295	
296	if isequal(OUTPUT.algorithm , 'medium-scale: active-set')
297	fval = f*x;
298	if computeLambda
299	llb = length(lb);
300	lub = length(ub);
301	lambda.lower = zeros(llb,1);
302	lambda.upper = zeros(lub,1);
303	arglb = ~isinf(lb); lenarglb = nnz(arglb);
304	argub = ~isinf(ub); lenargub = nnz(argub);
305	lambda.eqlin = lambdaqp(1:neqcstr,1);
306	lambda.ineqlin = lambdaqp(neqcstr+1:neqcstr+nineqcstr,1);
307	lambda.lower(arglb) = lambdaqp(neqcstr+nineqcstr+1:neqcstr+nineqcstr+lenarglb);
308	lambda.upper(argub) = lambdaqp(neqcstr+nineqcstr+lenarglb+1:neqcstr+nineqcstr+lenarglb+lenargub);
309	end
310	output.cgiterations = [];
311	
312	if exitflag == 1
313	normalTerminationMsg = sprintf('Optimization terminated.');
314	if verbosity > 0
315	disp(normalTerminationMsg)
316	end
317	if isempty(msg)
318	output.message = normalTerminationMsg;
319	else
320	% append normal termination msg to current output msg
321	output.message = sprintf('%s\n%s',msg,normalTerminationMsg);
322	end
323	else
324	output.message = msg;
325	end
326	else % large-scale & simplex algorithms
	% The constraint violation is always computed for medium-scale.

№ строки	Программа linprog_Matlab 2009b
327	% Compute constraint violation when x is not empty (lipsol/simplex presolve
328	% can return empty x).
329	if computeConstrViolation && ~isempty(x)
330	output.constrviolation = max([0; norm(Aeq*x-Beq, inf); (lb-x); (x-ub); (A*x-B)]);
331	else
332	output.constrviolation = [];
333	end
334	end
335	Пустая строка
336	
337	
338	

Приложение 3. Команды Матлаба R2009b, используемые в программах Монографии

1. INPUTDLG

Create and open input dialog box

Syntax

```
answer = inputdlg(prompt)
answer = inputdlg(prompt,dlg_title)
answer = inputdlg(prompt,dlg_title,num_lines)
answer = inputdlg(prompt,dlg_title,num_lines,defAns)
answer = inputdlg(prompt,dlg_title,num_lines,defAns,options)
```

Description

`answer = inputdlg(prompt)` creates a modal dialog box and returns user input for multiple prompts in the cell array. `prompt` is a cell array containing prompt strings.

Note A modal dialog box prevents the user from interacting with other windows before responding. For more information, see `WindowState` in the MATLAB [Figure Properties](#).

`answer = inputdlg(prompt,dlg_title)` `dlg_title` specifies a title for the dialog box.

`answer = inputdlg(prompt,dlg_title,num_lines)` `num_lines` specifies the number of lines for each user-entered value. `num_lines` can be a scalar, column vector, or matrix.

- If `num_lines` is a scalar, it applies to all prompts.
- If `num_lines` is a column vector, each element specifies the number of lines of input for a prompt.
- If `num_lines` is a matrix, it should be size `m-by-2`, where `m` is the number of prompts on the dialog box. Each row refers to a prompt. The first column specifies the number of lines of input for a prompt. The second column specifies the width of the field in characters.

`answer = inputdlg(prompt,dlg_title,num_lines,defAns)` `defAns` specifies the default value to display for each prompt. `defAns` must contain the same number of elements as `prompt` and all elements must be strings.

`answer = inputdlg(prompt,dlg_title,num_lines,defAns,options)` If `options` is the string `'on'`, the dialog is made resizable in the horizontal direction. If `options` is a structure, the fields shown in the following table are recognized:

Field	Description
Resize	Can be <code>'on'</code> or <code>'off'</code> (default). If <code>'on'</code> , the window is resizable horizontally.
WindowState	Can be either <code>'normal'</code> or <code>'modal'</code> (default).
Interpreter	Can be either <code>'none'</code> (default) or <code>'tex'</code> . If the value is <code>'tex'</code> , the prompt strings are rendered using LaTeX.

If the user clicks the Cancel button to close an `inputdlg` box, the dialog returns an empty cell array:

```
answer =
    {}
```

Remarks

`inputdlg` uses the [uiwait](#) function to suspend execution until the user responds.

The returned variable `answer` is a cell array containing strings, one string per text entry field, starting from the top of the dialog box.

To convert a member of the cell array to a number, use [str2num](#). To do this, you can add the following code to the end of any of the examples below:

```
[val status] = str2num(answer{1}); % Use curly bracket for subscript
if ~status
    % Handle empty value returned for unsuccessful conversion
    % ...
```

```
end
```

% `val` is a scalar or matrix converted from the first input

Users can enter scalar or vector values into `inputdlg` fields; `str2num` converts space- and comma-delimited strings into row vectors, and semicolon-delimited strings into column vectors. For exam-

ple, if `answer{1}` contains `'1 2 3;4 -5 6+7i'`, the conversion produces:

```
val = str2num(answer{1})
```

```
val =
```

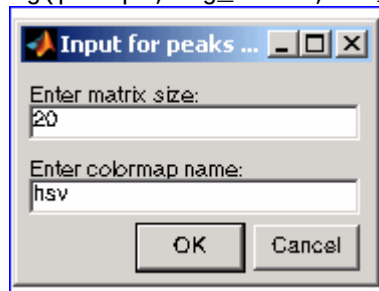
```
1.0000    2.0000    3.0000
4.0000   -5.0000    6.0000 + 7.0000i
```

Example

Example 1

Create a dialog box to input an integer and colormap name. Allow one line for each value.

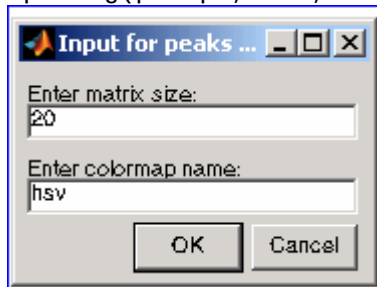
```
prompt = {'Enter matrix size:', 'Enter colormap name:'};
dlg_title = 'Input for peaks function';
num_lines = 1;
def = {'20', 'hsv'};
answer = inputdlg(prompt, dlg_title, num_lines, def);
```



Example 2

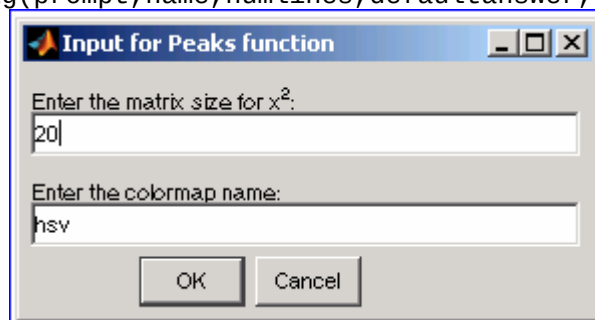
Create a dialog box using the default options. Then use the options to make it resizable and not modal, and to interpret the text using LaTeX.

```
prompt={'Enter the matrix size for x^2:', ...
        'Enter the colormap name:'};
name='Input for Peaks function';
numlines=1;
defaultanswer={'20', 'hsv'};
answer=inputdlg(prompt, name, numlines, defaultanswer);
```



```
options.Resize='on';
options.WindowStyle='normal';
options.Interpreter='tex';
```

```
answer=inputdlg(prompt, name, numlines, defaultanswer, options);
```



See Also

[dialog](#), [errordlg](#), [helpdlg](#), [listdlg](#), [msgbox](#), [questdlg](#), [warndlg](#)

[figure](#), [str2num](#), [uiwait](#), [uiresume](#)

[Predefined Dialog Boxes](#) for related functions

[Provide feedback about this page](#)

input

inputname

2. KEYBOARD

Input from keyboard

Syntax

keyboard

Description

keyboard, when placed in an M-file, stops execution of the file and gives control to the keyboard. The special status is indicated by a K appearing before the prompt. You can examine or change variables; all MATLAB commands are valid. This keyboard mode is useful for debugging your M-files..

To terminate the keyboard mode, type the command

return

then press the Return key.

See Also

[dbstop](#), [input](#), [quit](#), [pause](#), [return](#)

3. MENU

Generate menu of choices for user input

Syntax

```
choice = menu('mtitle','opt1','opt2',...,'optn')
```

```
choice = menu('mtitle',options)
```

displays the

Description

choice = menu('mtitle','opt1','opt2',...,'optn') menu whose title is in the string variable 'mtitle' and whose choices are string variables 'opt1', 'opt2', and so on. The menu opens in a modal dialog box. menu returns the number of the selected menu item, or 0 if the user clicks the close button on the window.

choice = menu('mtitle',options), where options is a 1-by-N cell array of strings containing the menu choices.

If the user's terminal provides a graphics capability, menu displays the menu items as push buttons in a figure window (Example 1). Otherwise, they will be given as a numbered list in the Command Window (Example 2).

Remarks

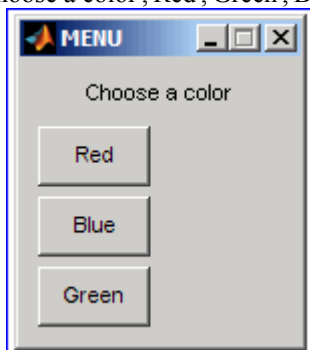
To call menu from a uicontrol or other ui object, set that object's Interruptible property to 'on'. For more information, see [Uicontrol Properties](#).

Examples

Example 1

On a system with a display, menu displays choices as buttons in a dialog box:

choice = menu('Choose a color','Red','Green','Blue') displays



The number entered by the user in response to the prompt is returned as choice (i.e., choice = 2 implies that the user selected Blue).

After input is accepted, the dialog box closes, returning the output in choice. You can use choice to control the color of a graph:

```
t = 0:1:60;
```

```
s = sin(t);
```

```
color = ['r','g','b']
```

```
plot(t,s,color(choice))
```

Example 2

On a system without a display, menu displays choices in the Command Window:

```
choice = menu('Choose a color','Red','Blue','Green')
```

displays the text

```
----- Choose a color -----
```

```
1) Red
```

```
2) Blue
```

```
3) Green
```

Select a menu number:

See Also

[guide](#), [input](#), [uicontrol](#), [uimenu](#)

4. ASSIGNIN

Assign value to variable in specified workspace

SYNTAX

```
assignin(ws, 'var', val)
```

DESCRIPTION

`assignin(ws, 'var', val)` assigns the value `val` to the variable `var` in the workspace `ws`. `var` is created if it doesn't exist. `ws` can have a value of 'base' or 'caller' to denote the MATLAB base workspace or the workspace of the caller function.

The `assignin` function is particularly useful for these tasks:

- Exporting data from a function to the MATLAB workspace
- Within a function, changing the value of a variable that is defined in the workspace of the caller function (such as a variable in the function argument list)

Remarks

The MATLAB base workspace is the workspace that is seen from the MATLAB command line (when not in the debugger). The caller workspace is the workspace of the function that called the M-file. Note that the base and caller workspaces are equivalent in the context of an M-file that is invoked from the MATLAB command line.

Examples

This example creates a dialog box for the image display function, prompting a user for an image name and a colormap name. The `assignin` function is used to export the user-entered values to the MATLAB workspace variables `imfile` and `cmap`.

```
prompt = {'Enter image name:','Enter colormap name:'};
```

```
title = 'Image display - assignin example';
```

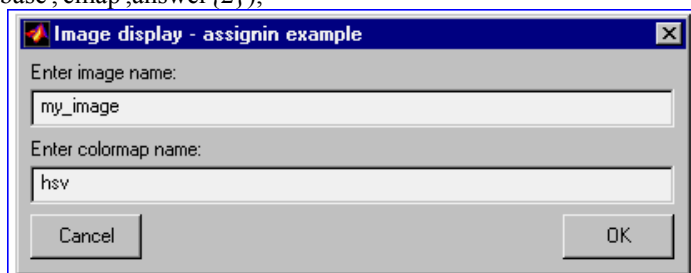
```
lines = 1;
```

```
def = {'my_image','hsv'};
```

```
answer = inputdlg(prompt,title,lines,def);
```

```
assignin('base','imfile',answer{1});
```

```
assignin('base','cmap',answer{2});
```



See Also

[evalin](#)

5. NUM2STR

Convert number to string

Syntax

```
str = num2str(A)
```

```
str = num2str(A, precision)
```

```
str = num2str(A, format)
```

Description

The `num2str` function converts numbers to their string representations. This function is useful for labeling and titling plots with numeric values.

`str = num2str(A)` converts array `A` into a string representation `str` with roughly four digits of precision and an exponent if required.

`str = num2str(A, precision)` converts the array `A` into a string representation `str` with maximum precision specified by *precision*. Argument *precision* specifies the number of digits the output string is to contain. The default is four.

`str = num2str(A, format)` converts array `A` using the supplied *format*. (See [fprintf](#) for format string details.) By default, `num2str` displays floating point values using '%11.4g' format (four significant digits in exponential or fixed-point notation, whichever is shorter).

If the input array is integer-valued, `num2str` returns the exact string representation of that integer. The term integer-valued includes large floating-point numbers that lose precision due to limitations of the hardware.

`num2str` removes any leading spaces from the output string. Thus, `num2str(42.67, '%10.2f')` returns a 1-by-5 character array '42.67'.

Examples

`num2str(pi)` is 3.142.

`num2str(eps)` is 2.22e-16.

`num2str(randn(2,2),3)` produces the following string matrix:

```
num2str(randn(2,2),3)
```

```
ans =
```

```
0.538   -2.26
```

```
1.83    0.862
```

`num2str` with a format of '%10.5e\n' returns a matrix of strings in exponential format, having 5 decimal places, with each element separated by a newline character:

```
x = rand(2,3) * 9999;      % Create a 2-by-3 matrix.
```

```
A = num2str(x, '%10.5e\n') % Convert to string array.
```

```
A =
```

```
6.87255e+003
```

```
1.55597e+003
```

```
8.55890e+003
```

```
3.46077e+003
```

```
1.91097e+003
```

```
4.90201e+003
```

See Also

[mat2str](#), [int2str](#), [str2num](#), [sprintf](#), [fprintf](#)

6. NOW

Current date and time

Syntax

```
t = now
```

Description

`t = now` returns the current date and time as a serial date number. To return the time only, use `rem(now,1)`. To return the date only, use `floor(now)`.

Examples

```
t1 = now, t2 = rem(now,1)
```

```
t1 =
```

```
7.2908e+05
```

```
t2 =
```

```
0.4013
```

See Also

[clock](#), [date](#), [datenum](#)

7. DISP

Display text or array

Syntax

```
disp(X)
```

Description

`disp(X)` displays an array, without printing the array name. If `X` contains a text string, the string is displayed.

Another way to display an array on the screen is to type its name, but this prints a leading "X=", which is not always desirable.

Note that `disp` does not display empty arrays.

Examples

Example 1 — Display a matrix with column labels

One use of `disp` in an M-file is to display a matrix with column labels:

```
disp('    Corn    Oats    Hay')
```

```
disp(rand(5,3))
```

which results in

Corn	Oats	Hay
0.2113	0.8474	0.2749
0.0820	0.4524	0.8807
0.7599	0.8075	0.6538
0.0087	0.4832	0.4899
0.8096	0.6135	0.7741

Example 2 — Display a hyperlink in the Command Window

You also can use the `disp` command to display a hyperlink in the Command Window. Include the full hypertext string on a single line as input to `disp`:

```
disp('<a href = "http://www.mathworks.com">The MathWorks Web Site</a>')
```

which generates this hyperlink in the Command Window:

[The MathWorks Web Site](http://www.mathworks.com)

Click the link to display The MathWorks home page in a MATLAB Web browser.

Example 3 — Display multiple items on the same line

Use concatenation to display multiple items using `disp`. For example:

```
x = [1 2 3];
```

```
disp(['The values of x are: ', num2str(x)]);
```

displays

The values of x are: 1 2 3

If you want to display text without a trailing newline character, use `fprintf`. For example,

```
fprintf('%s %d %d %d ', 'The values of x are:', x(:));
```

displays text similar to the above, but does not include a newline.

See Also

[format](#), [int2str](#), [matlabcolon](#), [num2str](#), [rats](#), [sprintf](#), [fprintf](#)

8. NUM2STR

Convert number to string

Syntax

```
str = num2str(A)
```

```
str = num2str(A, precision)
```

```
str = num2str(A, format)
```

Description

The `num2str` function converts numbers to their string representations. This function is useful for labeling and titling plots with numeric values.

`str = num2str(A)` converts array `A` into a string representation `str` with roughly four digits of precision and an exponent if required.

`str = num2str(A, precision)` converts the array `A` into a string representation `str` with maximum precision specified by `precision`. Argument `precision` specifies the number of digits the output string is to contain. The default is four.

`str = num2str(A, format)` converts array `A` using the supplied `format`. (See `fprintf` for format string details.) By default, `num2str` displays floating point values using `'%11.4g'` format (four significant digits in exponential or fixed-point notation, whichever is shorter).

