Министерство образования и науки Российской Федерации

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

И. Г. Черноруцкий

МЕТОДЫ ПРИНЯТИЯ РЕШЕНИЙ

Санкт-Петербург

2012

УДК 681.3.06

ББК 22.18

Ч-49

Черноруцкий И.Г. **Методы принятия решений:** – учеб. пособие / И.Г. Черноруцкий .

Книга состоит из четырех частей: методы принятия решений, методы оптимизации, экспертные системы принятия решений, примеры систем поддержки принятия решений.

Основная направленность данного учебного издания – пользовательский, прикладной аспект. Излагаемые теория, методы и алгоритмы позволят читателю овладеть принципами корректного и обоснованного применения существующих программных систем поддержки принятия решений, а также создавать новые системы.

Учебное пособие предназначено для обучения студентов высших учебных заведений по направлению подготовки магистров «Информатика и вычислительная техника», «Программная инженерия», «Системный анализ и управление».

Предисловие 7
Введение 14
1 МЕТОДЫ ПРИНЯТИЯ РЕШЕНИЙ.. 30
1.1 Задача принятия решений. 30
1.1.1 Постановка задачи принятия решений. Критериальный язык описания выбора. 30
1.1.2 Описание выбора на языке бинарных отношений. Формальные модели задачи ПР. 36
1.1.3 Связь различных способов описания выбора. Однокритериальный и многокритериальный выбор. 41
1.2 Многокритериальные модели принятия решений в условиях определенности. 47
1.2.1 Методы многокритериальной оптимизации. 48
1.2.2 Максиминные стратегии. 55
1.2.3 Метод линейной свертки и главного критерия. Лексикографическая оптимизация 61
1.3 Принятие решений в условиях неопределенности. 70
1.3.1 Основные понятия 71
1.3.2 Принятие решений в условиях риска. 76
1.3.3 Критерии принятия решений в условиях полной неопределенности... 86
1.3.4 Некоторые трудности. 94
1.3.5 Принятие решений в условиях конфликта (элементы теории игр). 98
1.4 Многостадийные процессы принятия решений. 114
1.4.1 Постановка задачи. 114

1.4.2 Детерминистский случай. Метод Беллмана. 117
1.4.3 Многостадийные процессы в условиях неопределенности. 121
1.5 Методы многокритериального выбора на основе дополнительной информации пользователя. 130
1.5.1 Адаптивные процедуры выбора. 130
1.5.2 Выбор на основе метода t-упорядочения 138
1.5.3 Задачи с малым числом критериев и альтернатив 146
1.5.4 Метод ограничений 157
1.5.5 Рандомизированные стратегии принятия решений 159
1.6 КОММЕНТАРИЙ 163
2 ЭКСПЕРТНЫЕ СИСТЕМЫ ПРИНЯТИЯ РЕШЕНИЙ. 167
2.1 Введение. 167
2.1.1 Назначение и области применения экспертных систем. 169
2.1.2 Структура экспертной системы. 173
2.1.3 Основные классы и виды экспертных систем. 176
2.2 Продукционные экспертные системы. 178
2.2.1 Основные компоненты продукционной экспертной системы. 178
2.2.2 Прямая и обратная цепочки вывода 182
2.2.3 Простая диагностирующая экспертная система. 185
2.2.4 Формальное представление продукционной экспертной системы. 188
2.3 Представление и использование нечетких знаний 193
2.3.1 Элементы теории вероятностей. 193
2.3.2 Байесовский подход. 196

2.4 Нейлоровские диагностирующие системы. 200
2.4.1 Элементы механизма логического вывода. 200
2.4.2 Цены свидетельств – косвенная цепочка рассуждений. 203
2.4.3 Правила остановки. 207
2.4.4 Структура базы знаний и алгоритм логического вывода 209
2.4.5 Пример базы знаний. 213
3 ПРИМЕРЫ СИСТЕМ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ. 215
3.1 Quick choice - система многокритериального выбора вариантов. 215
3.1.1 Область применения системы.. 215
3.1.2 Необходимые исходные данные 216
3.1.3 Типы критериев 217
3.1.4 Функции, реализованные в системе. 218
3.1.5 Инсталляция системы.. 219
3.1.6 Первый запуск системы.. 222
3.1.7 Получение данных из текстового файла или базы данных.. 223
3.1.8 Принятие решений в диалоге с пользователем. 226
3.1.9 Метод ограничений. 231
3.1.10 Главное окно. 236
3.1.11 Главное меню программы. 237
3.1.12 Рабочие окна программы. 241
3.1.13 Нормализованные исходные данные. 1
3.1.14 Создание, загрузка и сохранение задачи 1
3.1.15 Создание отчета 259

3.1.16 Пример решения задачи 260
3.2 EXPERT – KN. ИНСТРУМЕНТАЛЬНОЕ СРЕДСТВО ПОСТРОЕНИЯ НЕЙЛОРОВСКИХ ДИАГНОСТИРУЮЩИХ СИСТЕМ. 266
3.2.1 Назначение и структура системы.. 266
3.2.2 Общая характеристика системы. 271
3.2.3 Инсталляция системы.. 273
3.2.4 Система в конкретной предметной области.......................... 275
3.2.5 Работа готовой экспертной системы. 295
3.2.6 Пример решения модельной задачи. 305
3.2.7 Заключение. 314
4 БИБЛИОГРАФИЧЕСКИЙ СПИСОК 315

Предисловие

В книге описываются методы теории принятия решений, составляющей важнейший раздел системного анализа. Термин «системный анализ» понимается здесь как совокупность методов, основанных на использовании компьютерных технологий и ориентированных на исследование сложных систем – технических, экономических, экологических, программных и т.д. Результатом этих исследований, как правило, является выбор определенной альтернативы: плана развития фирмы, параметров конструкции, стратегии управления проектом и т.п. Таким образом, системный анализ согласно принятой в данной книге интерпретации – это дисциплина, занимающаяся проблемами принятия решений в условиях, когда выбор альтернативы требует анализа сложной информации, характеризующей реальную ситуацию. С другой стороны, известный термин «исследование операций» часто трактуется как дисциплина, занимающаяся количественным обоснованием решений в различных областях целенаправленной человеческой деятельности. Поэтому можно сказать также, что наша книга посвящена исследованию операций.

Основная направленность данного издания – пользовательский, прикладной аспект. Пользовательский аспект в данном случае понимается несколько шире, чем это обычно принято. Речь пойдет не только и не столько о пользовательском аспекте по отношению к каким-то программным продуктам, пакетам и системам поддержки принятия решений (СППР), но, в основном, о пользовательском аспекте в смысле внутреннего функционального наполнения подобных программных продуктов, о реализованных в них базовых принципах и применяемой в данной области терминологии. Приводимые сведения помогут будущему пользователю «вскрывать» используемые СППР, по крайней мере, в общих чертах и представлять, каких результатов следует ожидать от конкретной СППР и каких она дать не сможет.

Ссылки на литературу внутри текста отсутствуют, однако в конце книги приведен полный список использованной литературы. Автор совсем не претендует на новизну и оригинальность приводимого материала, хотя некоторые подходы и методы кажутся ему новыми. Книга имеет откровенно учебный и вводный характер и поэтому при заимствовании соответствующих разделов из ранее изданных книг и статей автор стремился по мере возможностей передать дух и пафос авторского изложения – разумеется с необходимыми в подобных случаях согласованиями обозначений, сокращениями, дополнениями и комментариями учебного характера, а иногда и откровенно критическими замечаниями с позиций практической значимости предлагаемых алгоритмов.

Книга состоит из четырех частей. В первой части изложен материал, содержащий основы теории выбора вариантов из заданного множества альтернатив при различных типах неопределенностей. Рассмотрены задачи выбора в условиях неопределенности «среды» (принятие решений в условиях риска, в условиях полной неопределенности, в игровых ситуациях выбора), а также задачи выбора решений в условиях неопределенности цели – многокритериальные задачи выбора. Изложение сопровождается многочисленными модельными примерами, позволяющими легко оценивать ситуации на интуитивном уровне и облегчающими усвоение материала. Рассмотрение всех основных подходов и методов дано с алгоритмических позиций, позволяющих оценить в первую очередь практическую значимость обсуждаемого материала. Сложные математические модели в первой части почти не используются, что позволяет предъявлять минимальные требования к предварительной подготовке читателей.

Во второй части представлены методы поиска минимума (или максимума) вещественной функции от конечного числа вещественных переменных. Даны также элементы теории конечномерной оптимизации, играющей важнейшую роль при решении задач компьютерного моделирования. Представленные здесь методы и алгоритмы в контексте общей проблемы выбора или принятия решений трактуются как «тривиальные» задачи выбора, так как традиционные «неопределенности» отсутствуют (имеем детерминистские однокритериальные задачи выбора). Необходимость в методах оптимизации возникает непосредственно, когда какой-то результат может быть достигнут с помощью различных способов, описываемых конечными наборами вещественных чисел и основная задача заключается в поиске наилучшего в каком либо смысле варианта достижения цели. Кроме того, как показывается в первой части, общие задачи принятия решений при «раскрытии» соответствующих неопределенностей также сводятся в алгоритмическом плане к некоторой цепочке однокритериальных детерминистских задач. Таким образом, можно сказать, что теория конечномерной оптимизации является алгоритмическим фундаментом общей теории выбора.

Существует много публикаций, посвященных проблемам конечномерной оптимизации, а также программных разработок, пакетов и систем оптимизации. В частности, широкую и заслуженную известность получил пакет NAG (разрабатывается и поддерживается Numerical Analysis

Group), где реализованы достаточно мощные и эффективные алгоритмы конечномерной оптимизации.

Однако основную роль в развитии теории и методов оптимизации играли и играют в первую очередь математики. Поэтому в литературе, как отечественной, так и зарубежной, наиболее полно представлены лишь те вопросы теории (и практики!), где можно было получить математически завершенные результаты. Последнее заставляло исследователей ограничиваться наиболее удобными с математической точки зрения структурами и формализациями, связанными, в частности, с концепциями выпуклости, линейности и т.д. С другой стороны, практика реального компьютерного моделирования и численного эксперимента в различных областях часто выдвигает задачи, не обладающие указанными свойствами и при решении которых традиционными методами возникают значительные вычислительные трудности. Если условно назвать категорию исследователей, занимающихся реальным компьютерным моделированием, инженерами, то уместно вспомнить известный тезис: «Математики делают то, что можно, так как нужно, а инженеры - то, что нужно, так как можно».

Во второй части книги мы попытались рассмотреть «то, что нужно» - вопросы, оказывающиеся существенными при проведении реальных компьютерных вычислений. Наряду с важнейшими из широко известных алгоритмических средств здесь представлены также некоторые нетрадиционные вычислительные технологии, оказывающиеся, на наш взгляд, достаточно полезными в большом числе практических ситуаций.

Для чтения основного материала второй части требуются знания в области численного анализа, линейной алгебры, теории матриц и отдельных разделов курса математики, читаемых в технических и экономических вузах. Читатели, уже знакомые со стандартными методами конечномерной оптимизации, смогут найти некоторые нетрадиционные оценки этих методов и возможные альтернативные подходы к решению практических задач.

Третья часть книги посвящена введению в экспертные системы принятия решений и затрагивает вопросы, связанные с выбором решений в трудно формализуемых областях, где стандартные математические технологии моделирования оказываются практически неприменимыми. Таким образом, в отличие от рассмотренных ранее технологий моделирования и принятия решений, экспертные системы имеют дело со слабо структурированными задачами, в которых, в частности, трудно ожидать наличия достоверных численных оценок различных вариантов. Вообще, под экспертной системой здесь понимается программная диалоговая система, в которую включены знания специалистов о некоторой конкретной проблемной области и которая в этой обычно достаточно узкой и специализированной области способна принимать обоснованные решения, заменяя труд высококвалифицированных экспертов.

Последняя часть книги содержит пользовательское описание двух работающих систем поддержки принятия решений, разработанных под руководством автора и основанных на изложенных в книге концепциях. В первой системе реализована оригинальная технология многокритериального выбора вариантов на основе метода t-упорядочения. Вторая система разработана с применением нейлоровской концепции построения диагностирующих байесовских экспертных систем. Указанные системы были разработаны, соответственно, А.В.Рассохиным и С.А.Кудаковым в качестве магистерских диссертаций при их обучении на кафедре «Информационные и управляющие системы» факультета технической кибернетики С.Петербургского государственного технического университета (бывший Политехнический институт имени Петра Великого). Более подробную информацию об этих системах и условиях доступа к ним можно найти на сайте университета.

Книга предназначена для различных категорий читателей. Во-первых, это студенты вузов и других учебных заведений, занятые изучением дисциплин, связанных с современными информационными технологиями и компьютерным моделированием. Во-вторых, это уже дипломированные специалисты, работающие в фирме или на производстве и желающие оценить возможности компьютерной поддержки для своих внутренних проблем. И наконец, это современные руководители, желающие применить в своей работе достижения из указанной области. В частности, знание основных результатов и принципов теории принятия решений и оптимизации позволит им не только лично руководствоваться ими, но и выдавать обоснованные задания своему системному аналитику или отделу системного анализа фирмы.

В настоящее время можно указать большое число предметных областей и практических ситуаций, когда выбор решения может и должен основываться на излагаемых в данной книге методах и технологиях. В частности, теория выбора и принятия решений, а также теория оптимизации могут быть использованы в таких областях как:

задачи выбора оптимальной номенклатуры товара в торговых и иных организациях;

задачи выбора персонала в фирме (например, при приеме на работу);

задачи рациональной организации разработки программного обеспечения для компьютерных систем;

задачи, решаемые в риэлтерских фирмах, оказывающих услуги населению на рынке недвижимости (например, подбор квартир);

задачи оптимального выбора параметров (числовых характеристик) какой либо системы (или организации) – пректируемой или реально существующей;

формирование оптимальных стратегий поведения на рынке ценных бумаг;

задачи принятия решений на финансовом рынке в условиях риска и неопределенности;

задачи максимизации доходов в условиях аукционных торгов и т.д.

Количество соответствующих примеров может быть существенно увеличено. Автор выражает благодарность участникам научного семинара по проблемам принятия решений И.И. Селину и А.В.Рассохину за подготовку текстов описаний систем в четвертой части книги.

Введение

Как показывает практика (и мы продемонстрируем это на реальных примерах), широко бытующее мнение о том, что достаточно иметь хорошее программное обеспечение (ПО) из соответствующей области (а оно обычно есть), чтобы с успехом приступать к решению практических задач, оказывается принципиально неверным. В простейших случаях (например «проблемы», решаемые бухгалтерами) трудностей может и не быть, но в таких алгоритмически сложных областях, как принятие решений, управление, системное проектирование и т.д., ситуация совершенно иная.

Наличие хорошего ПО в соответствующей организации или фирме и хороших аппаратных средств – это лишь необходимое, но не достаточное условие. Кроме этого совершенно обязательной является высокая профессиональная подготовка лица, принимающего решение (ЛПР). Это не обязательно глава фирмы, это может быть специальный человек (так называемый системный аналитик) или группа лиц - отдел системного анализа. Указанное замечание относится не только к области принятия решений, но и к другим областям компьютерного моделирования, требующим привлечения нетривиальных математических моделей, лежащих в основе любых современных информационных технологий.

Приведем характерный пример, иллюстрирующий справедливость вышеприведенных замечаний. Мы выбрали для критики не очень свежую публикацию, однако и по сей день казусы, подобные излагаемым ниже, регулярно встречаются в практической работе, когда пренебрегают продекларированными выше (и ниже!) простыми принципами.

Рассматриваемая задача и ее решение взяты из книги Д. Табака и Б.

Куо (Оптимальное управление и математическое программирование.– М: Наука, 1975 г.; оригинал вышел в 1971 году в США). Авторы – профессора Коннектикутского и Иллинойского университетов, соответственно; перевод выполнен под редакцией Я.З. Цыпкина.

Рассматривается управляемая система второго порядка с одним управлением. Система описывается следующими разностными уравнениями состояния:

$$y_1(i+1) - y_1(i) - T_{i+1}(-y_1^2(i) + y_2(i) + u(i)) = 0$$
$$y_2(i+1) - y_2(i) - T_{i+1} y_1(i) = 0, \quad i = 0, 1, ..., N-1,$$

где $y_j(i)$ – j-я компонента вектора состояния в дискретный момент t_i; $T_i = t_i - t_{i-1}$.

Задача заключается в выборе такой сеточной функции $u(i)$, чтобы перейти из заданного начального состояния

$$y(0) = (0, 1)$$

в целевую область

$$[y_1(N) - 10]^2 - y_2^2(N) - 1 \le 0$$

при минимуме показателя качества

$$J = \sum_{i=1}^{N} \left[y_1^2(i) + y_2^2(i) + 0.1u^2(i-1) \right]$$

и выполнении ограничений:

$$0 \le u(i) \le 1, \ \ \mathrm{i} = 0, 1, ..., N-1;$$

$$y_j(i) \ge 0, \ \ i = 1, ..., N; \ j = 1, 2.$$

При N = 12 (число периодов дискретизации) сформулированная задача естественным образом представляется как стандартная хорошо изученная задача нелинейного программирования с функционалом J, 48 переменными и 37 условиями. Здесь искомыми считаются все величины $y_j(i)$, $u(i)$, T_i. Для решения подобных задач нелинейного программирования разработаны «эффективные» методы и реализующие их программные системы оптимизации. Остается только воспользоваться ими, что авторы книги и проделали. Они использовали хорошо зарекомендовавший себя метод и программу последовательной минимизации без ограничений, разработанные известными американскими авторами Фиакко и Мак-Кормиком (метод штрафных функций). Полученные результаты представлены в виде таблицы и графика, которые должны убедить читателя в эффективности применяемого подхода. Приведем фрагмент полученной таблицы:

Таблица В.1.1

i	T_i (сек.)	t_i (сек.)	u (i–1)	y_1 (i)	y_2 (i)
1	0.00291	0.00291	0.0553	0.00265	0.964
2	0.00196	0.00487	0.5000	0.00375	0.930
3	0.00176	0.00663	0.5000	0.00434	0.898
4	0.00169	0.00832	0.0466	0.00467	0.867
5	0.00164	0.00996	0.0435	0.00479	0.839
6	0.00162	0.01158	0.0419	0.00473	0.812
7	0.00159	0.01317	0.0406	0.00480	0.786
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
12	13.18224	13.20211	0.0339	9.26978	0.683

Далее авторы книги занимаются анализом и обсуждением результатов, совершенно не замечая (вместе с переводчиком книги на русский язык и редактором перевода), что задача фактически не решена и ими получен случайный набор чисел. Действительно, согласно разностным уравнениям системы, переменная $y_2(i)$ должна монотонно возрастать:

$$y_2(i + 1) - y_2(i) = T_{i+1}y_1(i) > 0.$$

Однако в таблице $y_2(i)$ монотонно убывает, т.е. в полученных результатах не удалось отразить даже качественные характеристики решения. При этом использовалась передовая по тем временам вычислительная техника (IBM-7094) и прекрасное программное обеспечение.