If the input array is integer-valued, `num2str` returns the exact string representation of that integer. The term integer-valued includes large floating-point numbers that lose precision due to limitations of the hardware.

`num2str` removes any leading spaces from the output string. Thus, `num2str(42.67, '%10.2f')` returns

a 1-by-5 character array '42.67'.

Examples

`num2str(pi)` is 3.142.

`num2str(eps)` is 2.22e-16.

`num2str(randn(2,2),3)` produces the following string matrix:

`num2str(randn(2,2),3)`

`ans =`

0.538 -2.26

1.83 0.862

`num2str` with a format of `%10.5e\n` returns a matrix of strings in exponential format, having 5 decimal places, with each element separated by a `newline` character:

`x = rand(2,3) * 9999;` % Create a 2-by-3 matrix.

`A = num2str(x, '%10.5e\n')` % Convert to string array.

`A =`

6.87255e+003

1.55597e+003

8.55890e+003

3.46077e+003

1.91097e+003

4.90201e+003

See Also

[mat2str](#), [int2str](#), [str2num](#), [sprintf](#), [fprintf](#)

9. STR2NUM

Convert string to number

Syntax

`x = str2num('str')`

`[x, status] = str2num('str')`

Description

Note `str2num` uses the [eval](#) function to convert the input argument. Side effects can occur if the string contains calls to functions. Using [str2double](#) can avoid some of these side effects.

`x = str2num('str')` converts the string `str`, which is an ASCII character representation of a numeric value, to numeric representation. `str2num` also converts string matrices to numeric matrices. If the input string does not represent a valid number or matrix, `str2num(str)` returns the empty matrix in `x`.

The input string can contain one or more numbers separated by spaces, commas, or semicolons, such as '5', '10,11,12', or '5,10;15,20'. In addition to numerical values and delimiters, the input string can also include a decimal point, leading + or - signs, the letter e or d preceding a power of 10 scale factor, or the letter i or j indicating a complex or imaginary number.

The following table shows several examples of valid inputs to `str2num`:

String Input	Numeric Output	Output Class
'500'	500	1-by-1 scalar double
'500 250 125 67'	500, 250, 125, 67	1-by-4 row vector of double
'500; 250; 125; 62.5'	500.0000 250.0000 125.0000 62.5000	4-by-1 column vector of double
'1 23 6 21; 53:56'	1 23 6 21 53 54 55 56	2-by-5 matrix of double
'12e-3 5.9e-3'	0.0120 0.0059	vector of double
'uint16(500)'	500	16-bit unsigned integer

If the input string does not represent a valid number or matrix, `str2num(str)` returns the empty matrix in `x`.

[x, status] = str2num('str') returns the status of the conversion in logical status, where status equals logical 1 (true) if the conversion succeeds, and logical 0 (false) otherwise.

Space characters can be significant. For instance, str2num('1+2i') and str2num('1 + 2i') produce x = 1+2i, while str2num('1 +2i') produces x = [1 2i]. You can avoid these problems by using the [str2double](#) function.

Examples

Input a character string that contains a single number. The output is a scalar double:

```
A = str2num('500')
A =
    500
```

```
class(A)
ans =
    double
```

Repeat this operation, but this time using an unsigned 16-bit integer:

```
A = str2num('uint16(500)')
A =
    500
```

```
class(A)
ans =
    uint16
```

Try three different ways of specifying a row vector. Each returns the same answer:

```
str2num('2 4 6 8')      % Separate with spaces.
ans =
     2     4     6     8
```

```
str2num('2,4,6,8')      % Separate with commas.
ans =
     2     4     6     8
```

```
str2num('[2 4 6 8]')    % Enclose in brackets.
ans =
     2     4     6     8
```

Note that the first two of these commands do not need the MATLAB square bracket operator to create a matrix. The str2num function inserts the brackets for you if they are needed.

Use a column vector this time:

```
str2num('2; 4; 6; 8')
ans =
     2
     4
     6
     8
```

And now a 2-by-2 matrix:

```
str2num('2 4; 6 8')
ans =
     2     4
     6     8
See Also
```

[num2str](#), [str2double](#), [hex2num](#), [sscanf](#), [sparse](#), [char](#), [special characters](#)

10. CLOCK

Current time as date vector

Syntax

```
c = clock
```

Description

c = clock returns a 6-element date vector containing the current date and time in decimal form:

```
[year month day hour minute seconds]
```

The sixth element of the date vector output (seconds) is accurate to several digits beyond the decimal point. The

statement `fix(clock)` rounds to integer display format.

Remarks

When timing the duration of an event, use the `tic` and `toc` functions instead of `clock` or `etime`. These latter two functions are based on the system time which can be adjusted periodically by the operating system and thus might not be reliable in time comparison operations.

See Also

[cputime](#), [datenum](#), [datevec](#), [now](#), [etime](#), [tic](#), [toc](#)

Пример

```
>> clock
```

```
ans =
```

```
1.0e+003 *
```

```
2.0160 0.0030 0.0290 0.0200 0.0570 0.0259
```

```
>> fix(clock)
```

```
ans =
```

```
year month day hour minute s econds
2016 3 29 20 58 25
```

```
>> ans(4)
```

```
ans =
```

```
20
```

```
ans(4:6)
```

```
ans =
```

```
21 12 52
```

11. DATENUM

Convert date and time to serial date number

Syntax

`N = datenum(V)`

`N = datenum(S, F)`

`N = datenum(S, F, P)`

`N = datenum([S, P, F])`

`N = datenum(Y, M, D)`

`N = datenum(Y, M, D, H, MN, S)`

`N = datenum(S)`

`N = datenum(S, P)`

Description

`datenum` is one of three conversion functions that enable you to express dates and times in any of three formats in your MATLAB application: a string (or *date string*), a vector of date and time components (or *date vector*), or as a numeric offset from a known date in time (or *serial date number*). Here is an example of a date and time expressed in the three MATLAB formats:

Date String: '24-Oct-2003 12:45:07'

Date Vector: [2003 10 24 12 45 07]

Serial Date Number: 7.3188e+005

A serial date number represents the whole and fractional number of days from a specific date and time, where `datenum('Jan-1-0000 00:00:00')` returns the number 1. (The year 0000 is merely a reference point and is not intended to be interpreted as a real year in time.)

Values outside the normal range of each unit are automatically carried to the next. For example, month values greater than 12 are carried to years. All units can wrap and have negative values, with the following caveats:

- Month values less than 1 are set to 1.
- Day values, D, less than 1 are set to the last day of the previous month minus |D|.

`N = datenum(V)` converts one or more date vectors `V` to serial date numbers `N`. Input `V` can be an `m`-by-6 or `m`-by-3 matrix containing `m` full or partial date vectors respectively. A full date vector has six elements, specifying year, month, day, hour, minute, and second, in that order. A partial date vector has three elements, specifying year, month, and day, in that order. Each element of `V` must be a positive double-precision number. `datenum` returns a column vector of `m` date numbers, where `m` is the total number of date vectors in `V`.

`N = datenum(S, F)` converts one or more date strings `S` to serial date numbers `N` using format string `F` to interpret

each date string. Input *S* can be a one-dimensional character array or cell array of date strings. All date strings in *S* must have the same format, and that format must match one of the date string formats shown in the help for the [datestr](#) function. `datenum` returns a column vector of *m* date numbers, where *m* is the total number of date strings in *S*. MATLAB considers date string years that are specified with only two characters (e.g., '79') to fall within 100 years of the current year.

See the [datestr](#) reference page to find valid string values for *F*. These values are listed in Table 1 in the column labeled "Dateform String." You can use any string from that column except for those that include the letter *Q* in the string (for example, 'QQ-YYYY'). Certain formats may not contain enough information to compute a date number. In these cases, hours, minutes, seconds, and milliseconds default to 0, the month defaults to January, the day to 1, and the year to the current year.

`N = datenum(S, F, P)` converts one or more date strings *S* to date numbers *N* using format *F* and pivot year *P*. The pivot year is used in interpreting date strings that have the year specified as two characters. It is the starting year of the 100-year range in which a two-character date string year resides. The default pivot year is the current year minus 50 years.

`N = datenum([S, P, F])` is the same as the syntax shown above, except the order of the last two arguments are switched.

`N = datenum(Y, M, D)` returns the serial date numbers for corresponding elements of the *Y*, *M*, and *D* (year, month, day) arrays. *Y*, *M*, and *D* must be arrays of the same size (or any can be a scalar) of type double. You can also specify the input arguments as a date vector, [*Y M D*].

`N = datenum(Y, M, D, H, MN, S)` returns the serial date numbers for corresponding elements of the *Y*, *M*, *D*, *H*, *MN*, and *S* (year, month, day, hour, minute, and second) array values. `datenum` does not accept milliseconds in a separate input, but as a fractional part of the seconds (*S*) input. Inputs *Y*, *M*, *D*, *H*, *MN*, and *S* must be arrays of the same size (or any can be a scalar) of type double. You can also specify the input arguments as a date vector, [*Y M D H MN S*].

`N = datenum(S)` converts date string *S* into a serial date number. String *S* must be in one of the date formats 0, 1, 2, 6, 13, 14, 15, 16, or 23, as defined in the reference page for the [datestr](#) function. MATLAB considers date string years that are specified with only two characters (e.g., '79') to fall within 100 years of the current year. If the format of date string *S* is known, use the syntax `N = datenum(S, F)`.

`N = datenum(S, P)` converts date string *S*, using pivot year *P*. If the format of date string *S* is known, use the syntax `N = datenum(S, F, P)`.

Note The last two calling syntaxes are provided for backward compatibility and are significantly slower than the syntaxes that include a format argument *F*.

Examples

Convert a date string to a serial date number:

```
n = datenum('19-May-2001', 'dd-mmm-yyyy')
```

```
n =  
730990
```

Specifying year, month, and day, convert a date to a serial date number:

```
n = datenum(2001, 12, 19)
```

```
n =  
731204
```

Convert a date vector to a serial date number:

```
format bank  
datenum('March 28, 2005 3:37:07.033 PM')  
ans =  
732399.65
```

Convert a date string to a serial date number using the default pivot year:

```
n = datenum('12-jun-17', 'dd-mmm-yy')
```

```
n =  
736858
```

Convert the same date string to a serial date number using 1400 as the pivot year:

```
n = datenum('12-jun-17', 'dd-mmm-yy', 1400)  
n =  
517712
```

Specify format 'dd.mm.yyyy' to be used in interpreting a nonstandard date string:

```
n = datenum('19.05.2000', 'dd.mm.yyyy')
```

```
n =  
730625
```

See Also

[datestr](#), [datevec](#), [date](#), [clock](#), [now](#), [datetick](#)

12. DATESTR

Convert date and time to string format

Syntax

`S = datestr(V)`

`S = datestr(N)`

`S = datestr(D, F)`

`S = datestr(S1, F, P)`

`S = datestr(..., 'local')`

Description

`datestr` is one of three conversion functions that enable you to express dates and times in any of three formats in your MATLAB application: a string (or *date string*), a vector of date and time components (or *date vector*), or as a numeric offset from a known date in time (or *serial date number*). Here is an example of a date and time expressed in the three MATLAB formats:

Date String: '24-Oct-2003 12:45:07'

Date Vector: [2003 10 24 12 45 07]

Serial Date Number: 7.3188e+005

A serial date number represents the whole and fractional number of days from 1-Jan-0000 to a specific date. The year 0000 is merely a reference point and is not intended to be interpreted as a real year in time.

Values outside the normal range of each unit are automatically carried to the next. For example, month values greater than 12 are carried to years. All units can wrap and have negative values, with the following caveats:

- Month values less than 1 are set to 1.
- Day values, D, less than 1 are set to the last day of the previous month minus |D|.

`S = datestr(V)` converts one or more date vectors `V` to date strings `S`. Input `V` must be an `m`-by-6 matrix containing `m` full (six-element) date vectors. Each element of `V` must be a positive double-precision number. `datestr` returns a column vector of `m` date strings, where `m` is the total number of date vectors in `V`.

`S = datestr(N)` converts one or more serial date numbers `N` to date strings `S`. Input argument `N` can be a scalar, vector, or multidimensional array of positive double-precision numbers. `datestr` returns a column vector of `m` date strings, where `m` is the total number of date numbers in `N`.

`S = datestr(D, F)` converts one or more date vectors, serial date numbers, or date strings `D` into the same number of date strings `S`. Input argument `F` is a format number or string that determines the format of the date string output. Valid values for `F` are given in the table [Standard MATLAB Date Format Definitions](#), below. Input `F` may also contain a free-form date format string consisting of format tokens shown in the table [Free-Form Date Format Specifiers](#), below.

Date strings with 2-character years are interpreted to be within the 100 years centered around the current year.

`S = datestr(S1, F, P)` converts date string `S1` to date string `S`, applying format `F` to the output string, and using pivot year `P` as the starting year of the 100-year range in which a two-character year resides. The default pivot year is the current year minus 50 years. All date strings in `S1` must have the same format.

`S = datestr(..., 'local')` returns the date string in the localized format that you currently have selected by means of your computer's operating system. If you leave `local` out of the argument list, `datestr` returns the date string in the default format, which is US English.

The `local` argument must come last in the argument sequence.

Note The vectorized calling syntax can offer significant performance improvement for large arrays.

Standard MATLAB Date Format Definitions

dateform (number)	dateform (string)	Example
0	'dd-mmm-yyyy HH:MM:SS'	01-Mar-2000 15:45:17
1	'dd-mmm-yyyy'	01-Mar-2000
2	'mm/dd/yy'	03/01/00
3	'mmm'	Mar
4	'm'	M
5	'mm'	03
6	'mm/dd'	03/01
7	'dd'	01
8	'ddd'	Wed

dateform (number)	dateform (string)	Example
9	'd'	W
10	'yyyy'	2000
11	'yy'	00
12	'mmyy'	Mar00
13	'HH:MM:SS'	15:45:17
14	'HH:MM:SS PM'	3:45:17 PM
15	'HH:MM'	15:45
16	'HH:MM PM'	3:45 PM
17	'QQ-YY'	Q1-01
18	'QQ'	Q1
19	'dd/mm'	01/03
20	'dd/mm/yy'	01/03/00
21	'mmm.dd,yyyy HH:MM:SS'	Mar.01,2000 15:45:17
22	'mmm.dd,yyyy'	Mar.01,2000
23	'mm/dd/yyyy'	03/01/2000
24	'dd/mm/yyyy'	01/03/2000
25	'yy/mm/dd'	00/03/01
26	'yyyy/mm/dd'	2000/03/01
27	'QQ-YYYY'	Q1-2001
28	'mmyyyy'	Mar2000
29 (ISO 8601)	'yyyy-mm-dd'	2000-03-01
30 (ISO 8601)	'yyyymmddTHHMMSS'	20000301T154517
31	'yyyy-mm-dd HH:MM:SS'	2000-03-01 15:45:17

Note dateform numbers 0, 1, 2, 6, 13, 14, 15, 16, and 23 produce a string suitable for input to `datenum` or `datevec`. Other date string formats do not work with these functions unless you specify a date form in the function call.

Note For date formats that specify only a time (i.e., dateform numbers 13, 14, 15, and 16), MATLAB sets the date to January 1 of the current year.

Time formats like 'h:m:s', 'h:m:s.s', 'h:m pm', ... can also be part of the input array `S`. If you do not specify a format string `F`, or if you specify `F` as -1, the date string format defaults to the following:

1 If `S` contains date information only, e.g., 01-Mar-1995

16 If `S` contains time information only, e.g., 03:45 PM

0 If `S` is a date vector, or a string that contains both date and time information, e.g., 01-Mar-1995 03:45

The following table shows the string symbols to use in specifying a free-form format for the output date string. MATLAB interprets these symbols according to your computer's language setting and the current MATLAB language setting.

Note You cannot use more than one format specifier for any date or time field. For example, `datestr(n, 'dddd dd mmmm')` specifies two formats for the day of the week, and thus returns an error.

Free-Form Date Format Specifiers

Symbol	Interpretation	Example
yyyy	Show year in full.	1990, 2002
yy	Show year in two digits.	90, 02

Symbol	Interpretation	Example
mmmm	Show month using full name.	March, December
mmm	Show month using first three letters.	Mar, Dec
mm	Show month in two digits.	03, 12
m	Show month using capitalized first letter.	M, D
dddd	Show day using full name.	Monday, Tuesday
ddd	Show day using first three letters.	Mon, Tue
dd	Show day in two digits.	05, 20
d	Show day using capitalized first letter.	M, T
HH	Show hour in two digits (no leading zeros when free-form specifier AM or PM is used (see last entry in this table)).	05, 5 AM
MM	Show minute in two digits.	12, 02
SS	Show second in two digits.	07, 59
FFF	Show millisecond in three digits.	.057
AM or PM	Append AM or PM to date string (see note below).	3:45:02 PM

Note Free-form specifiers AM and PM from the table above are identical. They do not influence which characters are displayed following the time (AM versus PM), but only whether or not they are displayed. MATLAB selects AM or PM based on the time entered.

Remarks

A vector of three or six numbers could represent either a single date vector, or a vector of individual serial date numbers. For example, the vector [2000 12 15 11 45 03] could represent either 11:45:03 on December 15, 2000 or a vector of date numbers 2000, 12, 15, etc.. MATLAB uses the following general rule in interpreting vectors associated with dates:

A 3- or 6-element vector having a first element within an approximate range of 500 greater than or less than the current year is considered by MATLAB to be a date vector. Otherwise, it is considered to be a vector of serial date numbers.

To specify dates outside of this range as a date vector, first convert the vector to a serial date number using the [datenum](#) function as shown here:

```
datestr(datenum([1400 12 15 11 45 03]), ...
    'mmm.dd,yyyy HH:MM:SS')
```

```
ans =
```

```
Dec.15,1400 11:45:03
```

To convert a nonstandard date form into a MATLAB date form, first convert the nonstandard date form to a date number, using either [datenum](#) or [datevec](#).

Examples

Convert date vector *v* to a date string:

```
v = [2009, 4, 2, 11, 7, 18];
```

```
datestr(v)
```

```
ans =
```

```
02-Apr-2009 11:07:18
```

Return the current date and time in a string using the default format, 0:

```
datestr(now)
```

```
ans =
```

```
28-Mar-2005 15:36:23
```

Format the current date in the mm/dd/yy format. Note that you can specify this format either by number or by string.

```
datestr(now, 2) -or- datestr(now, 'mm/dd/yy')
```

```
ans =
```

03/28/05

This example uses several of the free-form format specifiers. Note the difference between the number of free-form specifiers (e.g., 'dd', 'ddd', 'dddd') and the output:

```
str = 'Sept 13, 1986';
```

```
[datestr(str, 'ddd ') datestr(str, 'mmm dd, "yy")]
```

```
ans =
```

```
Sat Sep 13, '86
```

```
[datestr(str, 'dddd ') datestr(str, 'mmmm dd, yyyy')]
```

```
ans =
```

```
Saturday September 13, 1986
```

Reformat the date and time, and also show milliseconds:

```
dt = datestr(now, 'mmmm dd, yyyy HH:MM:SS.FFF AM')
```

```
dt =
```

```
March 28, 2005 3:37:07.952 PM
```

Change the pivot year and note the effect on the output:

```
datestr('4/16/55', 1, 1900)
```

```
ans =
```

```
16-Apr-1955
```

```
datestr('4/16/55', 1, 2000)
```

```
ans =
```

```
16-Apr-2055
```

The date below uses a nonstandard date form (month=13). Call [datenum](#) inside of datestr to get the correct return value:

```
datestr(datenum('13/24/88', 'mm/dd/yy'))
```

```
ans =
```

```
24-Jan-1989
```

See Also

[datenum](#), [datevec](#), [date](#), [clock](#), [now](#), [datetick](#)

13. Operators

On this page...

[Arithmetic Operators](#)

[Relational Operators](#)

[Logical Operators](#)

[Operator Precedence](#)

Arithmetic Operators

Arithmetic operators perform numeric computations, for example, adding two numbers or raising the elements of an array to a given power. The following table provides a summary. For more information, see the [arithmetic operators](#) reference page.

Operator	Description
+	Addition
-	Subtraction
. *	Multiplication
. /	Right division
. \	Left division
+	Unary plus
-	Unary minus
:	Colon operator
. ^	Power
. '	Transpose

Operator	Description
'	Complex conjugate transpose
*	Matrix multiplication
/	Matrix right division
\	Matrix left division
^	Matrix power

Arithmetic Operators and Arrays

Except for some matrix operators, MATLAB arithmetic operators work on corresponding elements of arrays with equal dimensions. For vectors and rectangular arrays, both operands must be the same size unless one is a scalar. If one operand is a scalar and the other is not, MATLAB applies the scalar to every element of the other operand—this property is known as *scalar expansion*.

This example uses scalar expansion to compute the product of a scalar operand and a matrix.

```
A = magic(3)
```

```
A =
```

```

8     1     6
3     5     7
4     9     2
```

```
3 * A
```

```
ans =
```

```

24     3    18
9     15    21
12    27     6
```

[Back to Top](#)

Relational Operators

Relational operators compare operands quantitatively, using operators like "less than" and "not equal to." The following table provides a summary. For more information, see the [relational operators](#) reference page.

Operator	Description
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
==	Equal to
~=	Not equal to

Relational Operators and Arrays

The MATLAB relational operators compare corresponding elements of arrays with equal dimensions. Relational operators always operate element-by-element. In this example, the resulting matrix shows where an element of A is equal to the corresponding element of B.

```
A = [2 7 6;9 0 5;3 0.5 6];
```

```
B = [8 7 0;3 2 5;4 -1 7];
```

```
A == B
```

```
ans =
```

```

0     1     0
0     0     1
0     0     0
```

For vectors and rectangular arrays, both operands must be the same size unless one is a scalar. For the case where one operand is a scalar and the other is not, MATLAB tests the scalar against every element of the other operand. Locations where the specified relation is true receive logical 1. Locations where the relation is false receive logical 0.

Relational Operators and Empty Arrays

The relational operators work with arrays for which any dimension has size zero, as long as both arrays are the same size or one is a scalar. However, expressions such as

```
A == []
```

return an error if A is not 0-by-0 or 1-by-1. This behavior is consistent with that of all other binary operators, such as +, -, >, <, &, |, etc.

To test for empty arrays, use the function `isempty(A)`

[Back to Top](#)

Logical Operators

MATLAB offers three types of logical operators and functions:

- [Element-wise](#) — operate on corresponding elements of logical arrays.
- [Bit-wise](#) — operate on corresponding bits of integer values or arrays.
- [Short-circuit](#) — operate on scalar, logical expressions.

The values returned by MATLAB logical operators and functions, with the exception of bit-wise functions, are of type `logical` and are suitable for use with logical indexing.

Element-Wise Operators and Functions

The following logical operators and functions perform elementwise logical operations on their inputs to produce a like-sized output array.

The examples shown in the following table use vector inputs A and B, where

```
A = [0 1 1 0 1];
```

```
B = [1 1 0 0 1];
```

Operator	Description	Example
&	Returns 1 for every element location that is true (non-zero) in both arrays, and 0 for all other elements.	A & B = 01001
	Returns 1 for every element location that is true (non-zero) in either one or the other, or both arrays, and 0 for all other elements.	A B = 11101
~	Complements each element of the input array, A.	~A = 10010
xor	Returns 1 for every element location that is true (non-zero) in only one array, and 0 for all other elements.	xor(A,B) = 10100

For operators and functions that take two array operands, (&, |, and xor), both arrays must have equal dimensions, with each dimension being the same size. The one exception to this is where one operand is a scalar and the other is not. In this case, MATLAB tests the scalar against every element of the other operand.

Note MATLAB converts any finite nonzero, numeric values used as inputs to logical expressions to logical 1, or true.

Operator Overloading. You can overload the &, |, and ~ operators to make their behavior dependent upon the class on which they are being used. Each of these operators has a representative function that is called whenever that operator is used. These are shown in the table below.

Logical Operation	Equivalent Function
A & B	and(A, B)

Logical Operation	Equivalent Function
$A \mid B$	<code>or(A, B)</code>
$\sim A$	<code>not(A)</code>

Other Array Functions.

Two other MATLAB functions that operate logically on arrays, but not in an elementwise fashion, are `any` and `all`. These functions show whether *any* or *all* elements of a vector, or a vector within a matrix or an array, are nonzero.

When used on a matrix, `any` and `all` operate on the columns of the matrix. When used on an N-dimensional array, they operate on the first nonsingleton dimension of the array. Or, you can specify an additional dimension input to operate on a specific dimension of the array.

The examples shown in the following table use array input A, where

```
A = [0 1 2;
     0 -3 8;
     0 5 0];
```

Function	Description	Example
<code>any(A)</code>	Returns 1 for a vector where <i>any</i> element of the vector is true (nonzero), and 0 if no elements are true.	<code>any(A) ans =</code> 0 1 1
<code>all(A)</code>	Returns 1 for a vector where <i>all</i> elements of the vector are true (nonzero), and 0 if all elements are not true.	<code>all(A) ans =</code> 0 1 0

Note The `all` and `any` functions ignore any NaN values in the input arrays.

Short-Circuiting in Elementwise Operators.

When used in the context of an `if` or `while` expression, and only in this context, the elementwise `|` and `&` operators use short-circuiting in evaluating their expressions. That is, `A|B` and `A&B` ignore the second operand, B, if the first operand, A, is sufficient to determine the result.

So, although the statement `1|[]` evaluates to false, the same statement evaluates to true when used in either an `if` or `while` expression:

```
A = 1; B = [];
if(A|B) disp('The statement is true', end;
The statement is true
```

while the reverse logical expression, which does not short-circuit, evaluates to false

```
if(B|A) disp('The statement is true', end;
```

Another example of short-circuiting with elementwise operators shows that a logical expression such as the following, which under most circumstances is invalid due to a size mismatch between A and B,

```
A = [1 1]; B = [2 0 1];
A|B % This generates an error.
```

works within the context of an `if` or `while` expression:

```
if (A|B) disp('The statement is true', end;
The statement is true
```

Logical Expressions Using the find Function.

The `find` function determines the indices of array elements that meet a given logical condition. The function is useful for creating masks and index matrices. In its most general form, `find` returns a single vector of indices. This vector can be used to index into arrays of any size or shape.

For example,

```
A = magic(4)
A =
    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
```

```
i = find(A > 8);
A(i) = 100
```

```
A =
    100     2     3    100
     5    100    100     8
    100     7     6    100
     4    100    100     1
```

Note An alternative to using `find` in this context is to index into the matrix using the logical expression itself. See the example below.

The last two statements of the previous example can be replaced with this one statement:

```
A(A > 8) = 100;
```

You can also use `find` to obtain both the row and column indices of a rectangular matrix for the array values that meet the logical condition:

```
A = magic(4)
```

```
A =
    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
```

```
[row, col] = find(A > 12)
```

```
row =
```

```
1
```

```
4
```

```
4
```

```
1
```

```
col =
```

```
1
```

```
2
```

```
3
```

```
4
```

Bit-Wise Functions

The following functions perform bit-wise logical operations on nonnegative integer inputs. Inputs may be scalar or in arrays. If in arrays, these functions produce a like-sized output array.

The examples shown in the following table use scalar inputs A and B, where

```
A = 28; % binary 11100
```

```
B = 21; % binary 10101
```

Function	Description	Example
bitand	Returns the bit-wise AND of two nonnegative integer arguments.	<code>bitand(A,B) = 20</code> (binary 10100)
bitor	Returns the bit-wise OR of two nonnegative integer arguments.	<code>bitor(A,B) = 29</code> (binary 11101)
bitcmp	Returns the bit-wise complement as an n-bit number, where n is the second input argument to <code>bitcmp</code> .	<code>bitcmp(A,5) = 3</code> (binary 00011)
bitxor	Returns the bit-wise exclusive OR of two nonnegative integer arguments.	<code>bitxor(A,B) = 9</code> (binary 01001)

Short-Circuit Operators

The following operators perform AND and OR operations on logical expressions containing scalar values. They are *short-circuit* operators in that they evaluate their second operand only when the result is not fully determined by the first operand.

Operator	Description
<code>&&</code>	Returns logical 1 (true) if both inputs evaluate to true, and logical 0 (false) if they do not.

Operator	Description
	Returns logical 1 (true) if either input, or both, evaluate to true, and logical 0 (false) if they do not.

The statement shown here performs an AND of two logical terms, A and B:

A && B

If A equals zero, then the entire expression will evaluate to logical 0 (false), regardless of the value of B. Under these circumstances, there is no need to evaluate B because the result is already known. In this case, MATLAB short-circuits the statement by evaluating only the first term.

A similar case is when you OR two terms and the first term is true. Again, regardless of the value of B, the statement will evaluate to true. There is no need to evaluate the second term, and MATLAB does not do so.

Advantage of Short-Circuiting. You can use the short-circuit operators to evaluate an expression only when certain conditions are satisfied. For example, you want to execute an M-file function only if the M-file resides on the current MATLAB path.

Short-circuiting keeps the following code from generating an error when the file, `myfun.m`, cannot be found:

```
comp = (exist('myfun.m') == 2) && (myfun(x) >= y)
```

Similarly, this statement avoids divide-by-zero errors when b equals zero:

```
x = (b ~= 0) && (a/b > 18.5)
```

You can also use the && and || operators in [if](#) and [while](#) statements to take advantage of their short-circuiting behavior:

```
if (nargin >= 3) && (ischar(varargin{3}))
```

[Back to Top](#)

Приоритет операторов

(Operator Precedence)

You can build expressions that use any combination of arithmetic, relational, and logical operators. Precedence levels determine the order in which MATLAB evaluates an expression. Within each precedence level, operators have equal precedence and are evaluated from left to right. The precedence rules for MATLAB operators are shown in this list, ordered from highest precedence level to lowest precedence level:

- Parentheses ()
- Transpose (.'), power (.^), complex conjugate transpose ('), matrix power (^)
- Unary plus (+), unary minus (-), logical negation (~)
- Multiplication (.*), right division (./), left division (.\), matrix multiplication (*), matrix right division (/), matrix left division (\)
- Addition (+), subtraction (-)
- Colon operator (:)
- Less than (<), less than or equal to (<=), greater than (>), greater than or equal to (>=), equal to (==), not equal to (~=)
- Element-wise AND (&)
- Element-wise OR (|)
- Short-circuit AND (&&)
- Short-circuit OR (||)

Precedence of AND and OR Operators

MATLAB always gives the & operator precedence over the | operator. Although MATLAB typically evaluates expressions from left to right, the expression `a|b&c` is evaluated as `a|(b&c)`. It is a good idea to use parentheses to explicitly specify the intended precedence of statements containing combinations of & and |.

The same precedence rule holds true for the && and || operators.

Overriding Default Precedence

The default precedence can be overridden using parentheses, as shown in this example:

```
A = [3 9 5];
B = [2 1 5];
C = A./B.^2
C =
    0.7500    9.0000    0.2000

C = (A./B).^2
C =
    2.2500   81.0000    1.0000
```

14. isempty

Determine whether array is empty

Syntax

```
TF = isempty(A)
```

Description

TF = isempty(A) returns logical 1 (true) if A is an empty array and logical 0 (false) otherwise. An empty array has at least one dimension of size zero, for example, 0-by-0 or 0-by-5.

Examples

```
B = rand(2,2,2);
```

```
B(:, :, :) = [];
```

```
isempty(B)
```

```
ans = 1
```

See Also

[is*](#)

15. char

Convert to character array (string)

SYNTAX

```
S = char(X)
```

```
S = char(C)
```

```
S = char(t1, t2, t3, ...)
```

Description

S = char(X) converts the array X that contains nonnegative integers representing character codes into a MATLAB character array. The actual characters displayed depend on the character encoding scheme for a given font. The result for any elements of X outside the range from 0 to 65535 is not defined (and can vary from platform to platform). Use double to convert a character array into its numeric codes.

S = char(C), when C is a cell array of strings, places each element of C into the rows of the character array S. Use cellstr to convert back.

S = char(t1, t2, t3, ...) forms the character array S containing the text strings T1, T2, T3, ... as rows, automatically padding each string with blanks to form a valid matrix. Each text parameter, T_i, can itself be a character array. This allows the creation of arbitrarily large character arrays. Empty strings are significant.

Examples

To print a 3-by-32 display of the printable ASCII characters,

```
ascii = char(reshape(32:127, 32, 3))
```

```
ascii =
```

```
!"#$%&'()*+,-./0123456789:;<=>?
```

```
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_
```

```
'abcdefghijklmnopqrstuvwxyz{|}~
```

See Also

[ischar](#), [isletter](#), [isspace](#), [isstrprop](#), [cellstr](#), [iscellstr](#), [get](#), [set](#), [strings](#), [strvcat](#), [text](#)

16. Other Data Structures

On this page...

[Multidimensional Arrays](#)

[Cell Arrays](#)

Characters and Text Structures

Multidimensional Arrays

Multidimensional arrays in the MATLAB environment are arrays with more than two subscripts. One way of creating a multidimensional array is by calling [zeros](#), [ones](#), [rand](#), or [randn](#) with more than two arguments. For example,

```
R = randn(3,4,5);
```

creates a 3-by-4-by-5 array with a total of $3 \times 4 \times 5 = 60$ normally distributed random elements.

A three-dimensional array might represent three-dimensional physical data, say the temperature in a room, sampled on a rectangular grid. Or it might represent a sequence of matrices, $A^{(k)}$, or samples of a time-dependent matrix, $A(t)$. In these latter cases, the (i, j) th element of the k th matrix, or the t_k th matrix, is denoted by $A(i, j, k)$.