Книга содержит целый ряд других неверно решенных задач. Излишне говорить, что реализация на практике рекомендаций, полученных в результате подобных «исследований», может иметь крайне нежелательные, если не катастрофические последствия. Ведь рассматриваемые математические модели могут относиться к потенциально опасным реальным системам. Достаточно сказать, что в данной книге рассматриваются такие модели, как модель управления процессом отравления ксеноном в ядерных реакторах, модель управления ракетным ядерным реактором и т.п. Излишне также говорить, что примененные авторами обсуждаемой книги методы и технологии являются абсолютно корректными и с их помощью были успешно решены и решаются в настоящее время многие задачи компьютерного моделирования. Речь идет не о корректности применяемых методов , а о корректности их применения.

Таким образом, видимая тривиальность вычислительных задач моделирования вообще и задач принятия решений – в частности, а также наличие хорошо развитого современного программного обеспечения не дает оснований отказываться от привлечения к соответствующей деятельности хорошо подготовленных и квалифицированных системных аналитиков. Данная книга и предназначена в первую очередь для начинающих системных аналитиков в области систем оптимизации и принятия решений.

Чтобы наглядно очертить тот круг задач, которые с разной степенью подробности будут затрагиваться в данной книге, рассмотрим несколько максимально упрощенных примеров из различных областей человеческой деятельности, которые можно трактовать как задачи принятия решений.

При этом под задачей *принятия решений* мы будем понимать задачу *выбора* наилучшего способа действия из некоторого множества допустимых вариантов. Дадим более точную формулировку.

Задано множество вариантов X (конечное или бесконечное). Выбор какого-либо из вариантов $x_i \in X$ приводит к некоторому исходу $y_j \in Y$, где Y – множество возможных исходов. Требуется выбрать такой x_i, чтобы получить наиболее благоприятный в определенном смысле исход y_j. Множество вариантов X часто называется также множеством альтернатив (хотя это противоречит канонам русского языка: альтернатив может быть только две). Мы также будем использовать термин альтернатива в указанном смысле.

Пример 1.Чтобы попасть из пункта A (остановка автобуса) в пункт B (лодочная станция) (Рис. В..1), человек должен пройти вначале по асфальтовой дороге (отрезок Ax), а затем по пляжу (отрезок xB). Известны скорости передвижения по асфальтовой дороге и по песку. Спрашивается, в каком месте необходимо свернуть с асфальтовой дороги, чтобы затратить меньше времени на весь путь.

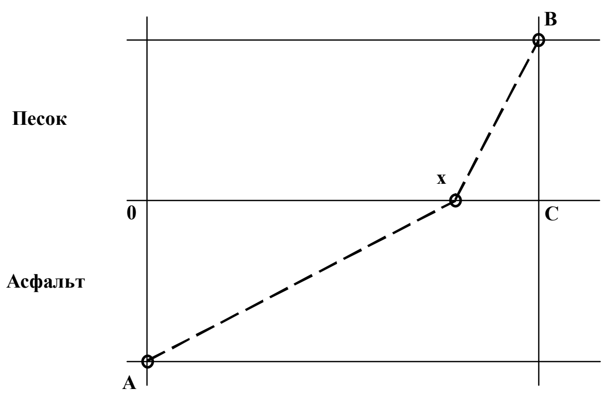


Рис. В.1.1

Сформулированную задачу можно рассматривать как задачу принятия решения: множество альтернатив состоит из множества точек прямой *OC*, т.е. из множества вещественных чисел x. Каждому решению соответствует исход или результат – маршрут *AxB*. Таким образом, имеем задачу **принятия решения в условиях определенности**. Каждый исход (т.е. маршрут) оценивается числом – временем передвижения по маршруту.

<u>Пример 2</u>. Предположим, что при разработке некоторой логической электронной схемы нас кроме функциональных требований интересуют два показателя: потребляемая схемой мощность (f_1) и время задержки распространения сигнала (f_2), причем мы хотим минимизировать оба эти показателя. В наших возможностях варьирование параметров (номиналов) части резистивных элементов схемы $R_1, \ldots, R_L$ в некоторых заданных границах. При этом каждому фиксированному набору $R = (R_1, \ldots, R_L)$ этих параметров соответствует определенное значение f_1 потребляемой мощности и значение f_2 времени задержки. Таким образом, взяв за альтернативы наборы значений R, а затем в качестве исходов – соответствующие им пары чисел (f_1, f_2), приходим к задаче выбора решения в условиях определенности. Изобразив все возможные пары чисел (f_1, f_2) на плоскости, получим некоторую область *F*, каждая точка которой представляет собой один из возможных исходов (Рис. В.).

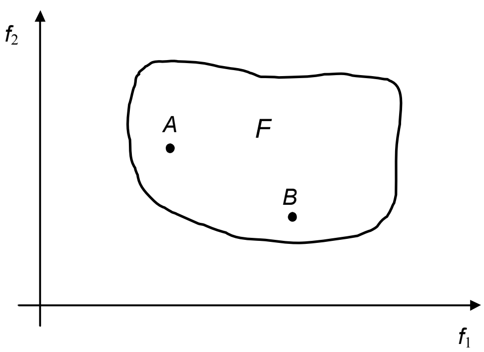


Рис. В.1.1

Так как для принятия решений в условиях определенности выбор альтернативы равнозначен выбору исхода, то принятие решения состоит здесь в выборе конкретной точки множества *F*. Какую точку надо взять в качестве оптимальной в данном случае?

По сравнению с примером 1 это уже более трудная задача, так как при наличии не одного, а двух показателей, оценивающих исход, ответить на вопрос, какое решение является наилучшим, гораздо сложнее. Например, в точке A (см. рис. В.1.2) значение показателя f_1 лучше (меньше), чем в точке B, но зато в точке B лучше значение показателя f_2. Какую из них предпочесть? В данном случае речь идет не столько о том, как найти оптимальное решение, сколько о том, что следует понимать под оптимальным решением, т.е. мы сталкиваемся здесь с трудностями не технического, а концептуального характера.

Пример 3. Студент факультета технической кибернетики, войдя в трамвай, решает, брать ли билет. Здесь исход этого решения определяется двумя обстоятельствами: его решением и фактом появления контролера. Таким образом, студент выступает здесь в качестве принимающего решение, а факт появления контролера – в качестве среды. Имеются всего две альтернативы у принимающего решение и два состояния среды. Как здесь численно оценить «полезности» исходов? проще всего в качестве оценок взять выраженные в условных единицах денежные потери, как указано в табл. В.1.2.

Таблица В.1.2

Альтернатива	Состояние среды	
	Появится контролер	Не появится контролер
Брать билет	2 у. е	2 у. е.
Не брать билета	8 у. е.	0

Какое следует принять решение, если целью считать минимизацию потерь? Это пример задачи принятия решений в условиях неопределенности.

Методы решения подобных задач существенно зависят от наличия дополнительной информации, например, о том, можно ли каждому состоянию среды приписать вероятность его наступления или нет. Кроме того принципиальным является вопрос о том, многократным или однократным является производимый выбор.

Пример 4 (дилемма заключенного). Арестованы два подозреваемых в совершении серьезного преступления. У прокурора нет полного доказательства их вины, и результаты судебного разбирательства дела полностью зависят от стратегии поведения подозреваемых. У каждого из них есть две альтернативы – сознаться в совершении преступления или нет. Возможные исходы представлены в табл. В.1.3 (Н – непризнание, П – признание; 1, 2 – номера задержанных).

Таблица В.1.3

1	2	
	Н	П
Н	(1 ,1)	(10, 0)
П	(0, 10)	(7, 7)

Таблица интерпретируется следующим образом. Если оба арестованных не признаются, то им будет предъявлено обвинение в совершении относительно незначительного преступления (например, связанного с незаконным владением оружием) и оба они получат по 1 году лишения свободы. Если один признается, а второй нет, то первый за выдачу сообщника и помощь в расследовании дела будет полностью освобожден от ответственности, а второй получит полный срок – 10 лет лишения свободы. Если же оба признаются, то оба понесут наказание, но за чистосердечное раскаяние срок заключения будет уменьшен до 7 лет.

Какое решение следует принять каждому из заключенных, чтобы минимизировать наказание?

Здесь, как мы видим, тоже существует неопределенность, но в отличие от предыдущего примера, где присутствовала так называемая «природная» неопределенность, или неопределенность среды, в данном случае мы имеем неопределенность типа «активный партнер». Эффективность решения в такой задаче существенно зависит от стратегии поведения второго лица, а также от информированности обоих субъектов о намерениях другой стороны. Такого типа конфликтные ситуации выбора рассматриваются в разд. 1.3.5.

Пример 5. Существует много примеров, когда лицо, принимающее решение, может указать лишь множество всех тех пар исходов, для которых первый исход в паре предпочтительнее второго. При этом какие-либо численные оценки исходов в принципе отсутствуют. Приведем конкретный пример. Молодой ученый выбирает место своей будущей работы, исходя из следующего множества альтернатив (здесь у.е. - некоторая условная единица, не обязательно совпадающая с конкретной денежной единицей):

1). x_1: ассистент в очень известном университете с окладом 250 у.е.;

2). x_2: доцент в электротехническом институте с окладом 350 у.е.;

3). x_3: профессор в малоизвестном периферийном институте с окладом 450 у.е.

Легко представить себе ситуацию, когда ученый предпочтет x_1 по сравнению с x_2, рассудив, что престиж известного университета и контакты с ведущими специалистами в данной области науки стоят 100 у.е. разницы в окладе. Данное предпочтение можно обозначить (x_1, x_2) или $x_1 \succ x_2$ (x_1

лучше x_2). Точно так же можно предположить, что $x_2 \succ x_3$. И в то же время, сравнивая x_1 и x_3, можно понять и выбор x_3 по сравнению с x_1 (слишком велика разница в окладе). Таким образом, система предпочтений задается множеством пар: (x_1, x_2), (x_2 ,x_3), (x_3, x_1). Следовательно, здесь нет самой предпочтительной альтернативы. Какими принципами следует руководствоваться для принятия решений в подобных ситуациях?

Пример 6. Большой класс практических задач составляют **трудно формализуемые** задачи принятия решений, не имеющие адекватного традиционного математического описания. В качестве примера можно привести задачи медицинской диагностики, в которых по известной исходной информации (результаты анализов, внешние проявления болезни) требуется принять решение о типе заболевания. Такие задачи могут решаться на основе использования специальных программных комплексов – экспертных систем. Понятно, что без применения специальных формализаций здесь оказываются неприменимыми все методы математического анализа (как дисциплины) и необходим особый подход. Важнейшее значение в таких системах принятия решений приобретают проблемы построения исходной базы знаний для конкретной (обычно достаточно узкой) предметной области и процедур логического вывода (правил), позволяющих делать разумные заключения из исходных фактов или утверждений. Характерным примером таких правил могут служить выражения типа: ЕСЛИ (условие) – ТО (действие), например:

ЕСЛИ x за рыночную экономику и радикальные экономические реформы,

ТО x будет голосовать за И.И. Иванова.

Указанный формат записи знаний характерен для важнейшего класса экспертных систем - продукционных экспертных систем, рассматриваемых в данной книге.

Пример 7. Существуют проблемы так называемого группового выбора решений, когда основная задача состоит в том, чтобы указать «справедливые» принципы учета индивидуальных выборов, приводящие к разумному общественному (или групповому) решению. В качестве содержательного примера можно привести заседание военного совета, когда каждый участник заседания высказывает свое мнение относительно плана проведения конкретной операции, а в конечном итоге должен быть выбран один, оптимальный вариант. Как это сделать? Какой результат выбора считать «хорошим», каким свойством он должен обладать? Таким образом, здесь мы, также как и в примере 2, имеем в первую очередь концептуальные трудности: какими показателями должен обладать разумный результат согласований индивидуальных предпочтений?

Простая модель задачи группового выбора формулируется следующим образом. Пусть множество вариантов решений X конечно: $X = \{x_1, x_2, \ldots, x_m\}$. Имеется группа из n членов, принимающих (выбирающих) решение. Каждый член группы с номером $i = 1,\ldots,n$ имеет свою систему предпочтений на множестве X, задаваемую с помощью бинарного отношения $R_i \subset X * X$,

$$R_i = \{(x_j, x_k),\ldots,(x_p, x_m)\}.$$

Здесь R_i – множество упорядоченных пар элементов из X, причем включение некоторой пары (x_s, x_t) во множество R_i означает, что с позиций i-го члена группы вариант x_s предпочтительнее варианта x_t: $x_s \succ x_t$. Требуется по заданной системе $R_1, \ldots, R_n$ индивидуальных предпочтений построить групповую (коллективную) систему предпочтений

$R = f\ (R_1,\ldots,R_n)$, где f – некоторая функция, реализующая принятый принцип согласования индивидуальных предпочтений. Казалось бы, достаточно использовать логически очевидное правило большинства (что обычно и происходит на практике при коллективном решении проблем). Однако существуют принципиальные трудности, связанные с естественными принципами согласования, типа правила большинства или оценивания по среднему баллу. В частности, хорошо известны парадоксы голосования, которые мы продемонстрируем на следующих примерах.

Принятие законопроекта в парламенте.

Пусть три парламентские группы, обладающие приблизительно одинаковым числом голосов, обсуждают три варианта некоторого законопроекта a, b, c с целью утверждения одного «наилучшего» варианта. Пусть системы предпочтений групп имеют соответственно следующий вид:

$$1.\ a \succ b \succ c, \qquad R_1 = \{(a,b), (b,c), (a,c)\}$$

$$2.\ b \succ c \succ a, \qquad R_2 = \{(b,c), (c,a), (b,a)\}$$

$$3.\ c \succ a \succ b, \qquad R_3 = \{(c,a), (a,b), (c,b)\}.$$

Решено действовать по правилу простого большинства. Тогда в результате голосования получим $a \succ b$ потому, что пара (a,b) присутствует в R_1 и R_3, а пара (b,a) – только в R_2. Аналогично устанавливаем, что $b \succ c$ и $c \succ a$, т.е.

$$a \succ b \succ c \succ a.$$

Получаем «порочный круг» и потерю свойства транзитивности в групповом предпочтении. По результатам данного голосования по-прежнему нельзя выбрать наилучший законопроект. Более того, легко видеть, что при умелом ведении заседания парламента спикер может обеспечить утверждение большинством голосов любого из трех вариантов. Действительно, спикер может предложить обсудить вначале какие-то два варианта, проголосовать и худший отсеять. Далее для обсуждения снова останутся два варианта – оставленный при первом рассмотрении и еще не рассматривавшийся. Тогда, очевидно, если на первое обсуждение выносятся варианты a, b, то оказывается $a \succ b$ и вариант b отбрасывается. Далее конкурируют a и c. В результате по принципу большинства имеем $c \succ a$ и в качестве окончательного варианта парламент выбирает вариант c. Если же, напротив, на первое обсуждение вынесем варианты d, c, то в итоге наилучшим окажется a. Точно также можно обеспечить выбор b в качестве наилучшего. Невинное на первый взгляд предложение о порядке рассмотрения оказывает решающее влияние на результат!

Выборы президента (парадокс многоступенчатого голосования).

Допустим, что на выборах президента некоторой компании (или государства) борются две партии, стремящиеся сделать победителем своего представителя. Мы ниже покажем, что при умелом ведении дела меньшинство может навязать свое мнение большинству, хотя голосование всегда будет проводиться по правилу большинства. Чтобы понять идею, достаточно изучить Рис. В..

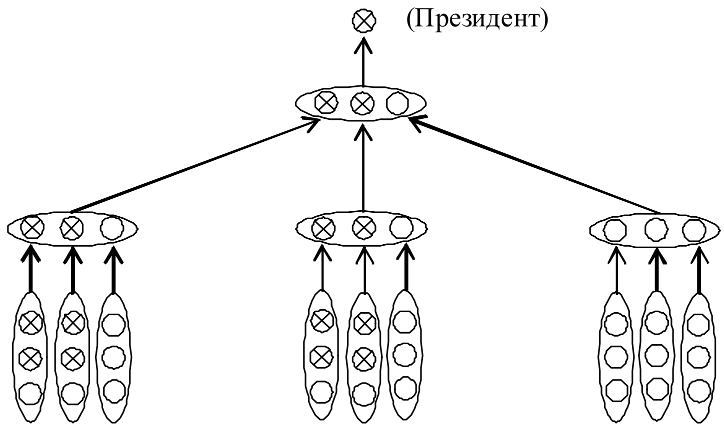


Рис. В.1.3

Из рисунка видно, что группа, владеющая восемью голосами, в итоге навязала свое мнение группе из девятнадцати выборщиков. Все дело, конечно, заключается в умелом группировании сил. Но с помощью современных избирательных технологий это можно реализовать, и это делается повсеместно с помощью целенаправленного вложения средств, организации агитационных поездок в нужные регионы и т.д. Как показал анализ, несколько президентов США в указанном смысле действительно представляли меньшинство в результате реализации системы многоуровневого голосования. При этом, чем больше ступеней, тем ярче проявляется указанный эффект.

И, наконец, последний пример.

Задача распределения ресурсов.