MATLAB and Dürer's versions of the magic square of order 4 differ by an interchange of two columns. Many different magic squares can be generated by interchanging columns. The statement

```
p = perms(1:4);
```

generates the $4! = 24$ permutations of $1:4$. The k th permutation is the row vector $p(k, :)$. Then

```
A = magic(4);
```

```
M = zeros(4,4,24);
```

```
for k = 1:24
```

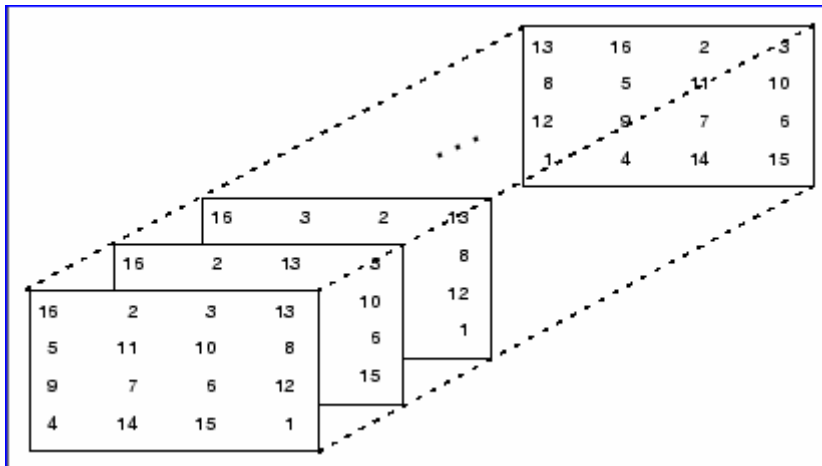
```
    M(:,:,k) = A(:,p(k,:));
```

```
end
```

stores the sequence of 24 magic squares in a three-dimensional array, M . The size of M is `size(M)`

```
ans =
```

```
4    4    24
```



Note The order of the matrices shown in this illustration might differ from your results. The `perms` function always returns all permutations of the input vector, but the order of the permutations might be different for different MATLAB versions.

The statement

```
sum(M,d)
```

computes sums by varying the d th subscript. So

```
sum(M,1)
```

is a 1-by-4-by-24 array containing 24 copies of the row vector

```
34    34    34
```

and

```
sum(M,2)
```

is a 4-by-1-by-24 array containing 24 copies of the column vector

```
34
```

```
34
```

```
34
```

```
34
```

Finally,

```
S = sum(M,3)
```

adds the 24 matrices in the sequence. The result has size 4-by-4-by-1, so it looks like a 4-by-4 array:

```
S =
```

```
204    204    204    204
```

```
204 204 204 204
204 204 204 204
204 204 204 204
```

Cell Arrays

Cell arrays in MATLAB are multidimensional arrays whose elements are copies of other arrays. A cell array of empty matrices can be created with the [cell](#) function. But, more often, cell arrays are created by enclosing a miscellaneous collection of things in curly braces, `{}`. The curly braces are also used with subscripts to access the contents of various cells. For example,

```
C = {A sum(A) prod(prod(A))}
```

produces a 1-by-3 cell array. The three cells contain the magic square, the row vector of column sums, and the product of all its elements. When `C` is displayed, you see

```
C =
[4x4 double] [1x4 double] [20922789888000]
```

This is because the first two cells are too large to print in this limited space, but the third cell contains only a single number, 16!, so there is room to print it.

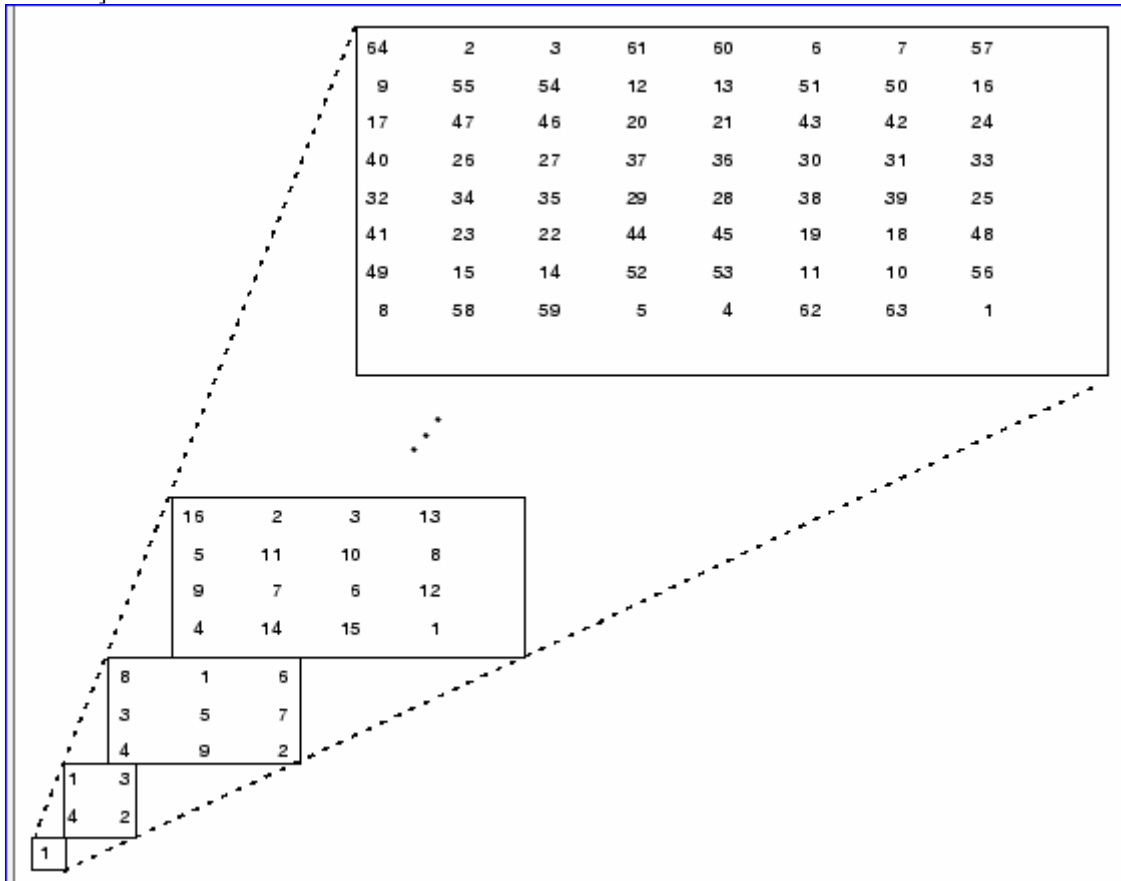
Here are two important points to remember. First, to retrieve the contents of one of the cells, use subscripts in curly braces. For example, `C{1}` retrieves the magic square and `C{3}` is 16!. Second, cell arrays contain *copies* of other arrays, not *pointers* to those arrays. If you subsequently change `A`, nothing happens to `C`.

You can use three-dimensional arrays to store a sequence of matrices of the *same* size. Cell arrays can be used to store a sequence of matrices of *different* sizes. For example,

```
M = cell(8,1);
for n = 1:8
    M{n} = magic(n);
end
M
```

produces a sequence of magic squares of different order:

```
M =
[      1]
[ 2x2 double]
[ 3x3 double]
[ 4x4 double]
[ 5x5 double]
[ 6x6 double]
[ 7x7 double]
[ 8x8 double]
```



You can retrieve the 4-by-4 magic square matrix with `M{4}`

Characters and Text

Enter text into MATLAB using single quotes. For example,

```
s = 'Hello'
```

The result is not the same kind of numeric matrix or array you have been dealing with up to now. It is a 1-by-5 character array.

Internally, the characters are stored as numbers, but not in floating-point format. The statement

```
a = double(s)
```

converts the character array to a numeric matrix containing floating-point representations of the ASCII codes for each character. The result is

```
a =
```

```
72 101 108 108 111
```

The statement

```
s = char(a)
```

reverses the conversion.

Converting numbers to characters makes it possible to investigate the various fonts available on your computer.

The printable characters in the basic ASCII character set are represented by the integers `32:127`. (The integers less than 32 represent nonprintable control characters.) These integers are arranged in an appropriate 6-by-16 array with `F = reshape(32:127,16,6);`

The printable characters in the extended ASCII character set are represented by `F+128`. When these integers are interpreted as characters, the result depends on the font currently being used. Type the statements

```
char(F)
```

```
char(F+128)
```

and then vary the font being used for the Command Window. Select **Preferences** from the **File** menu and then click **Fonts** to change the font. If you include tabs in lines of code, use a fixed-width font, such as Monospaced, to align the tab positions on different lines.

Concatenation with square brackets joins text variables together into larger strings. The statement

```
h = [s, ' world']
```

joins the strings horizontally and produces

```
h =
```

```
Hello world
```

The statement

```
v = [s; 'world']
```

joins the strings vertically and produces

```
v =
```

```
Hello
```

```
world
```

Note that a blank has to be inserted before the `'w'` in `h` and that both words in `v` have to have the same length. The resulting arrays are both character arrays; `h` is 1-by-11 and `v` is 2-by-5.

To manipulate a body of text containing lines of different lengths, you have two choices—a padded character array or a cell array of strings. When creating a character array, you must make each row of the array the same length. (Pad the ends of the shorter rows with spaces.) The [char](#) function does this padding for you. For example,

```
S = char('A','rolling','stone','gathers','momentum.')
```

produces a 5-by-9 character array:

```
S =
```

```
A
```

```
rolling
```

```
stone
```

```
gathers
```

```
momentum.
```

Alternatively, you can store the text in a cell array. For example,

```
C = {'A','rolling','stone','gathers','momentum.'}
```

creates a 5-by-1 cell array that requires no padding because each row of the array can have a different length:

```
C =
```

```
'A'
```

```
'rolling'
```

```
'stone'
```

```
'gathers'
```

```
'momentum.'
```

You can convert a padded character array to a cell array of strings with

```
C = cellstr(S)
```

and reverse the process with

```
S = char(C)
```

Structures

Structures are multidimensional MATLAB arrays with elements accessed by textual *field designators*. For example,

```
S.name = 'Ed Plum';
S.score = 83;
S.grade = 'B+';
```

creates a scalar structure with three fields:

```
S =
    name: 'Ed Plum'
   score: 83
  grade: 'B+'
```

Like everything else in the MATLAB environment, structures are arrays, so you can insert additional elements.

In this case, each element of the array is a structure with several fields. The fields can be added one at a time,

```
S(2).name = 'Toni Miller';
S(2).score = 91;
S(2).grade = 'A-';
```

or an entire element can be added with a single statement:

```
S(3) = struct('name','Jerry Garcia',...
             'score',70,'grade','C')
```

Now the structure is large enough that only a summary is printed:

```
S =
1x3 struct array with fields:
    name
   score
  grade
```

There are several ways to reassemble the various fields into other MATLAB arrays. They are mostly based on the notation of a *comma-separated list*. If you type

```
S.score
it is the same as typing
S(1).score, S(2).score, S(3).score
which is a comma-separated list.
```

If you enclose the expression that generates such a list within square brackets, MATLAB stores each item from the list in an array. In this example, MATLAB creates a numeric row vector containing the `score` field of each element of structure array `S`:

```
scores = [S.score]
scores =
    83    91    70
```

```
avg_score = sum(scores)/length(scores)
avg_score =
    81.3333
```

To create a character array from one of the text fields (`name`, for example), call the [char](#) function on the comma-separated list produced by `S.name`:

```
names = char(S.name)
names =
    Ed Plum
    Toni Miller
    Jerry Garcia
```

Similarly, you can create a cell array from the `name` fields by enclosing the list-generating expression within curly braces:

```
names = {S.name}
names =
    'Ed Plum'    'Toni Miller'    'Jerry Garcia'
```

To assign the fields of each element of a structure array to separate variables outside of the structure, specify each output to the left of the equals sign, enclosing them all within square brackets:

```
[N1 N2 N3] = S.name
N1 =
    Ed Plum
N2 =
    Toni Miller
N3 =
    Jerry Garcia
```

Dynamic Field Names

The most common way to access the data in a structure is by specifying the name of the field that you want to reference. Another means of accessing structure data is to use dynamic field names. These names express the field as a

variable expression that MATLAB evaluates at run-time. The dot-parentheses syntax shown here makes `expression` a dynamic field name:

```
structName.(expression)
```

Index into this field using the standard MATLAB indexing syntax. For example, to evaluate `expression` into a field name and obtain the values of that field at columns 1 through 25 of row 7, use

```
structName.(expression)(7,1:25)
```

Dynamic Field Names Example. The `avgscore` function shown below computes an average test score, retrieving information from the `testscores` structure using dynamic field names:

```
function avg = avgscore(testscores, student, first, last)
```

```
for k = first:last
```

```
    scores(k) = testscores.(student).week(k);
```

```
end
```

```
avg = sum(scores)/(last - first + 1);
```

You can run this function using different values for the dynamic field `student`. First, initialize the structure that contains scores for a 25-week period:

```
testscores.Ann_Lane.week(1:25) = ...
```

```
[95 89 76 82 79 92 94 92 89 81 75 93 ...
```

```
85 84 83 86 85 90 82 82 84 79 96 88 98];
```

```
testscores.William_King.week(1:25) = ...
```

```
[87 80 91 84 99 87 93 87 97 87 82 89 ...
```

```
86 82 90 98 75 79 92 84 90 93 84 78 81];
```

Now run `avgscore`, supplying the students name fields for the `testscores` structure at runtime using dynamic field names:

```
avgscore(testscores, 'Ann_Lane', 7, 22)
```

```
ans =
```

```
85.2500
```

```
avgscore(testscores, 'William_King', 7, 22)
```

```
ans =
```

```
87.7500
```

17. cell

Construct cell array

Syntax

```
c = cell(n)
```

```
c = cell(m, n)
```

```
c = cell([m, n])
```

```
c = cell(m, n, p, ...)
```

```
c = cell([m n p ...])
```

```
c = cell(size(A))
```

```
c = cell(javaobj)
```

Description

`c = cell(n)` creates an `n`-by-`n` cell array of empty matrices. An error message appears if `n` is not a scalar.

`c = cell(m, n)` or `c = cell([m, n])` creates an `m`-by-`n` cell array of empty matrices. Arguments `m` and `n` must be scalars.

`c = cell(m, n, p, ...)` or `c = cell([m n p ...])` creates an `m`-by-`n`-by-`p`-... cell array of empty matrices. Arguments `m`, `n`, `p`, ... must be scalars.

`c = cell(size(A))` creates a cell array the same size as `A` containing all empty matrices.

`c = cell(javaobj)` converts a Java array or Java object `javaobj` into a MATLAB cell array. Elements of the resulting cell array will be of the MATLAB type (if any) closest to the Java array elements or Java object.

Remarks

This type of cell is not related to ["cell mode"](#), a MATLAB feature used in debugging and publishing.

Examples

This example creates a cell array that is the same size as another array, `A`.

```
A = ones(2,2)
```

```
A =
```

```
1 1
```

```
1 1
```

```
c = cell(size(A))
```

```
c =
```

```
 [] []  
 [] []
```

The next example converts an array of `java.lang.String` objects into a MATLAB cell array.

```
strArray = java_array('java.lang.String', 3);  
strArray(1) = java.lang.String('one');  
strArray(2) = java.lang.String('two');  
strArray(3) = java.lang.String('three');
```

```
cellArray = cell(strArray)
```

```
cellArray =
```

```
'one'  
'two'  
'three'
```

See Also

[num2cell](#), [ones](#), [rand](#), [randn](#), [zeros](#)

18. cat

Concatenate arrays along specified dimension

Syntax

```
C = cat(dim, A, B)  
C = cat(dim, A1, A2, A3, A4, ...)
```

Description

`C = cat(dim, A, B)` concatenates the arrays `A` and `B` along `dim`.

`C = cat(dim, A1, A2, A3, A4, ...)` concatenates all the input arrays (`A1`, `A2`, `A3`, `A4`, and so on)

along `dim`.

`cat(2, A, B)` is the same as `[A, B]`, and `cat(1, A, B)` is the same as `[A; B]`.

Remarks

When used with comma-separated list syntax, `cat(dim, C{:})` or `cat(dim, C.field)` is a convenient way to concatenate a cell or structure array containing numeric matrices into a single matrix.

Examples

Given

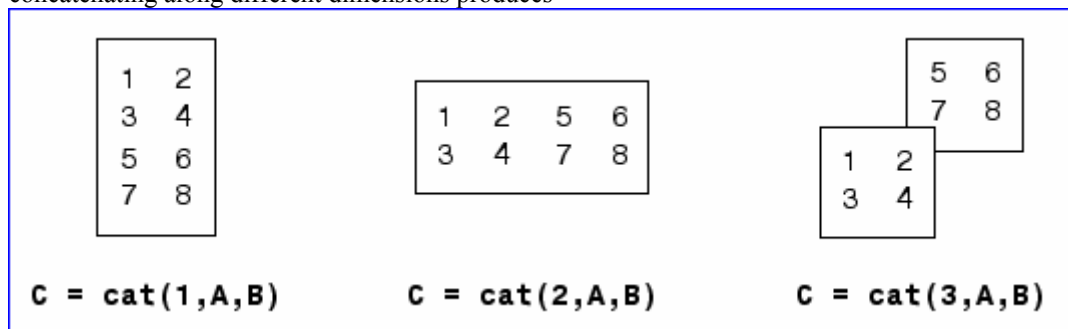
```
A =
```

```
1 2 5 6  
3 4 7 8
```

```
B =
```

```
1 2 5 6  
3 4 7 8
```

concatenating along different dimensions produces



The commands

```
A = magic(3); B = pascal(3);
```

```
C = cat(4, A, B);
```

produce a 3-by-3-by-1-by-2 array.

See Also

[vertcat](#), [horzcat](#), [strcat](#), [strvcat](#), [num2cell](#), [special character](#) []

19. if

Execute statements if condition is true

Syntax

```
if expression, statements, end
```

Description

`if expression, statements, end` evaluates `expression` and, if the evaluation yields logical 1 (true) or a nonzero result, executes one or more MATLAB commands denoted here as `statements`.

`expression` is a MATLAB expression, usually consisting of variables or smaller expressions joined by rela-

tional operators (e.g., `count < limit`), or logical functions (e.g., `isreal(A)`). Simple expressions can be combined by logical operators (`&&`, `|`, `~`) into compound expressions such as the following. MATLAB evaluates compound expressions from left to right, adhering to [operator precedence](#) rules.

```
(count < limit) && ((height - offset) >= 0)
```

Nested `if` statements must each be paired with a matching `end`.

The `if` function can be used alone or with the [elseif](#) and [elseif](#) functions. When using [elseif](#) and/or [else](#) within an `if` statement, the general form of the statement is

```
if expression1
    statements1
elseif expression2
    statements2
else
    statements3
end
```

See [Program Control Statements](#) in the MATLAB Programming Fundamentals documentation for more information on controlling the flow of your program code.

Remarks

Nonscalar Expressions

If the evaluated `expression` yields a nonscalar value, then every element of this value must be true or nonzero for the entire expression to be considered true. For example, the statement `if (A < B)` is true only if each element of matrix `A` is less than its corresponding element in matrix `B`. See [Example 2](#), below.

Partial Evaluation of the expression Argument

Within the context of an [if](#) or [while](#) expression, MATLAB does not necessarily evaluate all parts of a logical expression. In some cases it is possible, and often advantageous, to determine whether an expression is true or false through only partial evaluation.

For example, if `A` equals zero in statement 1 below, then the expression evaluates to `false`, regardless of the value of `B`. In this case, there is no need to evaluate `B` and MATLAB does not do so. In statement 2, if `A` is nonzero, then the expression is true, regardless of `B`. Again, MATLAB does not evaluate the latter part of the expression.

```
1) if (A && B)          2) if (A || B)
```

You can use this property to your advantage to cause MATLAB to evaluate a part of an expression only if a preceding part evaluates to the desired state. Here are some examples.

```
while (b ~= 0) && (a/b > 18.5)
if exist('myfun.m') && (myfun(x) >= y)
if iscell(A) && all(cellfun('isreal', A))
```

Empty Arrays

In most cases, using `if` on an empty array treats the array as `false`. There are some conditions however under which `if` evaluates as `true` on an empty array. Two examples of this, where `A` is equal to `[]`, are

```
if all(A), do_something, end
if 1|A, do_something, end
```

The latter expression is `true` because of short-circuiting, which causes MATLAB to ignore the right side operand of an OR statement whenever the left side evaluates to `true`.

Short-Circuiting Behavior

When used in the context of an [if](#) or [while](#) expression, and only in this context, the element-wise `|` and `&` operators use short-circuiting in evaluating their expressions. That is, `A|B` and `A&B` ignore the second operand, `B`, if the first operand, `A`, is sufficient to determine the result.

See [Short-Circuiting in Elementwise Operators](#) for more information on this.

Examples

Example 1 - Simple if Statement

In this example, if both of the conditions are satisfied, then the student passes the course.

```
if ((attendance >= 0.90) && (grade_average >= 60))
    pass = 1;
end;
```

Example 2 - Nonscalar Expression

Given matrices `A` and `B`,

```
A =      B =
 1   0    1   1
 2   3    3   4
```

Expression	Evaluates As	Because
<code>A < B</code>	<code>false</code>	<code>A(1,1)</code> is not less than <code>B(1,1)</code> .

Expression	Evaluates As	Because
1) $A < (B + 1)$	true	Every element of A is less than that same element of B with 1 added.
$A \& B$	false	A(1,2) is false, and B is ignored due to short-circuiting.
$B < 5$	true	Every element of B is less than 5.

See Also

[else](#), [elseif](#), [end](#), [for](#), [while](#), [switch](#), [break](#), [return](#), [relational operators](#), logical operators ([elementwise](#) and [short-circuit](#)),

20. report

Generate reports from report templates

Syntax

```
report
report (filename)
[report1, report2, ...] = report (rptfile1, rptfile2, ...)
report (...,-oOPATH,...)
report (...,-fFORMAT, ...)
```

Description

- 1) `report` with no arguments opens the Report Explorer. For more information on the Report Explorer, see [What Is the Report Explorer?](#)
- 2) `report (filename)` generates a report from the specified report template. When specifying the name of the report template, omit the `.rpt` file name extension. You can specify one or more report templates.
- 3) `[report1, report2, ...] = report (rptfile1, rptfile2, ...)` returns the names of the generated reports. If the MATLAB Report Generator software cannot generate a given report, its returned name is empty.

The `report` command also accepts the following flags, which set options for the report during the report-generation process. You can pass these flags to the `report` command anywhere among the input arguments.

1. `report (...,-oOPATH,...)` sets the name of the generated report. You can specify a path or a single file name for the `OPATH` path argument.
2. `report (...,-fFORMAT, ...)` sets the output format and file name extension of the generated report. Supported formats include:

- a. Adobe Acrobat PDF (`.pdf`)
- b. HTML (`.html`)
- c. Microsoft Word (`.doc`)
- d. Rich Text format (`.rtf`)

Examples

Example 1: Setting the format of the generated report

1. Generate the report `testrpt` in PDF format:
`report testrpt -fpdf`
2. Generate the report `testrpt` in RTF format:
`report testrpt -frtf`
3. Generate the report `testrpt` in Microsoft Word format:
`report testrpt -fdoc`

Note Only Microsoft Windows platforms support this option.

4. Generate a multipage HTML report from the `figloop-tutorial` report template:
`report figloop-tutorial -fhtml -shtml-!MultiPage`

Example 2: Specifying the file and path of the generated report

Generate a report named `simple-report` in the folder `/tmp/index.html`:

```
report ('simple-report','-o/tmp/index.html')
```

See Also

[Generating Reports](#)

21. blanks

Create string of blank characters

Syntax

```
blanks(n)
```

Description

`blanks(n)` is a string of `n` blanks.

Examples

`blanks` is useful with the `display` function. For example,

```
disp(['xxx' blanks(20) 'yyy'])
```

displays twenty blanks between the strings 'xxx' and 'yyy'.

`disp(blanks(n))` moves the cursor down `n` lines.

See Also

[clc](#), [format](#), [home](#)

22. `cellstr`

Create cell array of strings from character array

Syntax

```
c = cellstr(S)
```

Description

`c = cellstr(S)` places each row of the character array `S` into separate cells of `C`. Any trailing spaces in the rows of `S` are removed.

Use the [char](#) function to convert back to a string matrix.

Examples

Given the string matrix

```
S = ['abc'; 'defg'; 'hi ']
```

```
S =
```

```
abc
defg
hi
```

```
whos S
```

Name	Size	Bytes	Class
S	3x4	24	char array

The following command returns a 3-by-1 cell array.

```
c = cellstr(S)
```

```
c =
```

```
'abc'
'defg'
'hi'
```

```
whos c
```

Name	Size	Bytes	Class
c	3x1	294	cell array

See Also

[iscellstr](#), [strings](#), [char](#), [isstrprop](#)

23. `fprintf`

Write data to text file

Syntax

```
fprintf(fileID, format, A, ...)
```

```
fprintf(format, A, ...)
```

```
count = fprintf(...)
```

Description

`fprintf(fileID, format, A, ...)` applies the *format* to array *A* and any additional array arguments in column order, and writes the data to a text file. `fprintf` uses the encoding scheme specified in the call to [fopen](#).

`fprintf(format, A, ...)` formats data and displays the results on the screen.

`count = fprintf(...)` returns the number of bytes that `fprintf` writes.

Inputs

fileID

One of the following:

An integer file identifier obtained from [fopen](#).

1 for standard output (the screen).

2 for standard error.

Default: 1 (the screen)

format

String in single quotation marks that describes the format of the output fields. Can include combinations of the following:

Percent sign followed by a conversion character, such as '%s' for strings.

Operators that describe field width, precision, and other options.

Literal text to print.

Escape characters, including:

' '	Single quotation mark
%%	Percent character
\\	Backslash
\b	Backspace
\f	Form feed
\n	New line
\r	Carriage return
\t	Horizontal tab
\x <i>N</i>	Hexadecimal number, <i>N</i>
\o <i>N</i>	Octal number, <i>N</i>

Conversion characters and optional operators appear in the following order (includes spaces for clarity):

The following table lists the available conversion characters and subtypes.

Value Type	Conversion	Details
Integer, signed	%d or %i	Base 10 values
	%ld or %li	64-bit base 10 values
Integer, unsigned	%u	Base 10
	%o	Base 8 (octal)
	%x	Base 16 (hexadecimal), lowercase letters a–f
	%X	Same as %x, uppercase letters A–F
	%lu %lo %lx or %lX	64-bit values, base 10, 8, or 16
Floating-point number	%f	Fixed-point notation
	%e	Exponential notation, such as 3.141593e+00
	%E	Same as %e, but uppercase, such as 3.141593E+00
	%g	The more compact of %e or %f, with no trailing zeros
	%G	The more compact of %E or %f, with no trailing zeros

	%bX or %bX %bo %bu	Double-precision hexadecimal, octal, or decimal value Example: %bX prints pi as 400921fb54442d18
	%tX or %tX %to %tu	Single-precision hexadecimal, octal, or decimal value Example: %tX prints pi as 40490fdb
Characters	%C	Single character
	%S	String of characters

Additional operators include:

Field width

Minimum number of characters to print. Can be a number, or an asterisk (*) to refer to an argument in the input list. For example, the input list ('%12d', intmax) is equivalent to ('%*d', 12, intmax).

Precision

For %f, %e, or %E: Number of digits to the right of the decimal point.

Example: '%6.4f' prints pi as '3.1416'

For %g or %G: Number of significant digits.

Example: '%6.4g' prints pi as '3.142'

Can be a number, or an asterisk (*) to refer to an argument in the input list. For example, the input list ('%6.4f', pi) is equivalent to ('%*.4f', 6, 4, pi).

Flags

Action	Flag	Example
Left-justify.	'-'	%-5.2f
Print sign character (+ or -).	'+'	%+5.2f
Insert a space before the value.	' '	% 5.2f
Pad with zeros.	'0'	%05.2f

Identifier

Order for processing inputs. Use the syntax *n*\$, where *n* represents the position of the value in the input list.

For example, '%3\$s %2\$s %1\$s %2\$s' prints inputs 'A', 'B', 'C' as follows: C B A B.

The following limitations apply to conversions:

Numeric conversions print only the real component of complex numbers.

If you apply an integer or string conversion to a numeric value that contains a fraction, MATLAB overrides the specified conversion, and uses %e.

If you apply a string conversion (%S) to integer values, MATLAB:

Issues a warning.

Converts values that correspond to valid character codes to characters.

For example, '%S' converts [65 66 67] to ABC.

Different platforms display exponential notation (such as %e) with a different number of digits in the exponent.

Platform	Example
Windows	1.23e+004
UNIX	1.23e+04

Different platforms display negative zero (-0) differently.

Platform	Conversion Character		
	%e or %E	%f	%g or %G
Windows	0.000000e+000	0.000000	0

Others	-0.000000e+00	-0.000000	-0
--------	---------------	-----------	----

A Numeric or character array.

Examples

Print multiple values and literal text to the screen:

```
B = [8.8 7.7 ; ...
     8800 7700];
fprintf('X is %4.2f meters or %8.3f mm\n', 9.9, 9900, B)
```

MATLAB displays:

```
X is 9.90 meters or 9900.000 mm
X is 8.80 meters or 8800.000 mm
X is 7.70 meters or 7700.000 mm
```

Explicitly convert double-precision values with fractions to integer values, and print to the screen:

```
a = [1.02 3.04 5.06];
fprintf('%d\n', round(a));
```

Write a short table of the exponential function to a text file called `exp.txt`:

```
x = 0:1:1;
y = [x; exp(x)];

% open the file with write permission
fid = fopen('exp.txt', 'w');
fprintf(fid, '%6.2f %12.8f\n', y);
fclose(fid);
```

```
% view the contents of the file
type exp.txt
```

MATLAB import functions, all UNIX applications, and Microsoft Word and WordPad recognize `'\n'` as a newline indicator. However, if you plan to read the file with Microsoft Notepad, use `'\r\n'` to move to a new line when writing. For example, replace the previous call to `fprintf` with the following:

```
fprintf(fid, '%6.2f %12.8f\r\n', y);
```

On a Windows system, convert PC-style exponential notation (three digits in the exponent) to UNIX-style notation (two digits), and print data to a file:

```
a = [0.06 0.1 5 300]
```

```
% use sprintf to convert the numeric data to text, using %e
a_str = sprintf('%e\t', a);
```

```
% use strrep to replace exponent prefix with shorter version
a_str = strrep(a_str, 'e+0', 'e+');
a_str = strrep(a_str, 'e-0', 'e-');
```

```
% call fprintf to print the updated text strings
fid = fopen('newfile.txt', 'w');
fprintf(fid, '%s', a_str);
fclose(fid);
```

```
% view the contents of the file
type newfile.txt
```

Display a hyperlink ([The MathWorks Web Site](http://www.mathworks.com)) on the screen:

```
site = 'http://www.mathworks.com';
title = 'The MathWorks Web Site';
```

```
fprintf('<a href = "%s">%s</a>\n', site, title)
```

References

[1] Kernighan, B. W., and D. M. Ritchie, *The C Programming Language*, Second Edition, Prentice-Hall, Inc.,

1988.

[2] ANSI specification X3.159-1989: "Programming Language C," ANSI, 1430 Broadway, New York, NY 10018.

See Also

[disp](#) | [fclose](#) | [ferror](#) | [fopen](#) | [fread](#) | [fscanf](#) | [fwrite](#) | [sprintf](#)

How To

[Formatting Strings](#)

[Appending or Overwriting Existing Files](#)

24. num2cell

Convert numeric array to cell array

Syntax

`C = num2cell(A)`

`C = num2cell(A, dim)`

`C = num2cell(A, [dim1, dim2, ...])`

Description

`C = num2cell(A)` converts numeric array `A` into cell array `C` by placing each element of `A` into a separate cell in `C`. The output array has the same size and dimensions as the input array. Each cell in `C` contains the same numeric value as its respective element in `A`.

`C = num2cell(A, dim)` converts numeric array `A` into a cell array of numeric vectors, the dimensions of which depend on the value of the `dim` argument. Return value `C` contains `numel(A)/size(A, dim)` vectors, each of length `size(A, dim)`. (See [Example 2](#), below). The `dim` input must be an integer with a value from [ndims\(A\)](#) to 1.

`C = num2cell(A, [dim1, dim2, ...])` converts numeric array `A` into a cell array of numeric arrays, the dimensions of which depend on the values of arguments `[dim1, dim2, ...]`. Given the variables `X` and `Y`, where `X=size(A, dim1)` and `Y=size(A, dim2)`, return value `C` contains `numel(A)/prod(X, Y, ...)` arrays, each of size `X-by-Y-by-...`. All `dimN` inputs must be an integer with a value from [ndims\(A\)](#) to 1.

Remarks

`num2cell` works for all array types.

To convert the cell array back to a numeric array, use [cell2mat](#) or `cat(dim, C{:})`. See [Example 2 — Converting to Cell Array and Back to Numeric](#).