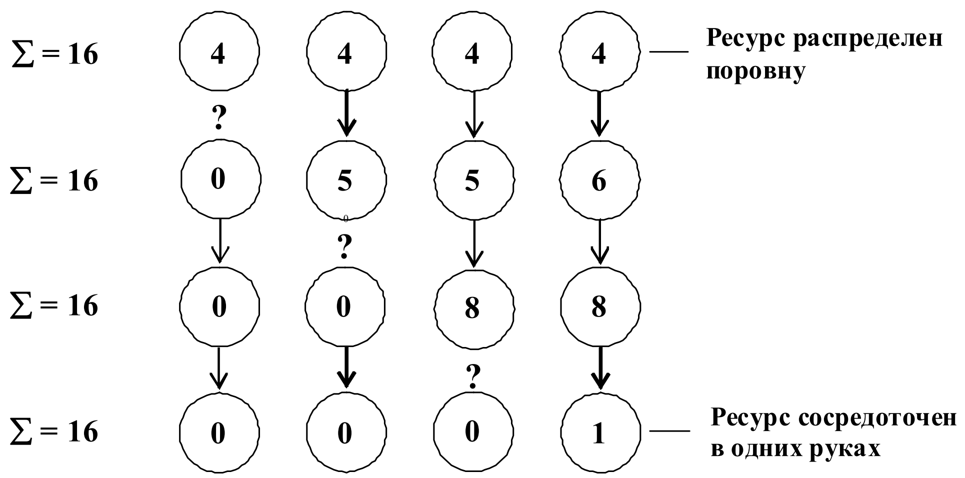
Пусть некоторый ресурс (например, денежный) распределен между n членами некоторого сообщества. При этом состоянием сообщества (системы) будем называть вектор $(a_1, a_2, \ldots, a_n)$, где a_i – объем ресурса, которым владеет i-ый член сообщества. Общий объем ресурса постоянен и равен:

$$a = \sum_{i=1}^{n} a_i \, .$$

Рассмотрим другое состояние той же системы $b = (b_1, b_2, \ldots , b_n)$. Очевидно, состояние «$b$» не хуже состояния «$a$» для i-го субъекта, если $b_i \geq a_i$. Будем теперь производить перераспределение ресурсов на основе очень сильного большинства: переход системы из некоторого состояния «a» в состояние «b» разрешен, если новое состояние будет не хуже старого для всех членов сообщества кроме, быть может, одного (тотально-мажоритарное правило). Последовательность состояний

$$a_1, a_2, \ldots , a_m$$

будем называть тотально-мажоритарным путем из a_1 в a_m, если каждый промежуточный переход их a_i в a_{i+1} был осуществлен на основе тотально-мажоритарного правила. Достаточно неожиданным является утверждение, что тотально-мажоритарный путь может связывать любые два состояния системы! Таким образом, опираясь на мнение «всего общества» можно производить любые перераспределения ресурса, в том числе и представленные на рис. В.1.4.



исходов. Отдельный раздел содержит введение в экспертные системы.

Рис. В.1.4

В книге преследуется цель не только ознакомить читателей с некоторыми общими алгоритмическими принципами функционирования современных автоматизированных систем оптимизации и принятия решений, но и дать средства для разработки собственного программного продукта, реализующего те или иные принципы принятия решений в конкретной (может быть уникальной) практической ситуации.

1 МЕТОДЫ ПРИНЯТИЯ РЕШЕНИЙ

1.1 Задача принятия решений.

1.1.1 Постановка задачи принятия решений. Критериальный язык описания выбора.

Задача принятия решений (ПР) возникает, когда присутствует несколько вариантов действий (альтернатив) для достижения заданного или желаемого результата. При этом требуется выбрать наилучшую в определенном смысле альтернативу.

Общую постановку задачи принятия решений, понимаемой нами как задачу выбора из некоторого множества, можно сформулировать следующим образом.

Пусть X – множество альтернатив, Y – множество возможных последствий (исходов, результатов). X, Y, – вообще говоря, произвольные абстрактные множества. Предполагается существование причинной связи между выбором некоторой альтернативы $x_i \in X$ и наступлением соответствующего исхода $y_i \in Y$. Кроме того, предполагается наличие механизма оценки качества такого выбора – обычно оценивается качество исхода. В некоторых случаях целесообразно полагать, что мы имеем возможность непосредственно оценивать качество альтернативы x_i, и множество исходов по существу выпадает из рассмотрения.

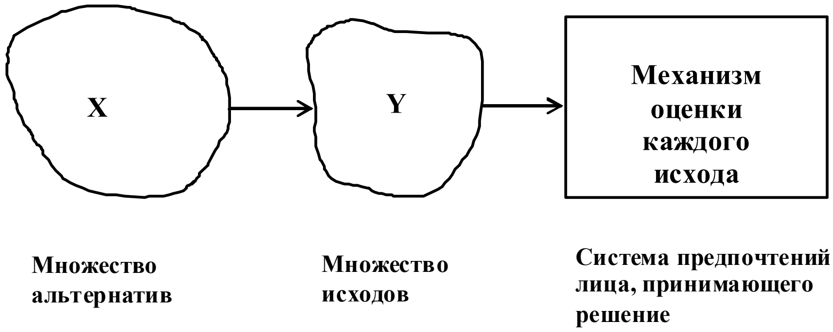


Рис. 1.1

Задача ПР может быть проиллюстрирована с помощью рис. 1.1.

Перейдем к анализу сформулированной задачи ПР.

Первый важный момент заключается в определении **характера связи** альтернатив с исходами. Как мы видели из примеров, эта связь может быть **детерминистской** (или, как часто говорят, детерминированной). В этом случае существует однозначное отображение

$$X \xrightarrow{\varphi} Y, \qquad (1.1.1)$$

т.е. реализуется функция $y = \varphi(x)$, $x \in X$, $y \in Y$. (рис. 1.2).

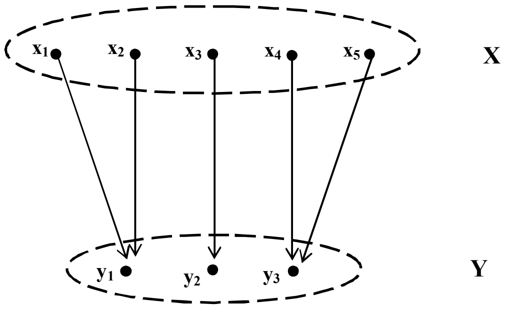


Рис. 1.1

Эта же связь может иметь вероятностный характер, когда выбор x определяет некоторую плотность распределения вероятностей на множестве Y (иногда говорят, что с каждым x связана некоторая лотерея). В этом случае выбор x_i уже не гарантирует наступление определенного исхода y_i, а сама задача ПР называется задачей ПР в условиях риска, (Рис. 1.3).

Графы, представленные на рис. 1.2., 1.3. называются графами связей альтернатив с исходами. Граф, представленный на последнем рисунке, является «взвешенным»: каждая стрелка характеризуется числом P_{ij} – вероятностью наступления исхода y_j при выборе альтернативы x_i. (В общем случае, как было сказано, задается соответствующая плотность распределения).

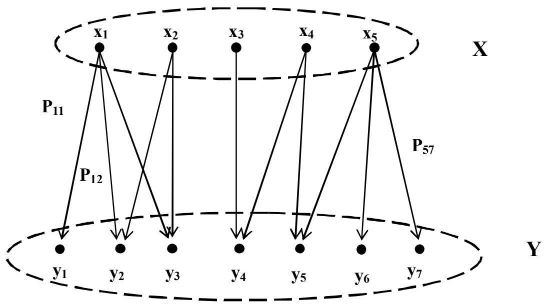


Рис. 1.2

Очевидно,

$$\forall i : \sum_j P_{i,j} = 1. \qquad (1.1.2)$$

Тот же самый рисунок иллюстрирует третий вид связи альтернатив с исходами, который реализуется в задачах ПР в условиях **полной неопределенности**. При этом предполагается, что информация вероятностного характера отсутствует (стрелки на графе не имеют весов).

Как мы видели, неопределенность при выборе и при реализации связи альтернатив с исходами может иметь и другой, возможно более сложный характер (см. «дилемму заключенного»), но мы пока ограничимся указанными тремя случаями, которые могут быть

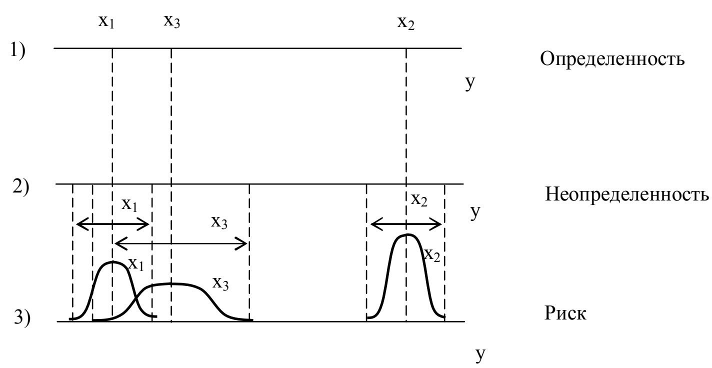


Рис. 1.3

проиллюстрированы также рис. 1.4 При этом случаю 1 соответствует ПР в условиях определенности; точками на оси y обозначены исходы, соответствующие выбору альтернатив x_1, x_2, x_3 (три альтернативы и три определенных исхода). Случай 2 характеризует задачу ПР в условиях неопределенности: после выбора любой из альтернатив x_1, x_2, или x_3 может быть указан лишь интервал расположения соответствующего исхода y. Последний случай 3 отражает ситуацию выбора в условиях риска. Показаны графики соответствующих плотностей распределения вероятностей наступления события y в зависимости от выбора альтернативы x_1, x_2, или x_3.

Заметим, что в каждом из рассмотренных случаев может дополнительно присутствовать свой механизм оценки качества исхода, не связанный непосредственно с механизмом появления y по заданному x. (Здесь предполагается, что числами y закодированы соответствующие исходы, которые могут оцениваться различным образом, например, по нескольким числовым критериям – см. далее).

Второй важный момент в общей задаче ПР состоит в изучении (задании)_**системы предпочтений** лица, принимающего решение (ЛПР). Существенно, что второй момент, по сути, никак не связан с первым и различные способы задания системы предпочтений могут быть реализованы для каждого вида связи альтернатив с исходами.

В некотором смысле простейшая ситуация возникает, когда каждый исход y можно оценить конкретным вещественным числом в соответствии с некоторым заданным отображением

$$f : Y \to R \qquad (1.1.3)$$

В этом случае сравнение исходов сводится к сравнению соответствующих им чисел, например, исход y_i может считаться более предпочтительным, чем y_j, если $f(y_i) > f(y_j)$ (задача максимизации). Исходы эквивалентны, если $f(y_i) = f(y_j)$. Для сравнения самих исходов употребляются выражения:

$$y_i \succ y_j, \; y_i \sim y_j.$$

Такая функция f называется целевой функцией, критериальной функцией, функцией полезности, функцией критерия оптимальности или даже просто критерием оптимальности. Последнее название не вполне корректно, ибо критерий оптимальности – это, вообще говоря, некоторое правило, позволяющее отличать «оптимальные» решения (исходы) от «неоптимальных» и сравнивать исходы между собой. В данном случае это правило связано с заданием целевой функции f. Как известно из математики, однозначное отображение произвольного множества на множество вещественных чисел называется функционалом. Поэтому целевые функции мы часто будем называть целевыми функционалами.

Если предположить, что связь между множеством альтернатив и множеством исходов детерминистская:

$$y = \varphi(x),$$

то функция f, заданная на множестве Y, трансформируется в некоторую функцию J, заданную на X и являющуюся суперпозицией φ и f:

$$J : X \to R, \; J = f \cdot \varphi.$$

В этом случае задача выбора оптимального исхода сводится к задаче выбора оптимальной альтернативы на множестве X и решается непосредственно методами теории оптимизации.

Более реалистичной часто оказывается ситуация, когда в отличие от предыдущего случая «качество» или «полезность» исхода y оценивается не одним числом f (y), а несколькими. Иначе говоря, предполагается, что существует несколько показателей качества решения (критериев), описываемых функциями

$$f_k : Y \to R, \ k = 1, 2, \dots , m,$$

причем каждую из **частных целевых функций** f_i требуется максимизировать (см. пример 2 из введения). Понятно, что в случае многокритериальных оценок исходов возникают существенно более сложные математические модели ситуации выбора, чем в однокритериальном случае. Критерии обычно противоречивы и, как правило, достигают максимумов в различных точках $y \in Y$. Следовательно, возникают не только алгоритмические трудности по решению соответствующих оптимизационных задач, но и чисто концептуальные трудности: что понимать под оптимальным решением в этом случае? Кроме того, здесь появляются уже и несравнимые по векторному критерию $f = (f_1, \dots , f_m)$ варианты y_i, y_j. Более подробно многокритериальные модели принятия решений будут рассмотрены далее.

Ограничиваясь пока указанными выше тремя способами связи альтернатив с исходами и двумя способами описания предпочтений ЛПР на критериальном языке, получим таблицу основных задач выбора

Таблица 1.1

	1 критерий	Много критериев
Определенность	z	Z
Неопределенность	$\tilde{z}$	$\tilde{Z}$

Здесь $z = f(y), \quad f : Y \to R;$

$$Z = f\ (y), \quad f = (f_1, \dots, f_m), \quad f_k : Y \to R, \quad k = 1, 2, \dots, m.$$

Волна сверху означает наличие неопределенности в задаче ПР.

Необходимо отметить, что в настоящее время в приложениях часто применяется именно критериальный язык описания предпочтений, поэтому следующая важнейшая группа проблем – это формирование критериев и целевых функций (функционалов). Эти проблемы, как будет показано, решаются в тесной связи с методами преодоления различных видов неопределенностей на основе тех или иных гипотез.

1.1.2 Описание выбора на языке бинарных отношений. Формальные модели задачи ПР.

Язык бинарных отношений - второй, более общий, чем критериальный, язык описания системы предпочтений ЛПР. Предполагается, что

- отдельный исход сам по себе не оценивается и критериальные функции не вводятся;
- каждая пара исходов y_i, y_j может находиться в одном из следующих отношений: y_i предпочтительнее (строго доминирует) y_j; y_j предпочтительнее y_i; y_i не менее предпочтителен, чем (не строго доминирует) y_j; y_j не менее предпочтителен, чем y_i; y_i эквивалентен y_j; y_i и y_j несравнимы между собой.

Будем далее предполагать, что свои предпочтения пользователь устанавливает в некотором множестве A. В стандартном случае – это множество исходов: A = Y. Однако при детерминистской связи X с Y возможно A = X, или при многокритериальной оценке исходов A = f (Y), f= f_1, ..., f_m. В последнем случае предполагается, что система предпочтений ЛПР задается непосредственно в пространстве векторных оценок исходов. При необходимости можно полагать, что это пространство и есть пространство исходов. В рассматриваемом случае система предпочтений пользователя задается с помощью соответствующего бинарного отношения R на A. Напомним, что бинарным отношением на множестве A называется произвольное подмножество R множества A^2, где A^2 – множество всех упорядоченных пар вида (a_i, a_j), где $a_i, a_j \in A$. Имеем, следовательно, $R \subseteq A^2$, в том числе $A^2 \subseteq A^2$. Основные свойства бинарных отношений (рефлексивность, симметричность, транзитивность, антирефлексивность и т.д.) предполагаются известными.

Существует наглядный способ задания бинарных отношений на конечных множествах, который мы используем в данной книге. Изобразим элементы конечного множества A точками на плоскости. Если задано отношение $R \subseteq A^2$ и $(a_i, a_j) \in R$, где $a_i, a_j \in A$, то проведем стрелку от a_i к a_j. Если $(a_i, a_i) \in R$, то у точки a_i нарисуем петлю-стрелку, выходящую из a_i и входящую в ту же точку. Получившаяся фигура называется ориентированным графом, а сами точки – вершинами графа. Заметим здесь же, что вместо $(a_i, a_j) \in R$ можно писать a_iRa_j.

Основной вопрос заключается в следующем. Пусть на множестве A задана система предпочтений ЛПР в виде бинарного отношения R (часто это отношение строгого доминирования). Что тогда следует понимать под решением задачи выбора? Этот основной вопрос в разных случаях (системах оптимизации и принятия решений, пакетах прикладных программ) решается неоднозначно и пользователям необходимо ясно осознавать, что же конкретно имеется в виду в каждом отдельном случае. Перейдем к точным формулировкам.

Пусть A – заданное множество и R – произвольное бинарное отношение на A. Тогда пара <A, R> называется **моделью выбора**.

<u>Определение.</u> Пусть задана модель <A, R>. Элемент $a^* \in A$

называется наилучшим по R в A, если $(a^*, a) \in R$ при $\forall a \in A \setminus a^*$.

На рис. 1.5а) наилучшими элементами являются a_1, a_2. На графе 1.5б) наилучших элементов нет.

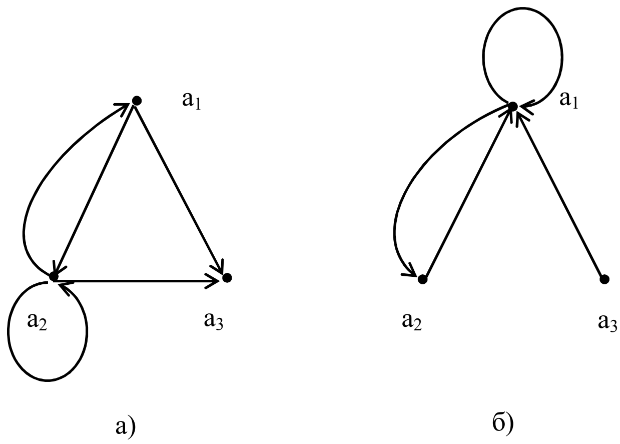


Рис. 1.4

На языке графов понятие наилучшего элемента соответствует наличию вершины, соединенной исходящими из нее стрелками со всеми остальными вершинами графа. При этом могут присутствовать и любые другие дополнительные соединения.

Если предположить, что бинарное отношение, представленное на рис. 1.5б), является отношением предпочтения, и стрелки означают некоторый вариант доминирования, то, по-видимому, целесообразно как-то исключить a_1 из дальнейшего рассмотрения, ибо этот элемент доминируется элементом a_3. С помощью понятия наилучшего элемента это сделать невозможно, хотя в случае, представленном на рис. 1.5а), мы исключили элемент a_3, как не являющийся наилучшим.