Examples

Example 1 — Converting to a Cell Array of the Same Dimensions

Create a 4x7x3 array of random numbers:

```
A = rand(4,7,3);
```

Convert the numeric array to a cell array of the same dimensions:

```
C = num2cell(A);
```

```
whos C
```

Name	Size	Bytes	Class	Attributes
------	------	-------	-------	------------

C	4x7x3	5712	cell	
---	-------	------	------	--

Example 2 — Converting to Cell Array and Back to Numeric

Create a 3-by-4 numeric array `A`:

```
A = [5:3:14; -9:4:3; 20:-4:8]
```

```
A =
```

```

5   8   11   14
-9  -5   -1    3
20  16   12    8
```

Convert `A` to a cell array `C1` of the same size. Convert it back to a numeric array using the [cell2mat](#) function:

```
C1 = num2cell(A)      N1 = cell2mat(C1)
```

```

C1 =      N1 =
 [5] [8] [11] [14]      5   8   11   14
 [-9] [-5] [-1] [3]     -9  -5   -1    3
 [20] [16] [12] [8]      20  16   12    8
```

Convert `A` to a cell array `C2` of 1-by-4 arrays. Convert it back to a numeric array using the [cat](#) function:

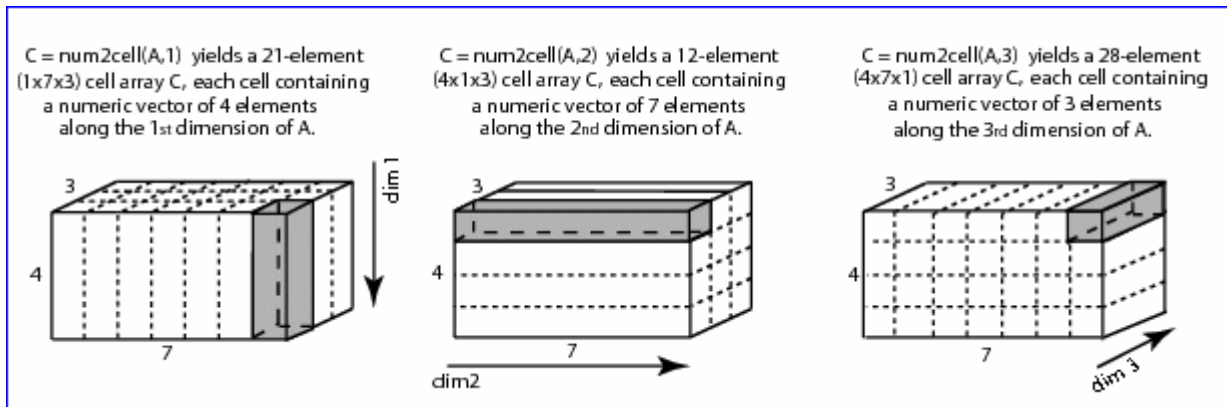
```
C2 = num2cell(A, 2)    N2 = cat(1, C2{:})
```

```

C2 =      N2 =
 [1x4 double]      5   8   11   14
 [1x4 double]      -9  -5   -1    3
 [1x4 double]      20  16   12    8
```

Example 3 — Converting to a Cell Array of Different Dimensions

The following diagrams show the output of `C=num2cell(A,dim)`, where `A` is a 4x7x3 numeric array, and `dim` is one dimension of the input array: either 1, 2, or 3 in this case. The shaded segment of each diagram represents one cell of the output cell array. Each of these cells contains those elements of `A` that are positioned along the dimension specified by `dim`:



Example 4 — Specifying More Than One Dimension for the Output

Given a 4x7x3 numeric array `A`

```
A = rand(4,7,3)
```

and the following two values `X` and `Y`

```
X = size(A,dim1)
```

```
Y = size(A,dim2)
```

then the statement

```
C = num2cell(A, [1 3])
```

returns in `C` a cell array of `numel(A)/prod(X,Y)` numeric arrays, each having the dimensions `X-by-Y`.

See Also

[cat](#), [mat2cell](#), [cell2mat](#)

25. Cell Arrays

On this pages...

[What Is a Cell Array?](#)

[Cell Array Operations](#)

[Creating a Cell Array](#)

[Concatenating Cell Arrays](#)

[Indexing into a Cell Array](#)

[Assigning Values to a Cell Array](#)

[Returning Data from a Cell Array](#)

[Using Cell Arrays with Functions](#)

[Converting Between Cell Array and Struct Array](#)

[Operator Summary](#)

[Function Summary](#)

What Is a Cell Array?

A cell array is a collection of containers called *cells* in which you can store different types of data. The figure shown below represents a 2-by-3 cell array. The cells in row one hold an array of unsigned integers, an array of strings, and an array of complex numbers. Row two holds three other types of arrays, the last being a second cell array nested in the outer one:

cell 1,1 3 4 2 9 7 6 8 5 1	cell 1,2 'Anne Smith' '9/12/94' 'Class II' 'Obs. 1' 'Obs. 2'	cell 1,3 .25+3i 8-16i 34+5i 7+.92i
cell 2,1 1.43 2.98 7.83 5.67 4.21	cell 2,2 -7 2 -14 8 3 -45 52 -16 3	cell 2,3 'text' 4 2 1 5 7.3 2.5 1.4 0 .02 + 8i

Each cell of a cell array contains some type of MATLAB array. The data in this array can belong to any one MATLAB or user-defined class, and can have any valid array dimensions; this includes 1-by-1 (a scalar array), or having one or more dimension equal to zero (an empty array). A second cell of the same array can belong to an entirely different class, and can also have different dimensions than the first. The capability to store arrays of mixed classes and sizes is the most significant feature of a cell array. Another common use of cell arrays is to store character strings that are of unequal length. A cell array that is used for this purpose is called a *cell array of strings*.

Like all MATLAB arrays, cell arrays must be rectangular in shape. That is, the length of all rows must be the same, the length of all columns the same, and so on for every dimension of the array.

In many respects, cell arrays are quite similar to struct arrays. See [Comparing Struct Arrays with Cell Arrays](#) for help in deciding which of these two classes best suits the needs of your applications.

[Back to Top](#)

Cell Array Operations

This table shows the operators used in creating, concatenating, and indexing into the cells of a cell array.

Operation	Syntax	Description
Creating	$C = \{A \ B \ D \ E\}$	Builds a cell array C that can contain data of unlike types in A , B , D , and E .
Concatenating	$C3 = \{C1 \ C2\}$	Concatenates cell arrays $C1$ and $C2$ into a two-element cell array $C3$ such that $C3\{1\} = C1$ and $C3\{2\} = C2$.
	$C3 = [C1 \ C2]$	Concatenates the <i>contents</i> of cell arrays $C1$ and $C2$, assuming that the dimensions of these arrays are compatible.
Indexing	$X = C(s)$	Returns the <i>cells</i> of array C that are specified by subscripts s .
	$X = C\{s\}$	Returns the <i>contents</i> of the cells of C that are specified by subscripts s .
	$X = C\{s\}(t)$	References one or more elements of an array that resides within a cell. Subscript s selects the cell, and subscript t selects the array element(s).

For more information on these operations, see [Creating a Cell Array](#), [Concatenating Cell Arrays](#), and [Indexing into a Cell Array](#), respectively.

[Back to Top](#)

Creating a Cell Array

- [Nesting One Cell Array in Another](#)
- [Creating Cell Arrays One Cell At a Time](#)
- [Alternative Assignment Syntax](#)
- [Preallocating Memory for the Array](#)

Creating cell arrays in MATLAB is similar to creating arrays of other MATLAB classes like `double`, `char`, and so on. The main difference is that, when creating a cell array, you enclose the array contents or indices with curly braces `{ }` instead of square brackets `[ ]`. The curly braces are cell array constructors, just as square brackets are nu-

meric array constructors. Use commas or spaces to separate elements and semicolons to terminate each row.

For example, to create a 2-by-2 cell array **A**, type

```
A = {[1 4 3; 0 5 8; 7 2 9], 'Anne Smith'; 3+7i, -pi:pi/4:pi};
```

This results in the array shown below.

cell 1,1 <table><tr><td>1</td><td>4</td><td>3</td></tr><tr><td>0</td><td>5</td><td>8</td></tr><tr><td>7</td><td>2</td><td>9</td></tr></table>	1	4	3	0	5	8	7	2	9	cell 1,2 'Anne Smith'
1	4	3								
0	5	8								
7	2	9								
cell 2,1 3+7i	cell 2,2 <table><tr><td>[-3.14...3.14]</td></tr></table>	[-3.14...3.14]								
[-3.14...3.14]										

Note The curly braces operator creates two-dimensional matrices only, (including 0-by-0, 1-by-1, and 1-by-n matrices). To create cell arrays of more than two dimensions, see [Creating Multidimensional Arrays](#).

Nesting One Cell Array in Another

To nest one cell array within another, enclose both inner and outer cell arrays with the curly braces `{ }`. The example shown here nests a cell array of vital signs inside a cell array of a person's medical record. (Defining the columns with a **header** is not usually required, and is just used here to make the example simpler):

```
header = {'Name', 'Age', 'Pulse/Temp/BP'};
```

```
records(1,:) = {'Kelly', 49, {58, 98.3, [103, 72]}};
```

```
header, records
```

```
header =
```

```
 'Name' 'Age' 'Pulse/Temp/BP'
```

```
records =
```

```
 'Kelly' [49] {1x3 cell}
```

It is often easier to build a nested cell array in steps. The example below creates the inner cell array, **vitalsigns**, first. The second statement then uses the **vitalsigns** array in creating the outer cell array, **records**:

```
vitalsigns = {60, 98.4, [105, 75]};
```

```
records(1,:) = {'Kelly', 49, vitalsigns}
```

```
record =
```

```
 'Name' 'Age' 'Pulse/Temp/BP'
```

```
 'Kelly' [ 49] {1x3 cell}
```

Verify the new values in the **records** cell array:

```
fprintf('pulse: %d temp: %3.1f bp: %d/%d\n', ...
```

```
 records{3}{:})
```

```
pulse: 60 temp: 98.4 bp: 105/75
```

Creating Cell Arrays One Cell At a Time

You also can create a cell array one cell at a time by using multiple assignment statements. MATLAB expands the size of the cell array with each assignment statement:

```
A(1,1) = {[1 4 3; 0 5 8; 7 2 9]};
```

```
A(1,2) = {'Anne Smith'};
```

```
A(2,1) = {3+7i};
```

```
A(2,2) = {-pi:pi/4:pi};
```

If you assign data to a cell that is outside the dimensions of the current array, MATLAB automatically expands the array to include the subscripts you specify. It fills any intervening cells with empty matrices. For example, the assignment below turns the 2-by-2 cell array **A** into a 3-by-3 cell array:

```
A(3,3) = {5};
```

3-by-3 Cell Array

cell 1,1 1 4 3 0 5 8 7 2 9	cell 1,2 'Anne Smith'	cell 1,3 []
cell 2,1 3+7i	cell 2,2 [-3.14...3.14]	cell 2,3 []
cell 3,1 []	cell 3,2 []	cell 3,3 5

Note If you already have a numeric array of a given name, don't try to create a cell array of the same name by assignment without first clearing the numeric array. If you do not clear the numeric array, MATLAB assumes that you are trying to *mix* cell and numeric syntaxes, and generates an error. Similarly, MATLAB does not clear a cell array when you make a single assignment to it. If any of the examples in this section give unexpected results, clear the cell array from the workspace and try again.

Handling Unassigned Cells. To keep all dimensions of a cell array even, MATLAB automatically fills in any unassigned cells as you build the cell array. For example, if you have a cell array that consists of a row of three elements, and you add one new cell to a second row, MATLAB adds two cells to the new row to keep all rows at the same length. The values of these two cells are set to the empty array []. This is called *scalar expansion*.

MATLAB handles other array types in a similar manner, except that it sets unassigned elements to zero instead of the empty array. This example adds a single element to a 1-by-3 array of type `double`, and then does the same to a cell array:

```
A = [2 4 6]; A(2,1) = 8
```

```
A =
     2     4     6
     8     0     0
```

```
C = {2 4 6}; C(2,1) = {8}
```

```
C =
 [2] [4] [6]
 [8] [] []
```

Alternative Assignment Syntax

When assigning values to a cell array, either of the syntaxes shown below is valid. You can use the braces on the right side of the equation, enclosing the value being assigned, as shown here:

```
A(1,1) = {[1 4 3; 0 5 8; 7 2 9]};
```

```
A(1,2) = {'Anne Smith'};
```

You can also use them on the left side, enclosing the array subscripts:

```
A{1,1} = [1 4 3; 0 5 8; 7 2 9];
```

```
A{1,2} = 'Anne Smith';
```

Preallocating Memory for the Array

MATLAB stores internal information for a cell array in a contiguous segment of memory called a *header*. If you increase the number of cells in a cell array over time, the size of the header also grows, thus using more of this segment in memory. This can eventually lead to "out of memory" errors.

If you can roughly estimate the dimensions of a cell array at the time you create it, you can preallocate the necessary space in memory and help to avoid this type of problem. See the documentation on [Data Structures and Memory](#) to help you make this estimate.

Note Unlike the internal header information of a cell array, the memory consumed by the *data* stored in a cell array is not contiguous. While preallocating memory can help avoid memory problems when increasing the dimensions of a cell array, it does not protect against shortages in memory due to the amount of data you store in the array. Even with preallocated cell (and struct) arrays, you need to take precautions against using more memory than is available.

How to Preallocate Memory. To allocate memory for a 25-by-50 cell array and initialize the entire array to

[], use either of the following two methods:

```
C = cell(25,50);
```

```
C{25,50} = [];
```

After the memory has been allocated, you can begin to construct the array by assigning data to it.

[Back to Top](#)

Concatenating Cell Arrays

There are two ways that you can create a new cell array from existing cell arrays:

- Concatenate entire cell arrays to individual cells of the new array. For example, join three cell arrays together to build a new cell array having three elements, each containing a cell array. This method uses the curly brace { } operator.
- Concatenate the *contents* of the cells into a new array. For example, join cell arrays of size m-by-n1, m-by-n2, and m-by-n3 together to yield a new cell array that is m-by-(n1+n2+n3) in size. This method uses the square bracket [] operator.

Here is an example. First, create three 3-row cell arrays of different widths:

```
C1 = {'Jan' 'Feb'; '10' '17'; uint16(2004) uint16(2001)};
```

```
C2 = {'Mar' 'Apr' 'May'; '31' '2' '10'; ...  
      uint16(2006) uint16(2005) uint16(1994)};
```

```
C3 = {'Jun'; '23'; uint16(2002)};
```

This creates arrays C1, C2, and C3:

```
      C1      C2      C3  
'Jan' 'Feb'  'Mar' 'Apr' 'May' 'Jun'  
'10'  '17'   '31'  '2'  '10'  '23'  
[2004] [2001] [2006] [2005] [1994] [2002]
```

Use the curly brace operator to concatenate entire cell arrays, thus building a 1-by-3 cell array from the three initial arrays. Each cell of this new array holds its own cell array:

```
C4 = {C1 C2 C3}
```

```
C4 =
```

```
{3x2 cell} {3x3 cell} {3x1 cell}
```

Now use the square bracket operator on the same combination of cell arrays. This time MATLAB concatenates the contents of the cells together and produces a 3-by-6 cell array:

```
C5 = [C1 C2 C3]
```

```
C5 =
```

```
'Jan'  'Feb'  'Mar'  'Apr'  'May'  'Jun'  
'10'   '17'   '31'   '2'   '10'   '23'  
[2004] [2001] [2006] [2005] [1994] [2002]
```

Note The notation {} denotes the empty cell array, just as [] denotes the empty matrix for numeric arrays. You can use the empty cell array in any cell array assignments.

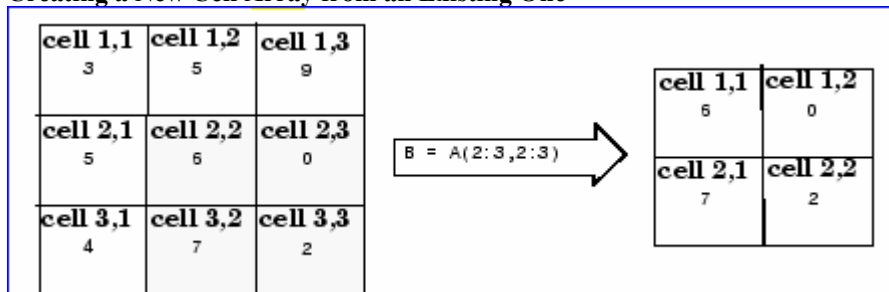
[Back to Top](#)

Indexing into a Cell Array

When working with cell arrays, you have a choice of selecting entire cells of an array to work with, or the contents of those cells. The first method is *cell indexing*, the second is *content indexing*:

- Cell indexing enables you to work with whole cells of an array. You can access single or multiple cells within the array, but you cannot select anything less than the complete cell. If you want to manipulate the cells of an array without regard to the contents of those cells, use cell indexing. This type of indexing is denoted by the parentheses operator ().

Use cell indexing to assign any set of cells to another variable, creating a new cell array.

Creating a New Cell Array from an Existing One

- Content indexing gives you access to the contents of a cell. You can work with individual elements of an array within a cell, but you can only do so for one cell at a time. This indexing uses the curly brace operator { }.

Note The examples in this section focus on two-dimensional cell arrays. For examples of higher-dimension cell arrays, see [Multidimensional Arrays](#).

This example shows how to use cell and content indexing. Start out by creating the following 3-by-3 cell array. The third element of each row is a nested cell array:
 header = {'Name', 'Age', 'Pulse/Temp/BP'};

```
records(1,:) = {'Kelly', 49, {58, 98.3, [103, 72]}};
records(2,:) = {'Mark', 25, {60, 98.6, [105, 75]}};
records(3,:) = {'Susan', 32, {71, 99.1, [110, 78]}};
```

Display the contents of the cell array. Defining the columns with a **header** is not usually required, and is just used here to make the example simpler:

```
header =
    'Name'    'Age'    'Pulse/Temp/BP'
records =
    'Kelly'    [49]    {1x3 cell}
    'Mark'     [25]    {1x3 cell}
    'Susan'    [32]    {1x3 cell}
```

Use content indexing (curly braces) to change one of the names. Content indexing gives you access to what is contained within the cells of the array:

```
records{3,1}='Susanne'
records =
    'Kelly'    [49]    {1x3 cell}
    'Mark'     [25]    {1x3 cell}
    'Susanne'   [32]    {1x3 cell}
```

Use cell indexing (parentheses) to delete an entire row. (You delete part of a cell array by assigning the empty array `[]` to it.) Cell indexing is appropriate here because you do not need access to the *contents* of the row:

```
records(1,:) = []
records =
    'Mark'     [25]    {1x3 cell}
    'Susanne'   [32]    {1x3 cell}
```

Indexing Into Inner Levels of the Cell Array

The cells of a cell array contain arrays of standard MATLAB data types. These arrays use the indexing syntax appropriate to the class of the array. The table below shows examples of statements that use a combination of cell and struct array indexing:

Action	Required Indexing
Access an element of an array in a cell of cell array C.	<code>C{3,15}(5,25)</code>
Access an element of array A, where A is a field of a structure that resides in cell array C.	<code>C{3,15}.A(5,20)</code>
Access an element of an array that resides in a nested cell array.	<code>C{3,15}{5,20}(50,5)</code>
Access an element of array B, where B is a field of structure A, and A is a field of a structure that resides in cell array C.	<code>C{3,15}.A(5,20).B(50,5)</code>
Access an element of cell array B, where B is a field of a structure that resides in cell array C.	<code>C{3,15}.B{5,20}</code>

Start this example by creating the `records` cell array taken from the example in the previous section:

```
records(1,:) = {'Kelly', 49, {58, 98.3, [103, 72]}};
records(2,:) = {'Mark', 25, {60, 98.6, [105, 75]}};
records(3,:) = {'Susan', 32, {71, 99.1, [110, 78]}};
```

Display information from cells in the nested cell array. This requires two adjacent expressions of content indexing, `{2,3}{3}`:

```
fprintf('Name: %s Systolic: %d Diastolic: %d\n', ...
    records{2,1}, records{2,3}{3})
Name: Mark Systolic: 105 Diastolic: 75
```

Indexing Tips

You can index into a nested array in stages rather than all at once. Consider breaking down this indexing expression

```
C{5,3}{4,7}(:,4)
```

into the following:

```
x = C{5,3};    % x is a cell array
y = x{4,7};    % y is also a cell array
z = y(:,4)     % z is a standard array
```

See the section on [Indexing Tips](#) in the documentation on "Structures" for indexing tips that apply to both the `cell` and `struct` classes.

Using Map Objects in Cell Array Indexing

If you want both the numeric indexing of cell arrays and the named containers of [structures](#), you can combine the two to some extent by implementing cell array indexing with a MATLAB Map object (see [Map Containers](#)). The Map

object provides a translation from a name string to a numeric array index. This implementation has the advantage of using less memory than a struct or cell array, but has the disadvantage of being slower.

The example shown here demonstrates the use of a Map object in locating information in a cell array. Note that the names given to the keys of a Map object do not have to adhere to the rules for variable names. In this example, each of the key names contain a space character. This is not allowed in variable names:

```
redSoxStats(57:60,1:4) = { ...
%
% AtBat Runs Hits HR
%
% 653, 118, 213, 17; ... % Pedroia
% 554, 98, 155, 9; ... % Ellsbury
% 538, 91, 168, 29; ... % Youkilis
% 423, 37, 93, 13; ... % Varitek

m1 = containers.Map({'Dustin Pedroia','Jacoby Ellsbury', ...
    'Kevin Youkilis', 'Jason Varitek'}, {57,58,59,60});

m2 = containers.Map({'AtBat','Runs','Hits', 'HR'}, ...
    {1,2,3,4});

player = 'Dustin Pedroia';
fprintf( ...
    '\n %s had %d At Bats and %d hits this season.\n', ...
    player, redSoxStats{m1(player), m2('AtBat')}, ...
    redSoxStats{m1(player), m2('Hits')})
```

Dustin Pedroia had 653 At Bats and 213 hits in the 2008 season.

[Back to Top](#)

Assigning Values to a Cell Array

Use the curly brace { } operator on the right side of the statement to assign values to a cell array:

To store four values in a 2-by-2 cell array, use

```
C = {magic(5), 'Hello'; uint8(100), [1:3:19]}
```

```
C =
```

```
[5x5 double] 'Hello'
[ 100] [1x7 double]
```

The following commands place the values into different cells of cell array C:

```
clear C
```

```
C(3,1:4) = {magic(5), 'Hello', uint8(100), [1:3:19]}
```

```
C =
```

```
    []    []    []    []
    []    []    []    []
[5x5 double] 'Hello' [100] [1x7 double]
```

```
clear C
```

```
C(2:3,5:6) = {magic(5), 'Hello', uint8(100), [1:3:19]}
```

```
C =
```

```
    []    []    []    []    []
    []    []    []    [5x5 double] [ 100]
    []    []    []    'Hello' [1x7 double]
```

The [deal](#) function offers an alternative method of writing to the cell array. These two statements produce the same result as the statements shown above that use the curly braces { } operator:

```
[C{3,1:4}] = deal(magic(5), 'Hello', uint8(100), [1:3:19]);
```

```
[C{2:3,5:6}] = deal(magic(5), 'Hello', uint8(100), [1:3:19]);
```

[Back to Top](#)

Returning Data from a Cell Array

This section describes the syntax to use to have MATLAB return data from a cell array, how to assign data from a cell array to a comma-separated list or separate output variables, and also how to plot the contents of a cell array.

Obtaining Values from the Array

The following table shows a number of different ways of returning data from a cell array. The variable C is a 3x4 cell array in which each cell contains a 2x5 array of class double.

Values to be acquired	MATLAB Statement	Data Structure Returned
Top level of cell array <code>c</code>	<code>c</code>	3x4 cell array.
Top level of cell array <code>c</code> , as a vector	<code>c(:)</code>	12x1 cell array.
Selected cells in cell array <code>c</code>	<code>c(2:3,1:3)</code>	2x3 cell array.
Full contents of one cell in cell array <code>c</code> .	<code>c{2,3}</code>	2x5 array of double.
Full contents of selected cells in cell array <code>c</code> .	<code>c{2:3,3:4}</code>	4-item comma-separated list of 2x5 double.
Full contents of all cells in cell array <code>c</code> .	<code>c{:}</code>	12-item comma-separated list of 2x5 double.
Selected elements of one cell in cell array <code>c</code> . You cannot use multiple elements of <code>c</code> with this syntax.	<code>c{3,4}(2,3:5)</code>	1x3 array of double.

The first three of these indexing expressions provide no access to individual elements of the cells. You could use these expressions to copy, rearrange, or delete parts of the cell array.

Assigning Cell Values to a Comma-Separated List

Accessing a single value from one cell of a cell array is no different from accessing one of the elements of any other MATLAB data type. Accessing multiple elements, however, can be quite different. Multiple elements of a cell array cannot be assigned to a single variable because they do not necessarily belong to the same class. Instead, MATLAB assigns values from a cell array to a series of separate variables called a comma-separated list. Here is an example of such a list:

First, create a 3-by-3 cell array called `records`:

```
records(1,:) = {'Kelly', 49, {58, 98.3, [103, 72]}};
records(2,:) = {'Mark', 25, {60, 98.6, [105, 75]}};
records(3,:) = {'Susan', 32, {71, 99.1, [110, 78]}};
```

Displaying one column of the cell array causes MATLAB to return three separate values, each, in succession, assigned to the `ans` variable:

```
records{:,2}
ans =
    49
ans =
    25
ans =
    32
```

The potential problem with this type of output is that MATLAB overwrites the `ans` variable for each value returned. If you only want to display these values, then this command should suit your purpose. The next section shows how to assign to variables that you can reuse.

Assigning Cell Values to Separate Variables

If you were to assign multiple elements of a cell array to just one variable, MATLAB uses that variable to return the first value, but is unable to return all values of the array:

```
x = records{:,2}
x =
    49
```

If you know how many values there are in the cell array elements you are trying to access, then you can provide that many outputs in the command, as shown here:

```
[v1 v2 v3] = records{:,1}
v1 =
    Kelly
v2 =
    Mark
v3 =
    Susan
```

As in the previous example, this is a comma-separated list. As you can see here, each return variable adopts the class and size of the cell array element assigned to it:

```
whos v1
Name      Size      Bytes Class  Attributes

v1        1x5         10 char
```

```
whos v2
```

```

Name      Size      Bytes Class  Attributes

v2        1x4        8 char
Plotting the Cell Array

```

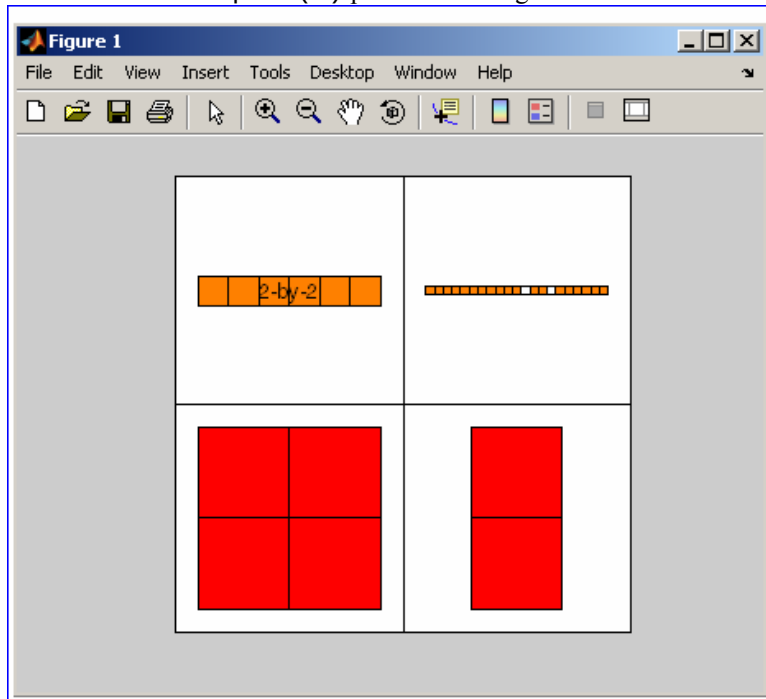
For a high-level graphical display of cell architecture, use the [cellplot](#) function. Consider a 2-by-2 cell array containing two text strings, a matrix, and a vector:

```

c{1,1} = '2-by-2';
c{1,2} = 'eigenvalues of eye(2)';
c{2,1} = eye(2);
c{2,2} = eig(eye(2));

```

The command `cellplot(c)` produces this figure.



[Back to Top](#)

Using Cell Arrays with Functions

This section describes how to apply a function to data contained within a cell array using the `cellfun` function, and also how to pass arguments to and from a function using cell arrays.

Applying a Function to the Cells of a Cell Array

Use the [cellfun](#) function to run a function on each field of a scalar cell array. This example runs an anonymous function on a cell array containing the days of a week. The anonymous function, `@(x)x(1:3)`, shortens each string to its first three characters. The function reference page for `cellfun` explains the use of the `UniformOutput` option:

```

days{1} = 'Sunday'; days{2} = 'Monday';
days{3} = 'Tuesday'; days{4} = 'Wednesday';
days{5} = 'Thursday'; days{6} = 'Friday';
days{7} = 'Saturday';

```

```

shortNames = cellfun(@(x)x(1:3), days, 'UniformOutput', false)
shortNames =

```

```
'Sun' 'Mon' 'Tue' 'Wed' 'Thu' 'Fri' 'Sat'
```

See the reference page for `cellfun` for additional help on using this function.

Passing Variable Numbers of Arguments

You can call a function with variable numbers of input or output arguments by using the terms [varargin](#) and [varargout](#) in the respective input and output argument lists for that function. The function being called provides access to these arguments using cell arrays named `varargin` and `varargout`. See [Passing Variable Numbers of Arguments](#) in the documentation on "Functions and Scripts".

Passing Arguments in a Cell Array

A simple and easily maintainable way to pass arguments to or from a function is to package them in a cell array, and then pass the entire cell array to the function. This example passes information pertaining to four United States presidents to the function `showPresInfo`:

```
USPres = cell(30,3); % Allocate memory for the array.
```

```
USPres{27,1} = 'William Howard Taft'; % 27th US President
USPres{27,2} = [1909, 1913]; % Term
USPres{27,3} = 'James S. Sherman'; % Vice President
```

```
USPres{28,1} = 'Woodrow Wilson'; % 28th
USPres{28,2} = [1913, 1921];
USPres{28,3} = 'Thomas R. Marshall';
```

```
USPres{29,1} = 'Warren G. Harding'; % 29th
USPres{29,2} = [1921, 1923];
USPres{29,3} = 'Calvin Coolidge';
```

```
USPres{30,1} = 'Calvin Coolidge'; % 30th
USPres{30,2} = [1923, 1929];
USPres{30,3} = 'Charles Dawes';
```

Write a short program to display the information passed in:

```
function showPresInfo(number, info)
info(number-26, :)
```

Call this program, passing rows 27 through 30 of the cell array:

```
showPresInfo(29, USPres(27:30, :))
```

```
ans =
```

```
    'Warren G. Harding'
```

```
    [1x2 double]
```

```
    'Calvin Coolidge'
```

Passing Selected Cells of a Cell Array

You can also pass selected cells of a cell array in a function call. This example passes the names of the four Vice Presidents in the form of a comma-separated list. In this case, the function being called, `showVPInfo`, receives these strings as four separate input arguments:

The value passed to this function is a list of four separate items:

```
USPres(27:30).vp
```

```
ans =
```

```
    James S. Sherman
```

```
ans =
```

```
    Thomas R. Marshall
```

```
ans =
```

```
    Calvin Coolidge
```

```
ans =
```

```
    Charles Dawes
```

Write a short program that displays the name of a selected Vice President. Use the [varargin](#) function to accept and unpack the four separate input arguments generated by the `USPres{27:30, 3}` input:

```
function showVPInfo(number, varargin)
```

```
str = ['The Vice President who served with ', ...
```

```
    'the %dth US President was %s\n'];
```

```
fprintf(str, number, varargin{number-26})
```

Run the program a couple of times to verify that the names of more than one Vice Presidents were passed:

```
showVPInfo(28, USPres{27:30,3})
```

```
The Vice President who served with the 28th US President was
```

```
    Thomas R. Marshall
```

```
showVPInfo(30, USPres{27:30,3})
```

```
The Vice President who served with the 30th US President was
```

```
    Charles Dawes
```

[Back to Top](#)

Converting Between Cell Array and Struct Array

The [cell2struct](#) function converts a cell array to a struct array. The statement

```
s = cell2struct(c,f,d)
```

converts a cell array `c` into a struct array `s` having the fields named in `f` and based on the `d` axis of the input cell array.

The [struct2cell](#) function converts a structure array to a cell array. The statement

```
c = struct2cell(s)
```

converts an `m`-by-`n` structure `s` that has `p` fields into a `p`-by-`m`-by-`n` cell array `c`:

Conversion Example

This example converts a 4-by-1-by-2 cell array `USPres_c1` to a 1-by-2 struct array `USPres_s1` with four fields, and then back to a cell array `USPres_c2` that is equal to the original.

Create the original cell array:

```
USPres_c1{1,1,1} = 'Franklin D. Roosevelt';
USPres_c1{2,1,1} = 'Democratic';
USPres_c1{3,1,1} = [1933, 1945];
USPres_c1{4,1,1} = ...
    {'John Garner','Henry Wallace','Harry S Truman'};
```

```
USPres_c1{1,1,2} = 'Harry S Truman';
USPres_c1{2,1,2} = 'Democratic';
USPres_c1{3,1,2} = [1945, 1953];
USPres_c1{4,1,2} = {'Alben Barkley'};
whos USPres_c1
Name      Size      Bytes Class  Attributes

USPres_c1  4x1x2      964 cell
```

Convert the cell array to a struct array:

```
USPres_s1 = cell2struct(USPres_c1, ...
    {'name','party','term','vp'}, 1);
whos USPres_s1
Name      Size      Bytes Class  Attributes

USPres_s1  1x2      1220 struct
```

Convert back to a cell array and compare it with the original:

```
USPres_c2 = struct2cell(USPres_s1);
whos USPres_c2
Name      Size      Bytes Class  Attributes

USPres_c2  4x1x2      964 cell
```

```
isequal(USPres_c1, USPres_c2)
ans =
    1
```

Operator Summary

This section summarizes the following types of operators that work with cell arrays:

- [Operators That Construct the Cell Array](#)
- [Operators That Concatenate Cells and Cell Content](#)
- [Operators Used for Cell Array Indexing](#)

Operators That Construct the Cell Array

Syntax	Description
<code>C = {A B D E}</code>	Builds a cell array <code>C</code> that can contain data of unlike types in <code>A</code> , <code>B</code> , <code>D</code> , and <code>E</code> .

OPERATORS THAT CONCATENATE CELLS AND CELL CONTENT

Syntax	Description
<code>C3 = {C1 C2}</code>	Concatenates cell arrays <code>C1</code> and <code>C2</code> into a two-element cell array <code>C3</code> , such that <code>C3{1} = C1</code> and <code>C3{2} = C2</code> .
<code>C3 = [C1 C2]</code>	Concatenates the <i>contents</i> of cell arrays <code>C1</code> and <code>C2</code> into a new cell array with <code>length(C3)==length(C1)+length(C2)</code> .