Введем вместо наилучшего элемента более слабое понятие максимального элемента.

<u>Определение.</u> Элемент $a^0 \in A$ называется максимальным в модели $\langle A, R \rangle$ или максимальным по R в A, если

$$\forall a \in A : (a, a^0) \in R \rightarrow (a^0, a) \in R.$$

Множество всех максимальных в $\langle A, R \rangle$ элементов обозначим через $Max_R A$.

Очевидно, граф отношения, имеющего максимальные элементы, должен содержать вершины, в которых каждой входящей в нее стрелке (если таковые имеются) соответствует «компенсирующая», выходящая стрелка, направленная в вершину, из которой исходит указанная входящая стрелка.

В примерах, приведенных на рис. 1.5 максимальными по R элементами будут: а) a_1, a_2; б) a_2, a_3.

Легко доказать, что наилучший по R в A элемент является и максимальным. Обратное верно только, если отношение R обладает специальным свойством слабой связности:

$$\forall a_1, a_2 : [a_1 \neq a_2] \to [(a_1, a_2) \in R] \vee [(a_2, a_1) \in R].$$

На рис. 1.5б) отношение R не является слабо связным.

Иногда используется понятие R - оптимального элемента .

<u>Определение.</u> Элемент $a^0 \in A$ называется R - оптимальным на A, если

$$\forall a \in A,\ a \neq a^0 \ \to (a, a^0) \notin R.$$

Иначе говоря, здесь требуется существование вершины, в которую не входит ни одной стрелки.

На рис. 1.5а) R - оптимальных элементов нет, на рис. б) элемент a_3 будет R - оптимальным.

Основным понятием для нас далее будет понятие максимального элемента.

<u>Определение.</u> Множество $\mathrm{Max}_R A$ называется **внешне устойчивым**, если для любого элемента $a \in A \setminus \mathrm{Max}_R A$ найдется такой $a^0 \in \mathrm{Max}_R A$, что справедливо $(a^0, a) \in R$.

Понятие внешней устойчивости оказывается существенным для проблемы ПР. Действительно, если множество $\mathrm{Max}_R A$ внешне устойчиво, то последующий выбор оптимального элемента (на основе, например, привлечения дополнительной информации) может производиться только в пределах множества $Max_R A$. В противном случае, когда устойчивости нет, такой вывод уже не будет иметь разумного обоснования.

Внешне устойчивое множество $Max_R A$ называется **ядром отношения** R в A. Иногда термин ядро применительно к множеству $Max_R A$ используется и без требования внешней устойчивости.

В примерах на рис. 1.5 множества $Max_R A$ являются внешне устойчивыми. На рис. 1.6 представлен случай, когда множество

$$Max_R A = \{a_1, a_2\}$$

не является внешне устойчивым.

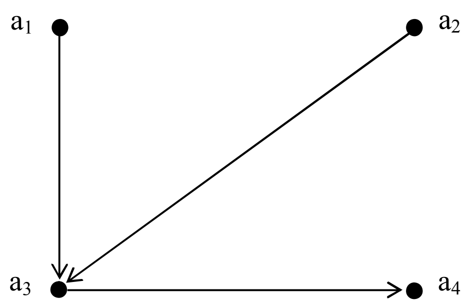


Рис. 1.5

Далее под задачей ПР, сформулированной на языке бинарных отношений, понимается задача выделения ядра - множества максимальных элементов из A по некоторому бинарному отношению R : $A^* = Max_R A$. Специфика конкретных задач ПР находит отражение в задании соответствующего множества вариантов A, а также в формировании бинарного отношения R, характеризующего вполне определенные цели принятия решений в той или иной практической ситуации. Весьма важным с практической точки зрения являются следующие специальные виды бинарных отношений:

- квазипорядок (R – рефлексивно и транзитивно);
- строгий порядок (R – антирефлексивно и транзитивно);
- эквивалентность (R – рефлексивно, симметрично и транзитивно).

1.1.3 Связь различных способов описания выбора. Однокритериальный и многокритериальный выбор.

В данном разделе мы рассмотрим связь критериального языка описания выбора и языка бинарных отношений.

Однокритериальный выбор. Пусть

$$f : Y \to R$$

есть целевая функция, которую требуется максимизировать. Тогда с помощью этой функции на множестве Y индуцируются два бинарных отношения R_1 и R_2:

$$(y_1, y_2) \in R_1 \leftrightarrow f(y_1) \geq f(y_2);$$

$$(y_1, y_2) \in R_2 \leftrightarrow f(y_1) > f(y_2).$$

Отношение R_1 является, очевидно, рефлексивным и транзитивным и, следовательно, определяет квазипорядок на Y. Отношение R_2 обладает свойством антирефлексивности ($\forall y$ неверно, то $f(y) > f(y)$) и транзитивности, являясь строгим порядком. В обоих случаях справедливо равенство:

$$\mathrm{Max}_R Y = \mathrm{Arg\,max}\, f(y), \; i = 1, 2;$$

причем множество максимизаторов функции f является внешне устойчивым в Y. Таким образом, задача максимизации целевой функции f на множестве Y эквивалентна задаче построения ядра одного из бинарных отношений R_1, R_2, совпадающего с множеством максимизаторов f на Y.

Многокритериальный выбор. Предположим теперь, что «качество» или «полезность» исхода оценивается не одним числом, а несколькими. Иначе говоря, предполагается, что существует несколько показателей качества решения, описываемых частными целевыми функциями

$$f_k : Y \to R,\ \ k = 1, 2, \dots , m,$$

которые требуется максимизировать.

В теории многокритериальных задач обычно используются следующие отношения доминирования:

$(y_i , y_j) \in R_p \leftrightarrow \forall k: [f_k(y_i) \geq f_k(y_j)] \wedge\ [f(y_i) \neq f(y_j)]$;

$(y_i , y_j) \in R_s \leftrightarrow \forall k: [f_k(y_i) > f_k(y_j)]$.

Здесь $f = (f_1,\ f_2,\ \dots\ ,\ f_m)$. Отношение доминирования R_p называется отношением Парето, а R_s – отношением Слейтера. Употребляется также запись $(y_i, y_j) \in R_i \leftrightarrow y_i \overset{t}{\succ} y_j$, $t = P, S$.

<u>Определение.</u> Если для некоторой точки $y^0 \in Y$ не существует более предпочтительной по Парето точки, т.е. такой точки y, что $(y, y^0) \in R_p$, то тогда точка y^0 называется **эффективным** или **Парето - оптимальным** решением многокритериальной задачи

$$f_k(y) \to \max,\ \ k = 1, 2, \dots , m;\ y \in Y.$$

Множество, включающее в себя все эффективные элементы множества Y, обозначается P_f (Y) или просто P(Y) (если ясно, о каком векторном критерии f идет речь) и называется множеством Парето для векторного отношения

$$f : Y \to R^m,\ \ f = (f_1, \dots , f_m).$$

Очевидно, $P(Y) \subseteq Y$. Образ множества P (Y) **в пространстве критериев** R^m обозначается P (f). Множество P (f) = f (P(Y)) называется множеством **эффективных оценок**. Множество эффективных оценок называется также множеством Парето в пространстве критериев.

Смысл введенного понятия эффективного решения состоит в том, что оптимальный исход следует искать только среди элементов множества недоминируемых элементов P(Y) (принцип Парето). В противном случае всегда найдется точка $y \in Y$, оказывающаяся более предпочтительной с учетом всех частных целевых функций $f_i(y)$.

Ясно, что бинарное отношение R_p является антирефлексивным, так как $\forall y \in Y$: $(y,y) \notin R_p$. Кроме того, легко установить, что

$$[(y_i, y_j)\in R_p] \wedge [(y_j, y_k)\in R_p] \rightarrow (y_i, y_k)\in R_p .$$

Таким образом, отношение R_p транзитивно. Отсюда следует, что R_p – частичный строгий порядок на Y.

Обычно цель решения многокритериальной задачи

$$f_k(y) \rightarrow \max_{y\in Y}$$

состоит в выделении множества Парето P(Y). При отсутствии дополнительной информации о системе предпочтений пользователя большего сделать нельзя.

Обратимся теперь к отношению R_S.

<u>Определение</u>. Точка $y' \in Y$ называется **слабо эффективным** решением многокритериальной задачи, или решением оптимальным по Слейтеру, если не существует более предпочтительной по Слейтеру точки, т.е. такой точки y, что $(y, y') \in R_S$.

Иначе говоря, исход «y» называется слабо эффективным, если он не может быть улучшен сразу по все m критериям «полезности», задаваемым с помощью частных целевых функций $f_i(y)$, $i = 1, 2, \dots, m$. Множество слабо эффективных элементов обозначается через $S_f(Y)$ или просто S(Y). Очевидно, $S(Y) \subseteq Y$, $P(Y) \subseteq S(Y)$. Аналогично предыдущему случаю вводим обозначение $S(f) = f(S(Y))$.

Введение понятия слабо эффективного решения вызвано, в частности, тем, что в результате многокритериальной оптимизации часто получаются именно эти решения, представляющие, вообще говоря, меньший интерес, чем эффективные решения.

Точно так же, как и в однокритериальных задачах выбора, цель решения многокритериальной задачи может быть сформулирована как задача построения ядра отношения доминирования R_p (отношения Парето). Легко доказать непосредственно, что в этом случае

$$P(Y) = \mathrm{Max}_{R_p} Y$$

с выполнением свойства внешней устойчивости множества Парето.

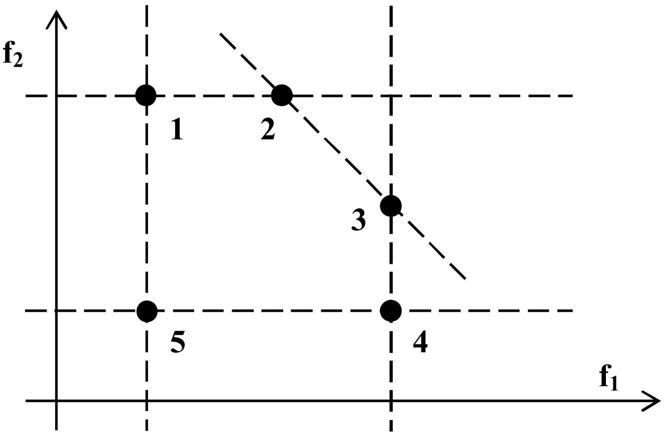


Рис. 1.6

Таким образом, мы видим, что задание целевых функций для оценки качества исходов, как в однокритериальном, так и многокритериальном случае, может порождать различные системы предпочтений, выраженные на языке бинарных отношений.

При этом задача построения ядра оказывается эквивалентной либо задаче построения множества максимизаторов скалярной целевой функции, либо задаче построения множества Парето для векторной целевой функции.

Пример. Пусть задана векторная целевая функция $f = (f_1, f_2)$ на множестве $Y = \{y_1, \ldots, y_5\}$, причем частные целевые функции f_i требуется максимизировать.

$$R_p = \{(y_2, y_1), (y_1, y_5), (y_2, y_5), (y_4, y_5), (y_3, y_5), (y_3, y_4)\}$$

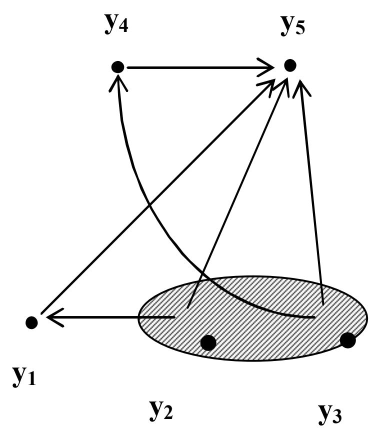


Рис. 1.7

Образы точек y_i в **пространстве критериев** (f_1, f_2) обозначим соответствующими числами (рис. 1.7)

Используя определение доминирования по Парето, можно построить само отношение R_p и его граф (рис. 1.8).

Используя определение ядра, с помощью непосредственного рассмотрения графа получаем:

$$\mathrm{Max}_{R_P} Y = \{y_2, y_3\}.$$

На рис. 1.8 ядро выделено штриховкой. Можно заметить, что наилучшие элементы (см. определение) в данном случае отсутствуют, а понятие максимального элемента позволяет в полной мере формализовать многокритериальную задачу выбора как задачу построения множества недоминируемых по Парето элементов.

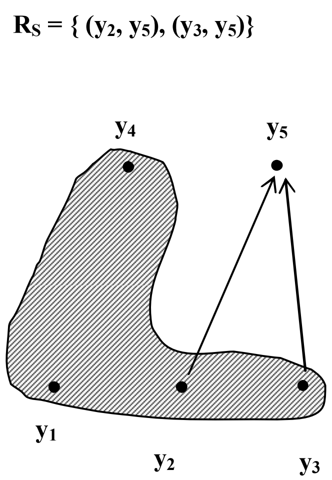


Рис. 1.8

С помощью аналогичных рассмотрений устанавливаем, что отношение Слейтера R_S (так же, как и отношение R_p) является строгим порядком и может быть представлено графически (рис. 1.9). Ядро выделено штриховкой.

Видно, что, во-первых, $R_S \subset R_P$, а во-вторых,

$$\mathrm{Max}_{R_P} Y \subset \mathrm{Max}_{R_S} Y = \{y_1, y_2, y_3, y_4\}.$$

Эти включения выполняются и в общем случае.

<u>Замечание.</u> Отношение Парето R_p порождает соответствующее отношение несравнимости R_H:

$$(y_1, y_2) \in R_H \leftrightarrow [(y_1, y_2) \notin R_p] \wedge [(y_2, y_1) \notin R_p].$$

Для последнего примера имеем, в частности,

$$R_H = \{(y_4, y_2), (y_2, y_3), (y_1, y_4), (y_4, y_1), \ldots\}.$$

Важно отметить, что, например,

$$(y_4, y_2) \in R_H,\ (y_2, y_3) \in R_H, \text{ но } (y_4, y_3) \notin R_H.$$

Таким образом, отношение несравнимости в многокритериальных задачах, являясь симметричным ($(y_i, y_j) \in R_H \rightarrow (y_j, y_i) \in R_H$), не является транзитивным.

1.2 Многокритериальные модели принятия решений в условиях определенности.

Рассматривается следующая модель задачи ПР:

X – множество альтернатив; Y – множество исходов; $f_i : Y \to R$, $i = 1,\ldots,m$ – множество показателей качества (критериев); $\varphi : X \to Y$ – детерминистская функция, отображающая множество альтернатив во множество исходов. Здесь R – множество вещественных чисел.

Таким образом, мы здесь предполагаем, что каждому решению $x \in X$ соответствует единственный элемент $y \in Y$, где $y = \varphi(x)$. «Качество» или «полезность» исхода y, а тем самым и соответствующего решения x оценивается несколькими (m) числами в соответствии с зависимостями f_i. По-прежнему предполагаем, что каждую из функций f_i требуется максимизировать.

С помощью суперпозиции

$$J_i(x) = f_i(\varphi(x)), \quad i = 1, \ldots, m$$

мы имеем возможность непосредственно оценивать качество самого решения x и работать с векторным отображением

$$J : x \to R^m, \quad J = (J_1, \ldots, J_m), \quad J(X) = F \subset R^m.$$

Более того, задание бинарного отношения предпочтения R' на множестве исходов Y индуцирует соответствующее бинарное отношение R'' на множестве X. Именно:

$$(x_1, x_2) \in R'' \leftrightarrow (\varphi(x_1), \varphi(x_2)) \in R'.$$

Соответственно возникает бинарное отношение R''' во множестве оценок $F \subset R^m$:

$$\forall z_1, z_2 \in F : (z_1, z_2) \in R''' \leftrightarrow (y_1, y_2) \in R',$$

где $z_1 = f(y_1)$, $z_2 = f(y_2)$. Поэтому в детерминистском случае (в условиях определенности) отношения предпочтения могут задаваться в любом из указанных трех множеств: X, Y, F. Далее в качестве основного отображения будет рассматриваться отображение

$$J : X \to F \subset R^m,$$

и соответственно системы предпочтений будут задаваться во множествах X , F.

В практических задачах часто непосредственно задается отображение J и, по сути, Y = F, т.е. в качестве исходов выступают сами оценки J_i.

В результате мы приходим к очень распространенной в приложениях многокритериальной модели принятия решений или задаче многокритериальной оптимизации вида

$$J_i(x) \to \max_{x \in X},\ i = 1, ..., m,\ X \subset R^n.$$

Мы здесь сделали еще одно уточнение: $X \subset R^n$. Т.е. мы предполагаем, что все альтернативы или решения параметризованы и каждому из решений соответствует точка $x \in R^n$, $x = (x_1, x_2, ..., x_n)$. И, наконец, вместо обозначений $J_i(x)$ мы снова вернемся к обозначениям $f_i(x)$. Множество X называется множеством допустимых значений и в разделах, посвященных многокритериальным задачам, будет обозначаться через D.

1.2.1 Методы многокритериальной оптимизации.

Рассматривается задача многокритериальной оптимизации вида

$$f_i(x) \to \max_{x \in D},\ f_i : D \to R,\ i = 1, ..., m;\ D \subseteq R^m. \quad (1.2.1)$$

Таким образом, задано m функций или функционалов f_i , отображающих множество D n-мерных векторов $x = (x_1, ..., x_n)$ во множество вещественных чисел R. Здесь предполагается, что выбор оптимальных значений x производится не во всем n-мерном пространстве R^n, а лишь в пределах некоторого его подмножества D. Например, можно интерпретировать задачу (1.2.1) как задачу оптимального выбора параметров x_1, ..., x_n некоторой системы (например, некоторого программного комплекса или перспективного плана развития фирмы), качество функционирования которой оценивается показателями $f_1, ..., f_m$ (см. пример 2 из введения). В этом случае ограничение $x \in D$ отражает наши технологические и иные возможности реализации тех или иных значений x_i. Кроме того, часть ограничений может формироваться на основе имеющейся априорной информации, позволяющей исключить из рассмотрения заведомо неудачные варианты x.