Operators Used for Cell Array Indexing

SYNTAX	DESCRIPTION
<code>X = C(s)</code>	Returns the <i>cells</i> of array <code>C</code> that are specified by subscripts <code>s</code> .
<code>X = C{s}</code>	Returns the <i>contents</i> of the cells of <code>C</code> that are specified by subscripts <code>s</code> .
<code>X = C{s}(v)</code>	References one or more elements of an array that resides within a cell. Subscript <code>s</code> selects the single cell, and subscript <code>v</code> selects the array element(s).
<code>X =</code>	Returns the contents of a nested cell array. Subscripts for the outer array <code>C</code> are <code>s₁</code> . These

SYNTAX	DESCRIPTION
$C\{s_1\}\{s_2\}$	subscripts can only refer to one cell of the outer array. Subscripts for the inner cell array are s_2 .
$X = C\{s_1\}\{s_2\}(v)$	Returns one or more elements of an array that reside in a nested cell array.
$X = C\{s\}(t) . f(v)$	Returns one or more elements of an array that reside in a struct field, where the struct resides in a cell of cell array C. Subscripts are s for the cell array, t for the struct array, and v for the lowest-level array.

[Back to Top](#)

Function Summary

This section summarizes the following types functions that work with cell arrays:

- [Functions Related to Constructing the Array](#)
- [Functions Related to the Type of the Array](#)
- [Functions Related to Obtaining Cell Array Contents](#)
- [Functions Related to Applying Functions to a Cell Array](#)
- [Functions Used with Cell Array Conversion](#)

Functions Related to Constructing the Array

Function	Description
cat	Concatenate arrays along specified dimension.
cell	Create cell array.
horzcat	Concatenate arrays horizontally.
length	Length of array.
ndims	Number of array dimensions.
numel	Number of elements in array or subscripted array expression.
repmat	Replicate and tile array.
reshape	Reshape array.
size	Size of array.
vertcat	Concatenate arrays vertically.

Functions Related to the Type of the Array

Function	Description
cell2struct	Convert cell array to structure array.
class	Create object or return class of object.
iscell	Determine whether input is cell array.
struct2cell	Convert structure to cell array.
whos	List variables in workspace.

Functions Related to Obtaining Cell Array Contents

FUNCTION	DESCRIPTION
CELLDISP	DISPLAY CELL ARRAY CONTENTS.
CELLPLOT	DISPLAY A GRAPHICAL DEPICTION OF A CELL ARRAY.
DEAL	COPY INPUT TO SEPARATE OUTPUTS.

Functions Related to Applying Functions to a Cell Array

Function	Description
cellfun	Apply function to each field of scalar cell array.

Function	Description
varargin	Variable length input argument list.
varargout	Variable length output argument list.

Functions Used with Cell Array Conversion

Function	Description
cell2struct	Convert cell array into struct array.
struct2cell	Convert struct array into cell array.
mat2cell	Divide matrix into cell array of matrices
cell2mat	Convert cell array of matrices to single matrix
num2cell	Convert numeric array to cell array

26. copyfile

Copy file or folder

GUI Alternatives

Copy files and folders using the Current Folder browser.

Syntax

```
copyfile('source','destination')
copyfile('source','destination','f')
[status] = copyfile(...)
[status, message] = copyfile(...)
[status,message,messageid] = copyfile(...)
```

Description

`copyfile('source','destination')` copies the file or folder named `source` to the file or folder `destination`. `source` and `destination` are 1 x n strings. Use full path names or path names relative to the current folder. To copy multiple files or folders, use one or more wildcard characters * after the last file separator in `source`. You cannot use a wildcard character in `destination`.

`copyfile('source','destination','f')` copies `source` to `destination`, even when `destination` is not writable. The state of the read-write attribute for `destination` does not change. You can use `f` with any syntax for `copyfile`.

`[status] = copyfile(...)` reports the outcome as a logical scalar, `status`. The value is 1 for success and 0 for failure.

`[status, message] = copyfile(...)` returns any warning or error message as a string to

message. When `copyfile` succeeds, message is an empty string.

`[status,message,messageid] = copyfile(...)` returns any warning or error identifier as a string to `messageid`. When `copyfile` succeeds, `messageid` is an empty string.

Remarks

- The timestamp for destination is the same as the timestamp for source.
- When source is a folder, destination must be a folder.
 - - o When source is a folder and destination does not exist, `copyfile` creates destination and copies the contents of source into destination.
 - o When source is a folder and destination is an existing folder, `copyfile` copies the contents of source into destination.
 - o When source is multiple files and destination does not exist, `copyfile` creates destination.
- For behavior not explicitly stated in the description or remarks, `copyfile` follows the behavior of the operating system copy command. For example, on UNIX platforms, `copyfile` is like the UNIX command `cp -p`.

Examples

Copy `myFun.m` from the current folder to `d:/work/Projects/`, keeping the same file name:

```
copyfile('myFun.m','d:/work/Projects/')
```

Make a copy of `myFun.m` in the current folder, assigning the name `myFun2.m` to it:

```
copyfile('myFun.m','myFun2.m')
```

Copy all files and subfolders whose names begin with `my`, from the `Projects` folder. Copy to the existing folder `newProjects`, which is at the same level as the current folder:

```
copyfile('Projects/my*','../newProjects/')
```

Copy the contents of the `Projects` folder to the `d:/work/newProjects` folder. `Projects` is in the current folder. `newProjects` does not exist.

```
copyfile('Projects','d:/work/newProjects')
```

Copy `myFun.m` from the current folder to `d:/work/restricted/myFun2.m`, where `myFun2.m` is read-only. Return output to determine success:

```
[status,message,messageid]=copyfile('myFun.m','d:/work/restricted/myFun2.m')
```

The results show that `copyfile` failed:

```
status =
0
```

```
message =
    Cannot write to destination: d:/work/restricted/myFun2.m. Use the 'f' op-
tion to override
```

```
messageId =
    MATLAB:COPYFILE:ReadOnly
```

Copy myFun.m from the current folder to d:/work/restricted/myFun2.m, where myFun2.m is read-only. Use 'f' to force the copy, even though myFun2.m is read-only. Return output to determine success:

```
[status,message,messageId]=copyfile('myFun.m','d:/work/restricted/myFun2.m','f')
```

The results show that copyfile succeeded:

```
status =
    1
```

```
message =
    ''
```

```
messageId =
    ''
```

See Also

[cd](#), [delete](#), [dir](#), [fileattrib](#), [filebrowser](#), [fileparts](#), [mkdir](#), [movefile](#), [rmdir](#)

User Guide topics:

- [Path Names in MATLAB](#)
- [Creating, Changing, and Deleting Files and Folders](#)

27. clear

Remove items from workspace, freeing up system memory

Syntax

```
clear
clear name
clear name1 name2 name3 ...
clear global name
clear -regexp expr1 expr2 ...
clear global -regexp expr1 expr2 ...
clear keyword
clear('name1','name2','name3',...)
```

Description

clear removes all variables from the workspace, releasing them from system memory.

`clear name` removes just the M-file or MEX-file function or variable name from your base workspace. If called from a function, `clear name` removes name from both the function workspace and in your base workspace. You can use wildcards (*) to remove items selectively. For example, `clear my*` removes any variables whose names begin with the string `my`. Clearing removes debugging breakpoints in M-files and reinitializes persistent variables. If name is global, `clear` removes it from the current workspace, but it remains accessible to any functions declaring it global. If name has been locked by `mlock`, it remains in memory.

Use a *partial path* to distinguish between different overloaded versions of a function. For example, `clear polynom/display` clears only the `display` method for `polynom` objects, leaving any other implementations in memory.

`clear name1 name2 name3 ...` removes `name1`, `name2`, and `name3` from the workspace.

`clear global name` removes the global variable `name`. If `name` is global, `clear name` removes `name` from the current workspace, but leaves it accessible to any functions declaring it as global. Use `clear global name` to remove a global variable completely.

`clear -regexp expr1 expr2 ...` clears all variables that match any of the [regular expressions](#) listed. This option only clears variables.

`clear global -regexp expr1 expr2 ...` clears all global variables that match any of the regular expressions listed.

`clear keyword` clears the items indicated by *keyword*. See the following section, Inputs, for a list of keywords and their descriptions.

`clear('name1', 'name2', 'name3', ...)` is the function form of the syntax. Use this form for variable names and function names stored in strings.

If you name a variable `all`, `classes`, `functions`, `java`, `import`, or `variables`, calling `clear` followed by that name deletes the variable with that name. `clear` does not interpret the name as a keyword in this context. For example, if the workspace contains variables `a`, `all`, `b`, and `ball`, `clear all` deletes the variable `all` only.

You can clear the handle of a figure or other object, but that does not remove the object itself. Use `delete` to remove objects and files. Deleting an object does not delete the variable, if any, used for storing its handle.

On UNIX systems, `clear` does not affect the amount of memory allocated to the MATLAB process.

The `clear` function does not clear Simulink models. Use [close](#) instead.

Inputs

keyword

A name specifying whether to clear variables, classes, or packages or subsets of them, as described in the following table:

Keyword	Items Cleared
<code>all</code>	Removes all functions, variables, global variables, and MEX-files from your base workspace. If called from a function, <code>clear all</code> also clears the function workspace, leaving both workspaces empty. <code>clear all</code> removes debugging

Keyword	Items Cleared
	breakpoints in M-files and reinitializes persistent variables. When issued from the Command Window prompt, also removes the Sun Microsystems Java packages import list.
<code>classes</code>	The same as <code>clear all</code> , but also clears MATLAB class definitions. This variant issues a warning and does not remove class definitions for objects outside the workspace (for example, in user data or persistent variables in a locked M-file). Call <code>clear classes</code> whenever you change a class definition, including when the number or names of properties, methods, or events or any of their attributes change.
<code>functions</code>	Clears all the currently compiled M-functions and MEX-functions from memory. If called from a function, <code>clear function</code> removes all functions in both the function workspace and in your base workspace. It also removes debugging breakpoints in the function M-file and reinitializes persistent variables.
<code>global</code>	Removes all global variables in both the function workspace and in your base workspace. If called from a function, <code>clear global</code> removes all global variables in both the function workspace and in your base workspace.
<code>import</code>	Removes the Java packages import list. Use this keyword only from the Command prompt; do not use it in a function.
<code>java</code>	The same as <code>clear all</code> , but also clears the definitions of all Java classes defined by files on the Java dynamic class path. See The Java Class Path in the External Interfaces documentation for more information). If any Java objects exist outside the workspace (for example, in user data or persistent variables in a locked M-file), <code>clear</code> issues a warning and does not remove the Java class definition. Issue a <code>clear java</code> command after modifying any files on the Java dynamic class path.
<code>mex</code>	The same as <code>clear functions</code> , but clears only MEX-functions, other than locked functions or functions that are currently in use. It also clears breakpoints and persistent variables.
<code>variables</code>	Clears all variables from the workspace.

Examples

Given a workspace containing the following variables

Name	Size	Bytes	Class
<code>c</code>	3x4	1200	cell array
<code>frame</code>	1x1		java.awt.Frame
<code>gbl1</code>	1x1	8	double array (global)
<code>gbl2</code>	1x1	8	double array (global)
<code>xint</code>	1x1	1	int8 array

you can clear a single variable, `xint`, by typing

```
clear xint
```

To clear all global variables, type

```
clear global
```

```
whos
```

Name	Size	Bytes	Class
------	------	-------	-------

```
c          3x4          1200  cell array
frame     1x1          java.awt.Frame
```

Using regular expressions, clear those variables with names that begin with Mon, Tue, or Wed:

```
clear('-regexp', '^Mon|^Tue|^Wed');
```

To clear all compiled M- and MEX-functions from memory, type `clear functions`. In the following case, `clear functions` was unable to clear one M-file function from memory, `testfun`, because the function is locked.

```
clear functions    % Attempt to clear all functions.
```

```
inmem
```

```
ans =
    'testfun'      % One M-file function remains in memory.
```

```
mislocked testfun
```

```
ans =
     1            % This function is locked in memory.
```

```
munlock testfun
```

```
% Once you unlock the function from memory, you can clear it.
```

```
clear functions
```

```
inmem
```

```
ans =
    Empty cell array: 0-by-1
```

Clearing handle graphics handles does not remove the objects themselves, nor does deleting the objects remove variables storing their handles.

```
hf = figure; % Creates figure object, stores handle in variable hf
delete(hf)   % Removes figure object, but not the variable hf
clear hf     % Removes hf from the workspace; figure could still exist
Alternatives
```

As an alternative to the `clear` function, use **Edit > Clear Workspace** in the MATLAB desktop.

See Also

[clc](#) | [clearvars](#) | [close](#) | [delete](#) | [import](#) | [inmem](#) | [load](#) | [memory](#) | [mlock](#) | [munlock](#) | [pack](#) | [persistent](#) | [save](#) | [who](#) | [whos](#) | [workspace](#)

How To

- [MATLAB Workspace](#)
- [Strategies for Efficient Use of Memory](#)

БИБЛИОГРАФИЯ

1. Е.Н. Бендерская, Д.Н. Колесников и др. Системный анализ и принятие решений. СПб, изд. СПбГТУ. 1999, 206 с.
2. С.А. Ашманов, А.В. Тимохов. Теория оптимизации в задачах и упражнениях. М.: Наука. Гл. ред. физ.-мат. 1991. – 448 с.
3. А.С. Гурин, Я.С. Дымарский, А.Д. Меркулов. Задачи и методы оптимального распределения ресурсов. М.: Сов. радио. 1968. – 464 с.
4. Задачи линейного программирования и методы их решения. <http://vvo.psati.ru/files/RPU/page2.files/-index3.htm>. 5.12.05.
5. И.Л. Акулич. Математическое программирование в примерах и задачах. Учеб. пособие для студентов эконом. спец. вузов. – М.: Высш. шк., 1986. – 319 с., ил.
6. Экономико-математические методы. Математические методы и модели в экономике. Раздаточный материал/ сост. Аксенова Р.Н. – Владивосток, ДВГАЭУ, 2001. – 92 с.
7. Ермольев Ю.Н., Ляшко И.И., Михалевич В.С., Тюнтя В.И. Математические методы исследования операций. Учебн. пособие для вузов. – Киев: Вища школа. Головное издательство. 1979. – 312 с.
8. Телешова Е. Лабораторная работа №5. Транспортная задача линейного программирования. www.5ballov.ru. Библиотека 5баллов.ru.
9. Вепринская Т. Методические указания к выполнению курсовой работы. Государственная морская академия им. адм. С.О. Макарова. <http://zf.boom.ru/cursovik.doc>, 13.12.05.
10. Е. В. Алексеева. Построение математических моделей целочисленного линейного программирования. Примеры и задачи. Учебное пособие. Новосибирский Государственный Университет. – Новосибирск. 2012. – 132 с. [Электронный ресурс] – Доступ: http://math.nsc.ru/~alekseeva/Textbooks/textbook_model_building.pdf, посещ. 1.09.16.
11. Общая распределительная задача линейного программирования. Allmath.ru. [zadachi6.htm](http://allmath.ru/zadachi6.htm), 13.12.05.
12. Сидин Э.Ф. Экономико-математическое моделирование. Учебное пособие. Оптимизационные ЭММ. <http://referat.cis2000.ru/books/book8/ch4.shtml>.
13. Дистанционный курс обучения по дисциплине МЕТОДЫ ОПТИМИЗАЦИИ СумГУ. http://www.kti.ru/data/84/graf_r.html, 15.11.2011.
14. Алексинская Т.В., Сербин В.Д., Катаев А.В. Учебно-методическое пособие по курсу "Экономико-математические методы и модели. Линейное программирование". Таганрог: Изд-во ТРТУ, 2001. 79 с. Электронная версия книги в Интернете - <http://www.aup.ru/books/m85/>
15. Решение задач линейного программирования с помощью поиска решений в среде EXCEL. Интернет-сайт: <http://asu.bru.mogilev.by>.
16. Вепринская Т. Методические указания к выполнению курсовой работы. Государственная морская академия им. адм. С.О. Макарова. <http://zf.boom.ru/cursovik.doc>, 13.12.05.
17. Справка Матлаба R2009b [электронный ресурс]. Help, Optimization Toolbox, fx–Functions, Minimization, fx–linprog, fx–bintprog.
18. LineSpec (Line Specification). Справка Матлаба R2009b [электронный ресурс] - MATLAB/fx–Functions/ Graphics/ Basic Plots and Graphs/ fx–LineSpec (Line Specification).

Рыкин Олег Романович

ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ В МАТЛАБЕ

**УНИВЕРСАЛЬНЫЕ ЛИНПРОГОПТИМИЗАТОРЫ:
ПРОИЗВОДИТЕЛЬНОСТЬ
И ТАБЛИЧНЫЙ ФОРМАТ РЕЗУЛЬТАТА**

ЗАДАНИЯ И ЗАДАЧИ

Налоговая льгота — Общероссийский классификатор продукции
ОК 005-93, т. 2; 95 3005 — научная и производственная литература

Подписано в печать 03.10.2016. Формат 60×84/16. Печать цифровая.

Усл. печ. л. 13,0. Тираж 33. Заказ 14809b.

Отпечатано с готового оригинал-макета, предоставленного автором,
в Типографии Политехнического университета.

195251, Санкт-Петербург, Политехническая ул., 29.

Тел.: (812) 552-77-17; 550-40-14.