Важнейшее значение при исследовании задач (1.2.1) имеет принцип Парето и связанные с ним понятия эффективного (Парето – оптимального) и слабо эффективного решения. Однако прежде чем перейти к рассмотрению численных методов построения множества Парето, обратимся к традиционным «инженерным» методам многокритериальной оптимизации, сводящим задачу (1.2.1) к некоторой ее однокритериальной версии.

1. Метод главного критерия. В методе главного критерия в качестве целевой функции выбирается один из функционалов f_i , например f_1, наиболее полно с точки зрения исследователя отражающий цель ПР. Остальные требования к результату, описываемые функционалами $f_2, ..., f_m$, учитываются с помощью введения необходимых дополнительных ограничений. Таким образом, вместо задачи (2.1) решается другая, уже

однокритериальная задача вида

$$f_1(x) \to \max_{x \in D'} ; \; D' \subseteq D \subseteq R^n;$$
$$D' = \{ x \in D \, / \, f_i(x) \ge t_i, i = 2, ..., m \}. \qquad (1.2.2)$$

Формально получили более простую задачу поиска максимума функционала f_1 на новом допустимом множестве D'. Добавились ограничения вида $f_i(x) \ge t_i$, показывающие, что мы согласны не добиваться максимальных значений для функционалов f_2, ..., f_m, сохраняя требование их ограниченности снизу на приемлемых уровнях. Важно понимать, что переход от задачи (2.1) к задаче (2.2) вовсе не есть переход от одной эквивалентной задачи к другой. Произошло существенное изменение исходной постановки задачи, которое в каждой конкретной ситуации требует отдельного обоснования. Мы еще вернемся далее к методу главного критерия и его анализу с позиций оптимальности по Парето. Здесь же отметим, что применение этого метода на интуитивном уровне обычно наталкивается на трудности, связанные с возможным наличием нескольких «главных» критериев, находящихся в противоречии друг с другом. Кроме того, не всегда ясен алгоритм выбора нижних границ t_i. Их необоснованное задание может привести, в частности, к пустому множеству D'.

2. Метод линейной свертки. Это наиболее часто применяемый метод «скаляризации» (свертки) задачи (1.2.1), позволяющий заменить векторный критерий оптимальности $f = (f_1, ..., f_m)$ на скалярный $J : D \to R$. Он основан на линейном объединении всех частных целевых функционалов f_1, ..., f_m в один:

$$J(x) = \sum_{i=1}^{m} \alpha_i f_i(x) \to \max_{x \in D}; \alpha_i > 0, \sum_{i=1}^{m} \alpha_i = 1. \qquad (1.2.3)$$

Весовые коэффициенты α_i могут при этом рассматриваться как показатели относительной значимости отдельных критериальных функционалов f_i. Чем большее значение мы придаем критерию f_j, тем больший вклад в сумму (1.2.3) он должен давать и, следовательно, тем большее значение α_j должно быть выбрано. При наличии существенно разнохарактерных частных критериев обычно бывает достаточно сложно указать окончательный набор коэффициентов α_i, исходя из неформальных соображений, связанных, как правило, с результатами экспертного анализа. Позже мы покажем, что, вообще говоря, априори не ясно, в каком отношении должны находиться весовые коэффициенты α_i и α_j, если известно желательное соотношение между f_i и f_j в оптимальной точке (например, мы можем требовать, чтобы $f_i = 0.1\ f_j$).

3. Метод максиминной свертки. Обычно применяется в форме

$$J(x) = \min_i f_i(x) \to \max_{x \in D}.$$

Здесь в отличие от метода линейной свертки на целевой функционал J(x) оказывает влияние только тот частный критерий оптимальности, которому в данной точке x соответствует наименьшее значение соответствующей функции f_i (x). И если в случае (2.3), вообще говоря, возможны «плохие» значения некоторых f_i за счет достаточно «хороших» значений остальных целевых функционалов, то в случае максиминного критерия производится расчет «на наихудший случай» и мы по значению J(x) можем определить гарантированную нижнюю оценку для всех функционалов $f_i(x)$. Этот факт расценивается как преимущество максиминного критерия перед методом линейной свертки.

При необходимости нормировки отдельных частных целевых функционалов, т.е. приведения во взаимное соответствие масштабов измерения значений отдельных $f_i(x)$, используется «взвешенная» форма максиминного критерия:

$$J(x) = \min_i \alpha_i f_i(x) \to \max_{x \in D}, \tag{1.2.4}$$

где весовые коэффициенты α_i удовлетворяют требованиям (1.2.3).

Подбирая различные значения α_i, можно определенным образом воздействовать на процесс оптимизации, используя имеющуюся априорную информацию. Приведем характерный пример.

Пример. (Решение системы неравенств). Весьма часто в задачах оптимального выбора параметров реальных систем (в так называемых задачах оптимального проектирования) технические, экономические и другие требования к проектируемой системе выражаются в виде «условий работоспособности», имеющих форму неравенств вида

$$y_i(x) \le t_i, \quad i = 1, \ldots, m. \tag{1.2.5}$$

Здесь функции $y_i(x)$ интерпретируются как частные показатели качества функционирования системы; $x = (x_1, \ldots, x_n)$ – вектор параметров, подлежащих выбору; t_i – допустимые верхние границы для заданных показателей качества (так называемые контрольные показатели). К форме (1.2.5) очевидным образом приводятся и обратные неравенства $z_k(x) \ge s_k$. Для этого достаточно положить

$$y_k = -z_k,\ t_k = -s_k.$$

Для решения системы неравенств (1.2.5) методами теории оптимизации поступают следующим образом. Вводят так называемые запасы f_j, отражающие степень выполнения каждого из неравенств (1.2.5). Простейшая форма запаса имеет вид

$$f_i(x) = t_i - y_i(x). \tag{1.2.6}$$

Имеем, следовательно, многокритериальную задачу максимизации всех запасов:

$$f_i(x) \to \max_{x \in D}, \; i = 1, \dots, m.$$

Максиминная свертка (максимизируется минимальный из запасов) приводит к следующей однокритериальной задаче:

$$J(x) = \min (t_i - y_i(x)) \to \max, \; x \in D.$$

При наличии весовых коэффициентов имеем задачу

$$J(x) = \min_i \alpha_i (t_i - y_i(x)) \to \max_{x \in D}. \qquad (1.2.7)$$

Весовые коэффициенты α_i в функционале (1.2.7) выполняют функцию нормирования частных критериев по значению. Это можно реализовать, например, следующим образом. Для каждого из ограничений (1.2.5) задаются характерные значения $\delta_i > 0$, определяющие эквивалентные, с точки зрения лица принимающего решения, приращения критериев f_i. Иначе говоря, утверждается, что увеличение критерия f_i на δ_i так же «хорошо», как и увеличение f_j на δ_j. В результате вместо задачи (1.2.7) будем иметь

$$J(x) = \min_i \left(\frac{t_i - y_i(x)}{\delta_i} \right) \to \max_{x \in D}. \qquad (1.2.8)$$

Таким образом, каждая разность $t_i - y_i$ «измеряется» в специальных единицах, определяемых $\delta_i > 0$. В качестве δ_i для нормировки иногда используются значения $f_i(x^0)$ в заданной начальной точке x^0, какие-либо иные «характерные» значения $f_i(x)$ или сами значения t_i, если они не равны нулю. (Подобные соображения могут быть использованы и при выборе весовых коэффициентов в методе линейной свертки).

Достаточно типичным для задач параметрической оптимизации, сформулированных в форме (1.2.5), можно считать случай, когда по условию задачи некоторые из показателей, например $y_1(x)$, нежелательно делать намного меньше, чем t_1. Требуется выполнение соответствующего неравенства (1.2.5), но с небольшим запасом. (Например, в транзисторных устройствах для правильного функционирования схемы может требоваться выполнение заданной степени насыщения транзистора, но значительное ее превышение нежелательно из-за увеличения времени переключения). В таких случаях можно воспользоваться регулирующими свойствами весовых коэффициентов. Именно, вместо задачи (1.2.8) решаем задачу

$$J(x) = \min_i \alpha_i' \left(\frac{t_i - y_i(x)}{\delta_i} \right) \to \max_{x \in D}, \qquad (1.2.9)$$

причем выбираем α_1' много большим, чем α_i' $i = 2, \ldots, m$. Выбор достаточно большого весового коэффициента α_1' приводит к тому, что, с одной стороны, при нарушении первого неравенства

$$y_1(x) \le t_1 \qquad (1.2.10)$$

мы имеем существенное ухудшение целевого функционала (1.2.9), так как разность $t_1 - y_1(x) < 0$, будучи умноженной на α_1', дает большое по абсолютной величине отрицательное число, определяющее значение

$$J(x) = \alpha_1' \frac{t_1 - y_1(x)}{\delta_1}.$$

С другой стороны, уже при незначительных положительных значениях запаса $f_1 = t_1 - y_1(x)$ он будет, сравним с запасами работоспособности по остальным показателям качества. Следовательно, увеличение α_1' вносит некоторый стабилизирующий фактор. В результате соответствующее условие работоспособности с высокой вероятностью будет выполнено, имея в то же время небольшой положительный запас в оптимальной точке.

1.2.2 Максиминные стратегии.

Вернемся к введенным в п. 1.1.3. понятиям эффективного и слабо эффективного решения многокритериальной задачи

$$f_i(x) \to \max_{x \in D}, i = 1,...,m; D \subseteq R^n.$$

Напомним, что в словесной формулировке эффективность решения $x^0 \in D$ означает, что оно не может быть улучшено по какому-либо показателю f_i без ухудшения ситуации по оставшимся показателям. Иначе говоря, если x^0 – эффективно (Парето - оптимально), то не существует других решений $x' \in D$, для которых справедливо:

$$f_i(x') \geq f_i(x^0), i = 1, \ldots, m, \qquad (1.2.11)$$

где хотя бы одно из неравенств (1.2.11) – строгое. И аналогично под слабо эффективным решением мы понимаем решение, которое не может быть улучшено одновременно по всем показателям. На рис 1.10 слабо эффективным решениям соответствуют «северная, северо-восточная и восточная части границы» множества достижимости f (D) (f (D) – это образ множества D для векторного отображения $f = (f_1, \ldots, f_m)$).

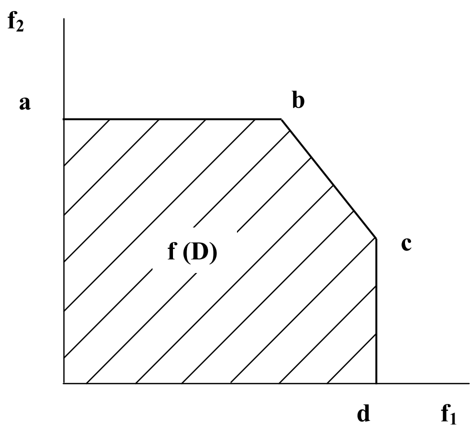


Рис. 1.9

Иначе говоря, в данном примере множество слабо эффективных оценок S(f) совпадает с множеством [a, b] ∪ [b, c] ∪ [c, d]. Множество P (f) эффективных оценок равно [b, c] и совпадает с «северо-восточной границей» множества f (D).

Основная задача данного и следующего разделов заключается в выяснении тех вычислительных средств, которые можно было бы использовать для построения аппроксимации множеств эффективных и слабо эффективных решений и оценок.

Справедлива следующая теорема.

Теорема. Пусть заданы произвольные числа $\alpha_i > 0$, $i = 1, \ldots, m$. Тогда решение задачи

$$\min_i \alpha_i (f_i - t_i) \to \max_{x \in D} \qquad (1.2.12)$$

при любых фиксированных t_i есть слабо эффективный вектор. Наоборот, любой слабо эффективный вектор x^0 может быть получен как решение задачи (1.2.12) при некоторых $\alpha_i > 0$ и $t_i < f_i(x^0)$, $i = 1, \ldots, m$.

Д о к а з а т е л ь с т в о. Прямое утверждение теоремы докажем от противного. Пусть x^0 есть решение задачи (1.2.12) и существует вектор $x' \in D$, для которого

$$f_i(x') > f_i(x^0), \quad i = 1, \ldots, m,$$

что эквивалентно предположению о том, что вектор x^0 не является слабо эффективным. Тогда для любых наборов $\{\alpha_i > 0\}$, $\{t_i\}$ будем иметь (докажите это!)

$$\alpha_i (f_i(x') - t_i) > \alpha_i (f_i(x^0) - t_i), \quad i = 1, \ldots, m$$

и, следовательно,

$$\min_i \alpha_i (f_i(x') - t_i) > \min_i \alpha_i (f_i(x^0) - t_i),$$

а это противоречит предположению о том, что x^0 есть решение задачи (1.2.12). Прямое утверждение теоремы доказано.

Докажем обратное утверждение. Пусть x^0 – слабо эффективный вектор: $x^0 \in S(D)$. Это означает, что не существует другого вектора x', для которого

$$\forall i : f_i(x') > f_i(x^0) \qquad (1.2.13)$$

($\forall i$ – для всех номеров i = 1, …, m).

По условию теоремы заданы такие $\{t_i\}$, что $\forall i : f_i(x^0) - t_i > 0$. Введем числа

$$\alpha'_i = \frac{1}{f_i(x^0) - t_i} > 0$$

и покажем, что

$$\max_{x \in D} \min_i \alpha'_i (f_i(x) - t_i) = \min_i \alpha'_i (f_i(x^0) - t_i) = 1, \qquad (1.2.14)$$

т.е., что при выбранных коэффициентах α'_i максимум реализуется на векторе x^0. Этим самым теорема будет доказана.

Из (1.2.13) следует, что для любого вектора x', отличного от x^0, будет существовать такой номер $i = i_0$, для которого

$$f_{i_0}(x') \le f_{i_0}(x^0) \qquad (1.2.15)$$

(это прямое следствие слабой эффективности вектора x^0). Из (1.2.15) получаем:

$$\alpha'_{i_0}(f_{i_0}(x') - t_{i_0}) \le \alpha'_{i_0}(f_{i_0}(x^0) - t_{i_0}) = 1$$

(напоминаем, что неравенства можно умножать на положительные числа α'_{i_0}). Но тогда и

$$\min_i \alpha'_i (f_i(x') - t_i) \le 1 .$$

Таким образом, доказано, что для любого x', отличного от x^0,

$$\min_i \alpha'_i (f_i(x') - t_i) \le \min_i \alpha'_i (f_i(x^0) - t_i) = 1,$$

а значит и максимум по x левой части последнего неравенства также не будет превышать единицы. Соотношение (1.2.14), а вместе с ним и теорема, доказаны.

Замечание. Если слабо эффективное решение x^0 получено как решение задачи (2.12) при каком-то наборе коэффициентов $\alpha_1, \ldots, \alpha_m$: $\alpha_i >$

0, то, очевидно, это же решение будет достигаться при любом наборе $k\alpha_1$, …, $k\alpha_m$, где k – произвольное положительное число. Поэтому можно считать, что всегда выполнено условие нормировки:

$$\sum_{i=1}^{m} \alpha_i = 1 \tag{1.2.16}$$

В противном случае вместо коэффициентов α_i мы будем рассматривать другие коэффициенты:

$$\overline{\alpha}_i = \alpha_i / \sum_{i=1}^{m} \alpha_i .$$

В силу приведенного замечания будем далее предполагать, что условие (1.2.16) всегда выполнено.

Из доказанной теоремы следуют важные выводы. Будем для простоты считать, что все функционалы f_1, …, f_m исходно положительны, т.е. принимают во всех точках допустимого множества D строго положительные значения: $\forall x \in D : f_i(x) > 0$, i = 1, …, m. Тогда для любого $x^0 \in S(D)$ будет выполнено условие $f_i > t_i$ при $t_i = 0$. Поэтому далее вместо задачи (2.12) будем рассматривать задачу

$$F(x,\alpha) \underline{\underline{\Delta}} \min_i \alpha_i f_i(x) \to \max_{x \in D}; \alpha = (\alpha_1, ..., \alpha_m), x = (x_1, ..., x_n) . \tag{1.2.17}$$

Обозначим множество решений задачи (1.2.17) при фиксированном наборе коэффициентов α через

$$X(\alpha) = \operatorname{Arg} \max_{x \in D} F(x,\alpha) .$$

Согласно доказанной теореме, множество

$$\bigcup_{\alpha \in A} X(\alpha), A = \left\{ \alpha = (\alpha_1, ..., \alpha_m) \middle| \alpha_i > 0, \sum_{i=1}^{m} \alpha_i = 1 \right\}$$

совпадает с множеством слабо эффективных решений:

$$\bigcup_{\alpha \in A} X(\alpha) = S(D).$$

Дадим геометрическую иллюстрацию доказанному утверждению для случая двух целевых функционалов f_1, f_2. Имеем

$$F(x, \alpha) = \min \{\alpha_1 f_1, \alpha_2 f_2\}.$$

Если рассматривать указанную зависимость в пространстве критериев, то получим функцию

$$\Phi (f_1, f_2) = \min \{\alpha_1 f_1, \alpha_2 f_2\}. \tag{1.2.18}$$

Построим линии уровня (линии постоянного значения) функции Ф на плоскости (f_1, f_2). Для этого рассмотрим прямую L, заданную уравнением

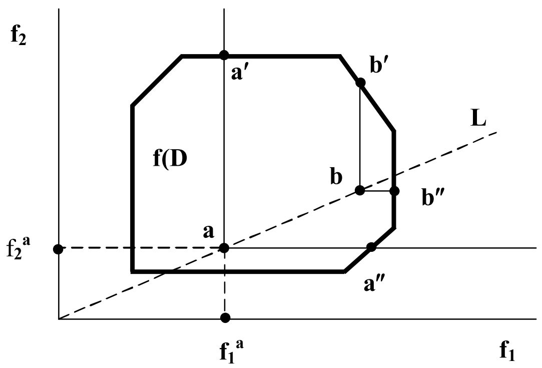


Рис. 1.10

$$\alpha_1 f_1 = \alpha_2 f_2$$

при некотором фиксированном наборе $\{\alpha_1, \alpha_2\}$. График прямой $f_2 = \dfrac{\alpha_1}{\alpha_2} f_1$ показан на рис. 1.11.

В любой точке этой прямой, например, в точке $a = (f_1^a, f_2^a)$, будем иметь $\alpha_1 f_1^a = \alpha_2 f_2^a$. При смещении из точки «а» вправо параллельно оси f_1 получим $\alpha_1 f_1 > \alpha_2 f_2^a$. Аналогичная ситуация наблюдается и при перемещении вверх из точки «а» параллельно оси ординат: будем иметь $\alpha_2 f_2 > \alpha_1 f_1^a$. Поэтому, согласно определению (1.2.18) функции Ф, ее линия уровня, соответствующая значению Ф = $\alpha_1 f_1^a = \alpha_2 f_2^a$, будет совпадать с «уголком» (a' a a'') с вершиной в точке «а», показанным на рис. 1.11. (естественно, что данную линию уровня целесообразно рассматривать только в пределах множества достижимости f(D)). Иначе говоря, во всех точках отрезков [a', a] и [a, a''] функция Ф будет иметь одно и то же значение, совпадающее с ее значением в вершине «уголка», равным по построению $\alpha_1 f_1^a = \alpha_2 f_2^a$.

Легко видеть, что любой «уголок» подобного типа с вершиной, расположенной на прямой L, также будет линией уровня, соответствующей своему значению функции Ф. Причем при удалении вдоль прямой L от начала координат на «северо-восток» мы будем получать линии уровня, отвечающие большим значения Ф. Например, на рис. 1.11. показана линия уровня (b′ b b″), где Ф (b) > Ф (a).

Таким образом, для каждого фиксированного набора весовых коэффициентов $\{\alpha_1, \alpha_2\}$ мы получаем целое семейство «уголковых» линий уровня функции Ф.

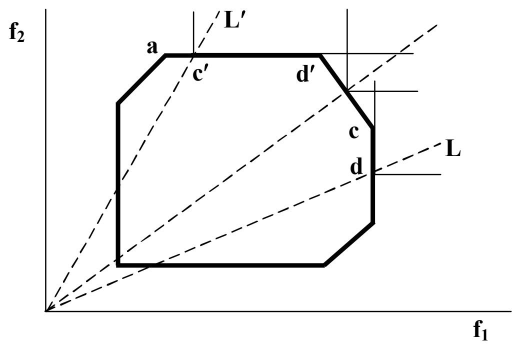


Рис. 1.11

Ясно, что решению основной задачи (1.2.17) будет соответствовать наиболее удаленное от начала координат положение «уголка» (в пределах множества достижимости f(D)), которому соответствует максимально возможное значение функции Ф, а значит и F. На рис. 1.12 показано множество слабо эффективных оценок (отрезок [c, d]), полученных в результате решения задачи (1.2.17). На этом же рисунке показано решение [c′, d′], полученное при другом наборе весовых коэффициентов, соответствующих прямой L′.

Продолжая такие построения, легко убедиться, что, перебирая всевозможные $\alpha \in A$, можно получить «северную, северо-восточную и восточную» части границы множества достижимости f (D):

$$S(f) = [a, d'] \cup [d', c] \cup [c, b].$$

Это и отвечает основному содержанию сформулированной теоремы.

Здесь важно отметить, что задачи оптимизации типа (1.2.17) могут иметь не единственное решение. Так, для α, отвечающего прямой L, мы в качестве решения получим целое множество [c, d] слабо эффективных оценок и соответствующих им слабо эффективных решений исходной многокритериальной задачи. Каждое из этих решений, вообще говоря, должно быть найдено.

Построенные на основе максиминной свертки вычислительные процедуры обычно подразумевают задание некоторой сетки в пространстве весовых коэффициентов A. Далее, для полученного конечного множества наборов весовых коэффициентов

$\alpha^1 = (\alpha_1^1,...,\alpha_m^1);$

....

$\alpha^N = (\alpha_1^N,...,\alpha_m^N)$

решается множество соответствующих однокритериальных задач (1.2.17) или (1.2.12). В результате приходим к построению требуемой аппроксимации множеств S(D) и S(f). Пользователь соответствующей программной системы обычно имеет возможность влиять на указанный процесс, управляя в той или иной мере выбором весовых коэффициентов. Последнее позволяет, в частности, получать более точные аппроксимации отдельных участков границ, представляющих наибольший интерес.

1.2.3 Метод линейной свертки и главного критерия. Лексикографическая оптимизация

Метод линейной свертки уже рассматривался нами на эвристическом уровне (так же, как и метод максиминной свертки), Здесь мы дадим более точные утверждения, касающиеся свойств получаемых решений.

Теорема. Пусть $\alpha \in A$. Тогда решение задачи

$$F_1(x,\alpha)\underline{\underline{\Delta}}\sum_{i=1}^{m}\alpha_i f_i(x) \to \max_{x\in D} \qquad (1.2.19)$$

есть эффективный вектор.

Д о к а з а т е л ь с т в о. Пусть $x^0 \in D$ есть решение задачи (1.2.19) и существует такой $x' \in D$, что $f_i(x') \geq f_i(x^0)$, а для $i = i_0$ имеем $f_{i_0}(x') > f_{i_0}(x^0)$. Тогда

$$\sum_{i=1}^{m}\alpha_i f_i(x') > \sum_{i=1}^{m}\alpha_i f_i(x^0)$$

и, следовательно, x^0 не максимизирует функционал F_1. Полученное противоречие доказывает, что точки x' со сформулированными выше свойствами не существуют и поэтому x^0 – эффективный вектор. Теорема доказана.

Замечание. Обратное утверждение без дополнительных предположений неверно. Существуют эффективные векторы, не являющиеся решениями задачи (1.2.19). Для доказательства этого утверждения достаточно привести хотя бы один опровергающий пример, что ниже и будет сделано.

Таким образом, согласно доказанной теореме

$$\bigcup_{\alpha\in A} X(\alpha) \subseteq P(D).$$

Здесь

$$X(\alpha)\underline{\underline{\Delta}}\operatorname{Arg}\max_{x\in D} F_1(x,\alpha).$$

Обратимся снова к геометрическим иллюстрациям для $m = 2$. В этом случае

$$F_1(x, \alpha) = \alpha_1 f_1(x) + \alpha_2 f_2(x) = \Phi_1(f_1, f_2),$$

где функция Φ_1 считается определенной в пространстве критериев (f_1, f_2). Построим линии уровня функции Φ_1:

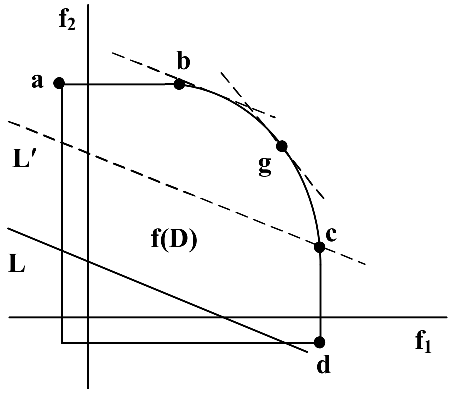


Рис. 1.12

$$\alpha_1 f_1 + \alpha_2 f_2 = \text{const.} \qquad (1.2.20)$$

Набор коэффициентов $\alpha = (\alpha_1,\ \alpha_2)$ считается фиксированным (неизменным в процессе всего рассмотрения). График прямой (1.2.20) на плоскости (f_1, f_2) показан на рис. 1.13.

Угловой коэффициент наклона прямой L определяется числами α_1, α_2 и равен ($-\ \alpha_1/\alpha_2$). При увеличении константы в правой части уравнения (1.2.20) прямая перемещается вверх параллельно L (занимая положение L′). Таким образом, мы имеем целое семейство линий уровня и максимум функции Φ_1, а вместе с ней и F_1 достигается в точках плоскости (f_1, f_2), соответствующих точкам касания (но не пересечения) самой «верхней» линии уровня и множества достижимости f(D). На рис. 1.13 точка b с координатами (f_1^b, f_2^b) реализует найденную рассмотренным методом эффективную оценку. Легко видеть, что ни одна из точек интервалов [a, b), (c, d], соответствующих слабо эффективным, но не эффективным решениям, не может являться точкой касания f(D) и какой-либо линии уровня функции Φ_1 (угловой коэффициент ($-\ \alpha_1/\alpha_2$) не может равняться нулю или бесконечности, так как все $\alpha_i > 0$ и их величина ограничена сверху единицей).

На рис. 1.13 показана точка g, являющаяся решением задачи (1.2.19) при другом наборе α весовых коэффициентов. Перебирая различные α, можно (как и в случае максиминной свертки) получить достаточно точную аппроксимацию множеств P(f) и P(D).

Ситуация, связанная с существованием эффективных решений, не являющихся решениями задачи (1.2.18) ни при каких α, проиллюстрирована на рис. 1.14.

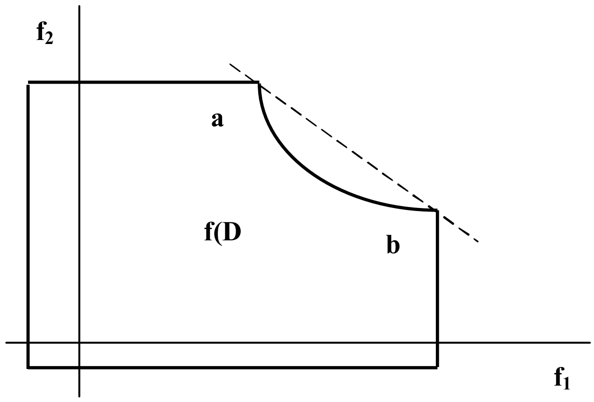


Рис. 1.13

Все точки дуги a, b являются эффективными оценками, но ни одна из них (кроме самих точек a и b) не может являться точкой касания множества f(D) с линиями уровня функции Φ_1 ни при каком наборе коэффициентов α. Таким образом, ясно, что отсутствие выпуклости множества f(D) приводит к принципиальным затруднениям при применении метода линейной свертки. Аналогично наличие строго прямолинейных участков «северо-восточной границы» множества f(D) может приводить к похожим трудностям из-за приближенного характера вычислений (точки внутри таких прямолинейных участков оказываются «неустойчивыми» решениями задачи (1.2.19)).

Замечание. Весьма часто при эвристическом выборе весовых коэффициентов в методе линейной свертки пытаются сразу определить желаемую эффективную точку, исходя из заданных оценок критериев $f_1, \ldots, f_m$ по «важности». Так, например, полагая, что критерий f_2 существенно «важнее» чем критерий f_1, мы бы желали в качестве единственного решения многокритериальной задачи получить эффективную точку «a» на рис. 1.15.

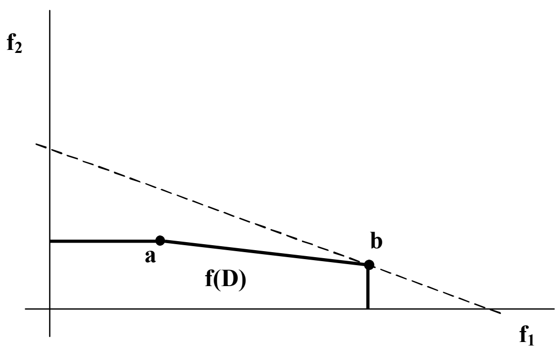


Рис. 1.14

Однако мы не знаем при этом, на сколько коэффициент α_2 должен превышать значение α_1, чтобы была получена именно искомая точка. На рис. 1.15 показана ситуация, когда $\alpha_2 > \alpha_1$ и в то же время найденной точке b соответствуют значения $f_1^b > f_2^b$!

(Мы здесь везде надеемся, что читатель понимает иллюстративный характер приводимых рисунков и графиков. На самом деле, при решении многокритериальных задач графическая информация полностью отсутствует и мы имеем дело с чисто аналитическими постановками соответствующих оптимизационных задач).

Метод «главного критерия» также можно интерпретировать с помощью понятия слабо эффективного решения.

Теорема. Решение задачи

$$f_1(x) \to \max, \; x \in D', \tag{1.2.21}$$

где множество

$$D' = \{x \in D \mid f_i(x) \geq t_i, \; i = 2, \ldots, m\}$$

не пустое, есть слабо эффективный вектор.

Д о к а з а т е л ь с т в о. Пусть x^0 есть решение задачи (1.2.21) и существует $x' \in D$, такой, что

$$f_i(x') > f_i(x^0), \; i = 1, \ldots, m. \tag{1.2.22}$$

Тогда $x' \notin D'$, так как в противном случае это противоречило бы свойству $f_1(x^0) \geq f_1(x)$ для $\forall x \in D'$. Следовательно, $x' \notin D'$ и поэтому существует такой номер $i = i_0$, для которого $f_{i_0}(x') < t_{i_0}$, что противоречит предположению (1.2.22). Теорема доказана.

Теорема. Любой эффективный вектор может быть получен как решение задачи (2.21) при некоторых t_i, $i = 2, \ldots, m$.

Д о к а з а т е л ь с т в о. Пусть $x^0 \in P(D)$; примем $t_i = f_i(x^0)$,

$i = 2, 3, \ldots, m$ и покажем, что

$$f_1(x^0) = \max f_1(x), \ x \in D'. \qquad (1.2.23)$$

Выберем произвольный $x' \in D'$; тогда

$$f_i(x') \geq f_i(x^0), \ i = 2, \ldots, m.$$

Если предположить, что $f_1(x') > f_1(x^0)$, то это будет противоречить свойству эффективности вектора x^0, следовательно, $f_1(x') \leq f_1(x^0)$,что эквивалентно (1.2.23). Теорема доказана.

Из доказанных теорем следует, что в качестве «главного» критерия может быть выбран любой из частных критериев. Независимо от этого выбора произвольное эффективное решение может быть получено как решение задачи (1.2.20) при соответствующем задании t_i.

Метод главного критерия допускает простую графическую иллюстрацию.

Рис. 1.16 отражает предположение, что в качестве «главного» выбран критерий f_1, а на значения функционала f_2 наложено ограничение $f_2 \geq t_2$. Образ $f(D')$ множества D' точек из D, удовлетворяющих указанному дополнительному ограничению, соответствует незаштрихованной части множества $f(D)$.

Максимизация критерия f_1 на множестве D', очевидно, приводит к построению отрезка [a, b] на рис. 1.16. Как видно из рис. 1.16, задавая различные t_2, можно получить аппроксимацию «северо-восточной» и «восточной» частей границы множества $f(D)$, куда входят все эффективные решения задачи.

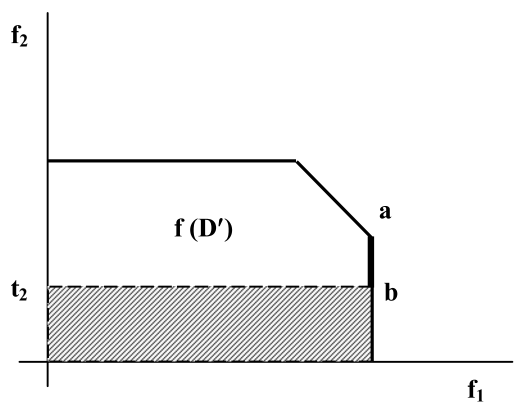


Рис. 1.15

При выборе в качестве «главного» критерия f_2 мы аналогично построим «северную» и «северо-восточную» части границы.

Пусть известен диапазон изменения функционала f_i :

$$f_i^H \le f_i \le f_i^B, \; i = 2, \ldots, m.$$

Тогда из доказательства последней теоремы следует, что соответствующие t_i (при работе с критерием f_1 как с «главным») должны меняться в тех же пределах, пробегая выбранную сетку значений (аналогично построению аппроксимаций множеств эффективных и слабо эффективных решений в методах максиминной и линейной сверток).

<u>Лексикографическая оптимизация</u>. Как мы видели, методом максиминной свертки могут быть получены все слабо эффективные решения, если вектор $\alpha = (\alpha_1, \ldots, \alpha_m)$ пробегает все множество векторов A. Напротив, по методу линейной свертки можно получить только эффективные решения, но не все. А. Джоффриону принадлежит идея выделения эффективных точек из множества S(D), построенного методом максиминной свертки, с помощью процедуры максимизации линейной свертки. В результате будут построены все эффективные и только эффективные решения.

<u>Теорема</u>. Пусть $\alpha_i > 0$, $i = 1, \ldots, m$. Тогда решение задачи

$$\sum_{i=1}^{m} f_i(x) \to \max_{x \in T_\alpha};$$

$$T_\alpha = \operatorname{Arg} \max_{x \in D} \min_i \alpha_i (f_i - t_i) \qquad (1.2.24)$$

есть эффективный вектор. Наоборот, любой эффективный вектор x^0 может быть получен как решение задачи (1.2.24) при некоторых $\alpha_i > 0$ и $t_i < f_i(x^0)$, $i=1,\ldots,m$.

Доказательство теоремы приводить не будем. Обсудим ее смысл и дадим геометрическую иллюстрацию.

Множество T_α при фиксированном наборе коэффициентов α получают методом максиминной свертки. Можно доказать, что, если T_α – непустое множество, то оно обязательно наряду со слабо эффективными решениями содержит хотя бы один эффективный вектор. (Упражнение: докажите последнее утверждение). Далее, максимизируя линейную свертку частных критериев (1.2.24) на множестве T_α, мы получаем в результате эффективное решение. Совершенно очевидно, что, если T_α состоит из одного элемента, то он же и будет эффективным решением и необходимость в максимизации (1.2.24), по существу, отпадает.

На рис. 1.17 показано множество T_α, найденное при определенном наборе коэффициентов α, и выделенная из него эффективная оценка «e».

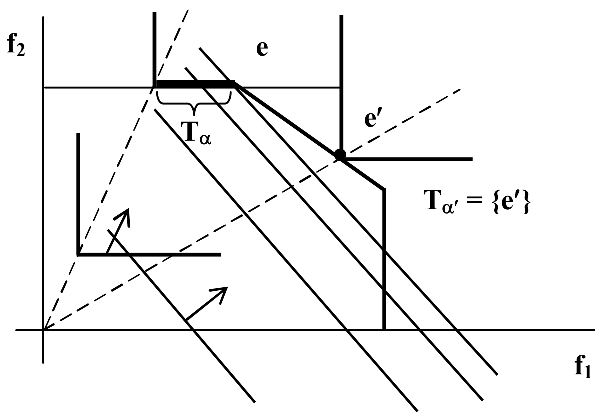


Рис. 1.16

При другом наборе коэффициентов α' множество $T_{\alpha'}$ (см. рис. 1.17) является одноэлементным: $T_{\alpha'} = \{e'\}$. Точка e′ также оказывается эффективной оценкой.

Термин «лексикографическая оптимизация» здесь использован в следующем смысле. Речь идет о так называемом лексикографическом упорядочении двух критериев – максиминного и линейного: вначале «срабатывает» максиминная свертка, и если в результате получен неоднозначный результат (множество T_α содержит более одного элемента), то выбираются тот или те из полученных элементов, которые максимизируют линейную свертку (1.2.24), т.е. только тогда «срабатывает» второй критерий. Данный процесс аналогичен поиску слова в словаре: вначале мы работаем с первой буквой, затем – со второй и т.д. (этим и определяется название).

Задача многокритериальной оптимизации (1.2.24) часто записывается в следующем компактном виде:

$$\left\{ \min_{i} \alpha_i (f_i - t_i), \sum_{i=1}^{m} f_i(x) \right\} \to \max_{x \in D}.$$

Выводы. Ни один из методов, представленных выше, не позволяет выделить единственное оптимальное решение. Решения, соответствующие различным наборам весовых коэффициентов, являются равноправными элементами множеств эффективных и слабо эффективных решений, которые согласно общей постановке задачи ПР реализуют ядра соответствующих бинарных отношений (отношений Парето и отношений Слейтера), т.е. и являются искомыми решениями. Однако с практической точки зрения, например, в задачах выбора вариантов (при покупке или заказе товаров, при выборе партнеров по бизнесу, при выборе вариантов программных средств и т.д.), а также в системах автоматизированного проектирования, часто требуется выбрать единственное решение (проект). Для этого должна привлекаться некоторая дополнительная информация о предпочтениях лица, принимающего решения. Принцип Парето в этом смысле позволяет лишь сузить класс возможных претендентов на решение и исключить из рассмотрения заведомо неконкурентоспособные варианты.

Методы выбора единственного решения многокритериальной задачи существуют и связаны с использованием моделей и процедур, предназначенных для структуризации и количественного описания субъективного мнения лица, принимающего решения. В разделе 1.5 эти вопросы будут рассмотрены более подробно.

1.3 Принятие решений в условиях неопределенности.

Мы ограничимся здесь обсуждением нескольких стандартных ситуаций неопределенности обстановки в процессе принятия решений.

По-прежнему считаем, что задано множество альтернатив X и множество возможных исходов Y. При рассмотрении задачи ПР в условиях определенности, когда каждому решению $x \in X$ соответствовал единственный исход $y \in Y$, было безразлично, на каком множестве X или Y задавать бинарное отношение предпочтения R. Ранее мы обычно задавали это отношение на множестве решений X. При этом на самом деле неявно задавалось и соответствующее отношение на множестве исходов Y. Действительно, пусть существует однозначная зависимость $y = \varphi(x)$, позволяющая по решению x определить единственный исход y. Альтернативы x' и x'', по существу, сравнивались по значениям оценок соответствующих исходов y' и y''. Иначе говоря, имели

$$x' \succ x'' \leftrightarrow y' \succ y'',$$

где использованы обозначения

$$x' \succ x'' \leftrightarrow (x', x'') \in R_X,$$

$$y' \succ y'' \leftrightarrow (y', y'') \in R_Y.$$

Таким образом, общая модель принятия решений $<A, R>$ могла быть сформулирована в виде $< X, R_X >$ либо $< Y, R_Y>$. Суть дела от этого не менялась. В таком случае говорят, что отображение $\varphi : X \to Y$ является <u>гомоморфизмом</u> модели $< X, R_X >$ в модель $< Y, R_Y >$, т.е. для всяких $x', x'' \in X$ из $x' R_X x''$ следует $\varphi(x') R_Y \varphi(x'')$.

При наличии неопределенных факторов ситуация усложняется.

Теперь мы уже не можем гарантировать наступление определенного исхода y при выборе решения x.

Будем считать, что наша система предпочтений связана с оценкой «полезностей» исходов y. Выбор x осуществляется с единственной целью – получить «хороший» исход y, принадлежащий ядру – множеству максимальных элементов из Y по заданному отношению R_Y. В данном случае модель ПР имеет вид $< Y, R_Y >$. В частности, отношение R_Y может быть задано, хотя это не единственный способ, как и раньше, с помощью однокритериальной или многокритериальной системы оценок исходов, т.е. на критериальном языке описания выбора.

1.3.1 Основные понятия

Таблица 1.2

X	Z
	$z_1 \ldots z_j \ldots z_m$
x_1	$y_{11} \ldots y_{1j} \ldots y_{1m}$
…	……………
x_j	$y_{j1} \ldots y_{jj} \ldots y_{jm}$
…	……………
x_n	$y_{n1} \ldots y_{nj} \ldots y_{nm}$

В случае, когда множество альтернатив X и исходов Y конечны, ситуацию выбора альтернативы в условиях неопределенности можно представить с помощью матрицы, называемой матрицей решений (табл. 1.2). Здесь $X = \{x_1, \ldots, x_n\}$, $Y = \{y_{11}, \ldots, y_{nm}\}$. Вектор $Z = \{z_1, \ldots, z_m\}$ описывает неопределенность обстановки и также предполагается конечным. По существу, имеется функция двух аргументов: $y = F(x, z)$, $F : X \times Z \to Y$.

Заданная матрица интерпретируется следующим образом. Если мы выбрали решение x_j, то могут реализоваться различные исходы из соответствующей строки матрицы: y_{j1}, …, y_{jm}. Какой именно исход реализуется, зависит от значения параметра неопределенности z, который может иметь различный содержательный смысл. Будем различать две основные ситуации:

1) вектор Z отражает так называемые «природные» неопределенности, т.е. неопределенность «состояния природы» в момент принятия решения;
2) множество $Z = \{z_1, \ldots, z_m\}$ есть множество альтернатив, на котором (одновременно с нами) осуществляет выбор решения второй субъект, руководствуясь своим отношением предпочтения R_y (неопределенность типа «активный партнер»). При этом выбираемое нами решение x, в свою очередь, характеризует неопределенность обстановки для второго субъекта.

Далее эти вопросы рассматриваются более подробно. Заметим здесь же, что представление задачи ПР с помощью табл. 3.1 имеет достаточно общий характер, в частности, включает в себя случай полной определенности. При этом таблица будет состоять из одного столбца (что эквивалентно наличию одного состояния среды).

В общем случае мы будем предполагать существование функции

$y = F(x, z)$,

где $x \in X$, $z \in Z$, $y \in Y$; X, Z, Y – множества (вообще говоря, абстрактные) альтернатив (решений), состояний среды и исходов, соответственно. Особенностью рассматриваемых ниже задач ПР является предположение о неизвестном в момент принятия решения значении параметра z. Саму функцию F будем называть функцией реализации. Таким образом, функция реализации ставит в соответствие каждой паре вида (x, z), где x – альтернатива, а z – состояние фактора неопределенности, исход y.

При этом, как указывалось, характер неопределенности, описываемый переменной z, может быть различным. Вначале мы будем предполагать, что z описывает некоторую неопределенность среды (неопределенность типа «активный партнер» будет рассмотрена в разд. 3.5). Кроме того, мы будем предполагать, что каждый исход y оценивается вещественным числом e (отражающим, если угодно, «полезность» исхода), и требуется максимизировать эту оценку. (Имеем, следовательно, однокритериальную оценку исхода). Для упрощения обозначений мы в таких случаях будем отождествлять y и e, считая, что исход y уже есть некоторая числовая оценка принятого решения. При этом функция реализации преобразуется в соответствующую вещественную функцию (целевую функцию) J(x,z), которую следует максимизировать или минимизировать по x в зависимости от смысла решаемой задачи.

В указанной ситуации вполне естественно ввести следующее бинарное отношение доминирования R_1 на множестве X:

$$x_1 \overset{R_1}{\succ} x_2 \leftrightarrow \forall z \in Z: J(x_1, z) \geq J(x_2, z),$$

причем хотя бы для одного z имеем строгое неравенство. (Здесь $\succ$ – знак строгого доминирования). Отношение эквивалентности может быть введено с помощью соотношения:

$$x_1 \sim x_2 \leftrightarrow \forall z \in Z: J(x_1, z) = J(x_2, z)$$

($\sim$ – знак эквивалентности).

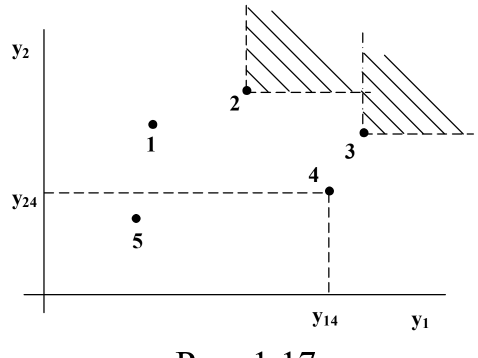


Рис. 1.17

Пример. Рассмотрим числовую матрицу решений, где $Z = \{z_1, z_2\}$, $X = \{x_1, \ldots, x_5\}$. В этом случае каждому решению $x_i \in X$ соответствуют две числовые оценки полезности

$y_{1i} = J(x_i, z_1)$

$y_{2i} = J(x_i, z_2)$,

отвечающие двум возможным состояниям среды. Таким образом, можно рассматривать значения y_{1i}, y_{2i} как координаты точки x_i в пространстве y_1, y_2. Пяти возможным решениям будут соответствовать пять точек в плоскости (y_1,y_2). Абсцисса каждой точки есть результат соответствующего решения при состоянии среды $z = z_1$, а ордината – при $z = z_2$ (рис. 1.18).

Совершенно очевидно, что здесь, также как и в многокритериальных задачах, действует принцип Парето и нас должны интересовать только недоминируемые в смысле отношения R_1 решения x_i. На рис. 1.18 это будут решения x_2, x_3. Остальные решения могут быть отброшены и в дальнейшем не учитываться.

Рассмотренный пример показывает, что все основные принципы и методы паретовского анализа многокритериальных задач переносятся на однокритериальные задачи принятия решений в условиях неопределенности. В частности, на основе принципа Парето исходное множество альтернатив X должно быть сокращено с целью удаления доминируемых по Парето вариантов. В общем случае это может быть выполнено с помощью уже рассмотренных ранее численных методов выделения Парето-оптимальных решений.

Далее мы обычно будем предполагать конечность множеств X, Y, Z и задавать функцию реализации с помощью соответствующей матрицы решений. Однако большинство из рассматриваемых методов допускают обобщение на бесконечномерный случай.

При рассмотрении методов принятия решений в условиях неопределенности используется понятие <u>оценочной функции</u>. Очевидно, что если принятие решений происходит в условиях определенности, то матрица решений будет содержать только один столбец. Принятие решений в условиях неопределенности состоит, по существу, тоже в формировании одностолбцовой матрицы решений и сведении задачи к случаю полной определенности. Эта процедура выполняется неоднозначно с помощью применения различных оценочных функций.

Пусть задана (n×m)–матрица решений $\{y_{ij}\}$. Оценочной функцией называется функция Ψ, преобразующая эту матрицу в одностолбцовую матрицу $\{y_i\}$:

$$y_i = \Psi\ (y_{11}, y_{12}, \ldots, y_{nm}),$$

т.е. y_i зависит от всех элементов исходной матрицы. Многие методы принятия решений имеют оценочные функции вида

$$y_i = \Psi\ (y_{i1}, \ldots, y_{im}),$$

когда i-тый элемент одностолбцовой матрицы зависит только от элементов i-той строки исходной матрицы решений.

Например, оценочная функция Ψ может быть задана с помощью соотношения

$$y_i = \min_j y_{ij} + \max_j y_{ij}.$$

После построения оценочной функции выбор наилучшей альтернативы x* производится из условия максимума или минимума значения оценочной функции (в зависимости от интерпретации элементов матрицы решений – «доходы» или «потери»), например,

$$i^* = \arg\min_i y_i, x^* = x_{i*}.$$

Без существенного ограничения общности можно полагать, что всякое решение в условиях неполной информации – сознательно или неосознанно – принимается в соответствии с некоторой оценочной функцией. Выбор самой оценочной функции – это неформальный акт и этот выбор всегда должен осуществляться с учетом качественных характеристик ситуации, в которой принимается решение.

Далее мы рассмотрим классические критерии принятия решений на основе различных оценочных функций.

1.3.2 Принятие решений в условиях риска.

Напомним, что когда говорят о принятии решений в условиях риска, обычно предполагают, что каждой альтернативе соответствует свое распределение вероятностей на множестве исходов. Если множества альтернатив и исходов конечны, то считаются известными вероятности каждого исхода, возможного при выборе данной альтернативы.

Типичную постановку задачи о принятия решений в условиях риска поясним с помощью конкретного примера.

<u>Пример 1</u>. Задача о замене вратаря.

На последней минуте хоккейного матча при ничейном счете тренер команды должен принять решение о замене вратаря шестым полевым игроком. Статистика, имеющаяся у тренера, показывает, что в аналогичных условиях в предыдущих встречах замена вратаря в одной шестой части случаев привела к выигрышу, в половине случаев – к ничьей и в одной трети случаев – к поражению. Если же вратарь не заменялся, то в 7/8 случаев встреча заканчивалась вничью, а в 1/8 части случаев команда проигрывала.

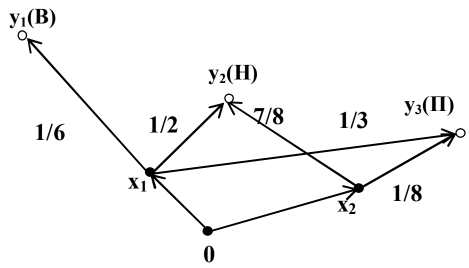


Рис. 1.18

Построим для этой задачи граф связей альтернатив и исходов. Здесь имеются две альтернативы: x_1 – заменить вратаря, x_2 – не делать замены. В любом случае возможны три исхода: выигрыш (В), ничья (Н) и поражение

(П). Принимая за вероятность каждого исхода частоту его появления в предыдущих матчах, получим граф, представленный на рис. 1.19. Решение задачи будет дано несколько позже.

Сформулированная задача ПР в условиях риска и приведенный пример не позволяют пока понять, где же здесь состояние среды? Какой характер имеет функция реализации F(x, z) и возможно ли вообще ее построение? Оказывается, что язык функций реализации является достаточно общим и позволяет описывать различные ситуации неопределенности, в том числе и рассмотренную выше.

Обратимся снова к задаче о замене вратаря. Задание функции реализации означает, что при известном z мы по каждому x уже однозначно определяем исход y. Таким образом, зная состояние среды z, мы должны точно знать, что будет, если мы выберем альтернативу x_1, и каков будет исход при выборе x_2. Введем следующие шесть искусственных «состояний среды»:

$z_1 : x_1 \to$ В,	$x_2 \to$ Н	$p(z_1) = (1/6)(7/8) = 7/48$
$z_2 : x_1 \to$ Н,	$x_2 \to$ Н	$p(z_2) = (1/2)(7/8) = 7/16$
$z_3 : x_1 \to$ П,	$x_2 \to$ Н	$p(z_3) = (1/3)(7/8) = 7/24$
$z_4 : x_1 \to$ В,	$x_2 \to$ П	$p(z_4) = (1/6)(1/8) = 1/48$
$z_5 : x_1 \to$ Н,	$x_2 \to$ П	$p(z_5) = (1/2)(1/8) = 1/16$
$z_6 : x_1 \to$ П,	$x_2 \to$ П	$p(z_6) = (1/3)(1/8) = 1/24$

В правом столбце указаны вероятности соответствующих событий.

Теперь функция реализации может быть задана в виде табл. 1.3. Около каждого состояния среды указана вероятность его появления.

Таблица 1.3

X	Z					
	z_1 (7/48)	z_2 (7/16)	z_3 (7/24)	z_4 (1/48)	z_5 (1/16)	z_6 (1/24)
x_1	В	Н	П	В	Н	П
x_2	Н	Н	Н	П	П	П

Рассмотрим теперь задачу ПР в более общем случае, когда имеется n альтернатив $x_1, \ldots, x_n$ и L исходов $y_1, \ldots, y_L$. В качестве «состояния среды» возьмем множество возможных согласно графу связей альтернатив и исходов отображений $z_j : X \to Y$, $j = 1, \ldots, S$. В случае конечных множеств X и Y будем иметь $S = \prod_{j=1}^{n} S_j$, где S_j – количество стрелок, исходящих из альтернативы x_j, на графе связей альтернатив и исходов (в нашем примере $S_1 = 3$, $S_2 = 2$, $S = 6$). Таким образом, каждое «состояние среды» z_j соответствует такому подграфу графа связей альтернатив и исходов (будем называть его <u>подграфом состояния</u>), в котором из каждой альтернативы x_i исходит только одна стрелка, указывающая, какой исход будет реализован при выборе альтернативы x_i (S – максимально возможное число таких подграфов). Следовательно, выбор «состояния среды» z_j и альтернативы x_i полностью определяет исход – обозначим его через y_j (x_i). Далее, каждому состоянию среды z_j соответствует вероятность его наступления (вероятность реализации соответствующего подграфа состояния):

$$p(z_j) = \prod_{i=1}^{n} p_i(y_j(x_i)), j = 1,\ldots,S,$$

где p_i $(y_j$ $(x_i))$ – заданная вероятность наступления исхода y_j при выборе альтернативы x_i. Таким образом, для вычисления p (z_j) достаточно перемножить числа, стоящие около стрелок, составляющих подграф состояния z_j. Теперь таблица, представляющая функцию реализации, уже может быть построена.

Установленная выше возможность представления задачи ПР в условиях риска в форме функции реализации означает, что статистическую неопределенность, проявляющуюся в неоднозначной (вероятностной) связи между средством и результатом, всегда можно интерпретировать как существование некоторой среды, оказывающей влияние на результат. Методологическое значение этого факта состоит в том, что достаточно широкий класс задач ПР может быть приведен к указанной стандартной форме – форме функции реализации. Отметим также, что многие практические задачи ПР непосредственно формулируются в форме функции реализации. Это, прежде всего, такие задачи, где реально существует среда, влияющая на результат принятия того или иного решения. В качестве примера могут быть указаны задачи принятия оптимальных проектных решений в условиях технологического разброса параметров изделия.

Итак, пусть задана функция реализации $y = F(x, z)$, где множества X, Y, Z уже не будем предполагать конечными. В условиях полной определенности, как мы видели, задана однозначная связь $y = \varphi(x)$, которая, очевидно, и является соответствующей функцией реализации («состояние среды» z задано и фиксировано). Основная задача ПР состоит в поиске ядра бинарного отношения R_Y в множестве исходов Y.

Будем считать, что задана функция $f : Y \to E$, отображающая множество исходов Y на множество вещественных чисел E. Бинарное отношение R_Y задается условием

$$(y', y'') \in R_Y \leftrightarrow f(y') > f(y'').$$

Тогда существует функционал $J: X \times Z \to E$ и задача ПР эквивалентна задаче оптимизации

$$J(x,z) \to \max_{x \in X}. \qquad (1.3.1)$$

В данном случае у функционала J появился новый аргумент z, так как вместо $y = \varphi(x)$ имеем в условиях риска в качестве функции реализации зависимость $y = F(x, z)$.

Таким образом, мы использовали здесь критериальный язык для задания бинарного отношения предпочтения на множестве исходов Y. Более того, исходы y оцениваются в данном случае по однокритериальной схеме, так как задана одна функция f (y) (целевая функция), характеризующая «полезность» исходов.

Таким образом, говоря о задаче ПР, сформулированной в виде (1.3.1), мы имеем в виду выбор решения (альтернативы) x в условиях, когда целевая функция задана, но задана не совсем точно – она содержит неопределенный параметр z. Решая задачу (1.3.1), мы можем определить x лишь как некоторую функцию параметра z: x = x (z). Если никакой информацией о факторе неопределенности z мы не располагаем, то и результат максимизации J произволен. При наличии статистической неопределенности мы предполагаем, что z – случайная величина, закон распределения которой известен.

Методологически важно различать две основные ситуации: 1) исход $y \in Y$, соответствующий принятому решению x, реализуется многократно; 2) исход y реализуется однократно. Например, выбор конструктивных параметров x изделия, выпускаемого серийно, дает пример многократной реализации исхода одного и того же выбора. Напротив, оптимальный выбор параметров уникального изделия – пример второй ситуации.

Обратимся к методам ПР при наличии многократно реализованного исхода. В этих случаях задачу (1.3.1) естественно заменить некоторой вероятностной задачей. Вполне разумным представляется выбор такой альтернативы x, которая максимизирует математическое ожидание критерия, т.е. является решением задачи

$$J_1(x) = \overline{J(x,z)} \to \max_{x \in X}, \qquad (1.3.2)$$

где черта сверху означает математическое ожидание случайной величины J (x, z). Правило выбора оптимальной альтернативы на основе решения задачи оптимизации (1.3.2) называется <u>критерием математического ожидания</u> (или<u> критерием Байеса-Лапласа)</u>. Если предположить, что функционал J характеризует «полезность» или «доход», полученный от решения x и реализовавшегося исхода y, то математическое ожидание можно рассматривать как «средний доход», и, решая задачу (1.3.2), мы фактически максимизируем «средний доход».

<u>Пример 2</u>. Вернемся к ситуации, описанной в примере 3 из введения. Обозначим через p вероятность появления контролера (вероятность его непоявления равна, следовательно, 1–p). Функция J(x,z) может быть представлена в виде матрицы доходов (1.3.2) (перед потерями поставили знак минус).

Таблица 1.4

X	Z	
	z_1 (p)	z_2 (1 – p)
x_1	– 2	– 2
x_2	– 8	0

В таблице J (x_1, z_1) = –2 и т.д. Имеем теперь:

$$\overline{J(x_1,z)} = p(-2) + (1-p)(-2) = -2,$$

$$\overline{J(x_2,z)} = p(-8) + (1-p)\cdot 0 = -8p.$$

Следовательно, согласно критерию (1.3.2), надо предпочесть первую альтернативу x_1 (брать билет) второй, если –2 > –8p, т.е. p > 1/4. В противном случае более предпочтительной следует признать альтернативу x_2. Если считать, что каждый вагон имеет одинаковые шансы посещения контролером, число вагонов равно k, а число контролеров равно r (предполагается, что r ≤ k), то можно положить, что $p \cong r/k$. Таким образом, если на 4 вагона приходится более 1 контролера, выгоднее брать билет!

Пример 3 (продолжение примера о замене вратаря). Будем численно оценивать исходы игры по получаемым очкам: В – 2 очка, Н – 1 очко, П – 0 очков. Тогда таблица, задающая функционал J(x,z), получается непосредственно из табл. 1.3 и имеет вид табл. 1.5.

Таблица 1.5

X	Z					
	z_1 (7/48)	z_2 (7/16)	z_3 (7/24)	z_4 (1/48)	z_5 (1/16)	z_6 (1/24)
x_1	2	1	0	2	1	0
x_2	1	1	1	0	0	0

Аналогично предыдущему примеру вычисляем:

$$\overline{J(x_1,z)} = 2(7/48) + 1(7/16) + 2(1/48) + 1(1/16) = 5/6;$$
$$\overline{J(x_2,z)} = 1(7/48) + 1(7/16) + 1(7/24) = 7/8.$$

Имеем $\overline{J(x_2,z)} > \overline{J(x_1,z)}$ и поэтому, руководствуясь критерием числа ожидаемых очков, принимаем решение, что в подобных ситуациях нецелесообразно заменять вратаря. «В среднем» такая стратегия приведет к успеху, хотя в каждой конкретной игре, конечно, может реализоваться любой возможный исход.

Упражнение. Указанный в последнем примере критерий (число очков) может быть неадекватен цели принимающего решение. Легко представить себе ситуацию, когда выигрыш оценивают числом t, показывающим, во сколько раз выигрыш важнее ничьей (при этом может быть, что $t > 2$). Определите, при каком t выгоднее предпочесть альтернативу x_1 (заменить вратаря). Ответ: $t \geq 2{,}25$.

Замена задачи $J(x, z) \to \max$ задачей $J_1 = M\{J(x, z)\} \to \max$, где $M\{\ldots\}$ - знак математического ожидания – не единственный способ перехода к статистической постановке. Можно поступить и иначе. Например, определенную роль может играть дисперсия критериальной функции J. И, может быть, имеет смысл иногда поступиться немного значением математического ожидания для уменьшения возможного разброса результатов, т.е. уменьшения значения дисперсии:

$$J_2(x) = \overline{J(x,z)} - k\overline{[J(x,z) - \overline{J(x,z)}]^2} \to \max_x . \qquad (1.3.3)$$

Здесь $\overline{[J(x,z) - \overline{J(x,z)}]^2}$ – дисперсия случайной величины $J(x, z)$; k – заданная постоянная. Эту постоянную целесообразно интерпретировать как <u>степень несклонности к риску</u>. Действительно, k определяет «степень важности» дисперсии по отношению к математическому ожиданию случайной величины J. Увеличение значения k приводит, вообще говоря, к уменьшению «среднего дохода» $\overline{J(x,z)}$, но зато уменьшается и вероятность отклонения от «среднего дохода» (в том числе, в сторону его уменьшения). Таким образом, чем больше k, тем менее склонно к риску лицо, принимающее решение. Критерий (1.3.3) обычно называется <u>критерием ожидаемое значение – дисперсия</u>.

Трудности решения задач (1.3.2), (1.3.3) связаны с высокой трудоемкостью процедуры вычисления математического ожидания. Мы должны сначала задать значения компонент вектора x и лишь затем провести усреднение – операцию, связанную, вообще говоря, с вычислением многомерных интегралов и поэтому требующую значительных затрат машинного времени. Иначе говоря, в отличие от детерминистской постановки задачи оптимизации, функционалы J_1, J_2 не заданы в явном виде как функции x. Все это часто заставляет заменять эти задачи на другие. Если решение задачи

$$J(x,z) \to \max_{x}$$

при фиксированном значении случайного параметра z сравнительно просто определяется, то вместо критерия математического ожидания применяется следующий критерий:

$$J_3(x) = J(x,\bar{z}) \to \max_{x}$$

где $\bar{z}$ – математическое ожидание случайной величины z. Здесь важно понимать, что задача максимизации функции J_3 вовсе не эквивалентна такой же задаче для J_1 из (1.3.2). Переход от J_1 к J_3 носит неформальный характер и требует каждый раз дополнительного обоснования и объяснения.

Таким образом, в случае многократной реализации исхода принятого решения проблема выбора мало чем отличается от ситуаций, в которых случайные факторы отсутствуют. Дополнительные сложности здесь носят чисто вычислительный характер и связаны с необходимостью выполнения операций усреднения.

Рассмотрим еще один часто упоминаемый критерий – <u>критерий (принцип) недостаточного основания Бернулли</u>. Этот критерий, по существу, применяется в условиях полной неопределенности, когда информация о вероятностях состояния среды z_i отсутствует.

Сам Яков Бернулли (1654 – 1705) формулировал его следующим образом: если нет данных к тому, чтобы считать одно событие из полной системы несовместимых событий более вероятным, чем другие, то все события нужно считать равновероятными.

В контексте нашей задачи при конечности рассматриваемых множеств этот принцип приводит к оценочной функции

$$y_i = \frac{1}{m}\sum_{j=1}^{m} y_{ij} \to \max_i .$$

При этом, очевидно, множитель $\frac{1}{m}$ может быть опущен без изменения вводимого упорядочения альтернатив и мы приходим просто к операции суммирования «доходов» по строкам матрицы решений.

При практическом использовании рассмотренных статистических критериев могут возникать, однако, значительные трудности, например, связанные с построением набора $\{z_i\}$ состояний среды (в дискретном случае). Обычно вполне справедливо указывается, что состояния z_i должны быть несовместны, а сам набор $\{z_i\}$ обладать свойством полноты – в обычном статистическом смысле. К сожалению, соблюдение этих важных требований часто тоже не спасает ситуации.

Пример. В определенных условиях следующие два набора состояний среды удовлетворяют вышеприведенным требованиям:

Набор 1: z_1 – цель неподвижна;

z_2 – цель перемещается.

Набор 2: z_1 – цель неподвижна;

z_2 – цель перемещается влево;

z_3 – цель перемещается вправо.

Однако, если соответствующие две задачи решать, например, по критерию недостаточного основания Бернулли, то получим различные результаты. Другими словами, проблема формирования наборов $\{z_i\}$ для сложных ситуаций принятия решения – отдельная и далеко не тривиальная задача.

Ситуация становится еще более сложной, если исход принятого решения реализуется однократно («одноразовое использование решения»).

Такой случай характерен, в частности, при решении задач оптимального выбора параметров уникальных изделий, например, мостов, ирригационных сооружений, финансовых проектов, программных продуктов и т.п. При этом информация о статистических характеристиках факторов неопределенности, даже если она и имеется, не имеет никакого смысла: какова бы ни была вероятность того, что значение некоторого числового параметра неопределенности z будет равно 10^{10} или 10^{-10}, мы ничего не сможем сказать о значении функционала J(x,z), которое реализуется в действительности при конкретном выборе x. Здесь мы в силу уникальности ситуации уже не можем «рассчитывать на средний случай». Такого типа задачи принятия решений необходимо решать особыми нестатистическими методами, либо (опять же допуская определенный риск) переходить к другой философии и, по существу, заменять вероятности некоторыми «коэффициентами уверенности» и т.п.

1.3.3 Критерии принятия решений в условиях полной неопределенности.

Как уже указывалось, применение методов теории вероятностей при однократной реализации исхода принятого решения, вообще говоря, неправомерно. Методологически близкая ситуация возникает и в случаях многократной реализации исхода, но при дополнительном предположении, что либо распределение вероятностей параметра z неизвестно, либо параметр неопределенности z изменяется неизвестным образом, но не является случайным (не обладает свойством статистической устойчивости).

В указанных ситуациях информация о факторе неопределенности z обычно имеет вид

$$z \in Z, \tag{1.3.4}$$

где Z – некоторое множество. Но подобной информации также недостаточно для однозначного решения задачи выбора альтернативы x. Напомним, что мы по-прежнему рассматриваем задачу оптимизации $J(x, z) \to \max\limits_{x}$. Из решения этой задачи мы можем определить вектор x как функцию z:

$$x = x\,(z). \tag{1.3.5}$$

Формула (1.3.5) позволяет лишь отобразить множество неопределенности природных факторов Z на множество $G_x \subset X$, которое естественно назвать множеством неопределенности решений x. Выбор конкретного элемента из множества G_x может основываться на введении различных разумных гипотез о поведении среды. Одна из важнейших гипотез такого типа называется гипотезой антагонизма. Она состоит в предположении, что среда ведет себя «наихудшим» (для принимающего решение) образом. В итоге в качестве оптимальной альтернативы выбирается решение следующей задачи оптимизации:

$$J_4(x) = \min_{z \in Z} J(x, z) \to \max_{x \in X}. \tag{1.3.6}$$

Из последнего соотношения видно, что для вычисления значения функционала J_4 (x) при фиксированном значении x решается задача минимизации J (x, z) по z, т.е. подбирается «наихудший» возможный вариант z. Соответствующее значение J и берется в качестве значения функционала J_4, соответствующего заданному x. Принцип выбора оптимальной альтернативы x* на основе решения задачи (1.3.6) называется также принципом гарантированного результата или принципом максимина (используется также название критерий Вальда). Число J_4 (x*) = J* называется гарантированной оценкой, а сам элемент x* – гарантирующим решением. Смысл введенных названий состоит в том, что, каково бы ни было значение параметра неопределенности z, выбор x = x* согласно формуле (1.3.6) гарантирует, что при любом z значение целевого функционала J (x, z) будет не меньше, чем J* (докажите последнее замечание). Оценочная функция данного критерия для дискретного случая имеет вид:

$$y_i = \min_j y_{ij}$$

Очевидно, что если значение функционала J(x,z) отражает не «полезность» альтернативы x, не «доход», а, напротив, – «потери», то исходная задача состоит в минимизации функции J(x,z), а максиминный критерий превращается в минимаксный:

$$J_5(x) = \max_{z \in Z} J(x, z) \to \min_{x \in X} . \qquad (1.3.7)$$

Максиминный критерий является крайне осторожным, «пессимистичным», что может иногда приводить к нелогичным выводам, противоречащим здравому смыслу.

Таблица 1.6

X	Z	
	z_1	z_2
x_1	10 100	100
x_2	10 000	10 000

Пример. Пусть функция J(x,z) задана с помощью табл. 1.6, где элементы матрицы решений имеют смысл «потерь», заданных в некоторых условных единицах и которые следует минимизировать. При выборе решения x_1 или x_2 мы по-прежнему не знаем, какое значение z_1 или z_2 примет фактор неопределенности z. Применение минимаксного критерия приводит к выбору x_2. Но интуитивно мы склонны выбрать x_1, поскольку совсем не исключено, что реализуется «состояние природы» z_2 и наш проигрыш будет существенно уменьшен (равен 100 ед.). В то же время при выборе x_2 мы гарантированно получим потери в 10 000 ед. при любом значении z.

Таблица 1.7

X	Z			
	z_1	z_2	…	z_S
x_1	y_{11}	y_{12}	…	y_{1S}
x_2	y_{21}	y_{22}	…	y_{2S}
…	…	…	…	…
x_n	y_{n1}	y_{n2}	…	y_{nS}

Предположим теперь, что задана табл. 1.7, представляющая функционал J (x, z). Здесь введены обозначения y_{ij} = J (x_i, z_j). Таким образом мы предполагаем конечность множеств X, Z. Поясним на этом примере, как можно исправить положение с излишней «осторожностью» максиминного (или минимаксного) критерия. Введем новую матрицу вместо $\{y_{ij}\}$ следующим образом:

$$r_{ij} = \begin{cases} \max\limits_{k=1,\dots,n} y_{kj} - y_{ij}, & \text{если y – «доход»;} \\ y_{ij} - \min\limits_{k=1,\dots,n} y_{ki}, & \text{если y – «потери».} \end{cases}$$

Иначе говоря, r_{ij} есть разность между наилучшим значением в столбце j и значением y_{ij} при том же j. Следовательно, обработка матрицы $\{y_{ij}\}$ идет «по столбцам».

Построенная таким способом матрица $\{r_{ij}\}$ называется <u>матрицей</u>

сожалений, так как, по существу, каждое число r_{ij} выражает «сожаление» лица, принимающего решение, по поводу того, что он не выбрал наилучшего решения относительно состояния z_j.

Критерий минимального сожаления, предложенный Сэвиджем, состоит в применении минимаксного критерия (независимо от того, какой характер имели элементы y_{ij} – «дохода» или «потерь») к матрице сожалений $\{z_{ij}\}$:

$$J_6(x) = \max_{j=1,\dots,S} r_{ij} \to \min_{i=1,\dots,n} \Rightarrow i_*, x^* = x_{i_*},$$

т.е. числа r_{ij} всегда носят характер «потерь» и их необходимо минимизировать. Соответствующая оценочная функция (при условии, что исходная матрица является матрицей доходов) имеет вид:

$$y_i = \max_j(\max_i y_{ij} - y_{ij}) \to \min_i.$$

Обратимся снова к последнему примеру. Матрица сожалений будет иметь вид табл. 3.7. В этом случае имеем:

Таблица 1.8

X	Z	
	z_1 (ед.)	z_2 (ед.)
x_1	100	0
x_2	0	9 900

$$i = 1: \max_{j=1,2} y_{ij} = \max\{100,0\} = 100;$$

$$i = 2: \max_{j=1,2} y_{ij} = \max\{0,9900\} = 9900.$$

В результате, согласно критерию Сэвиджа, выбираем первую альтернативу x_1, к чему мы и стремились интуитивно.

Следующий критерий оптимальности принимаемого решения называется критерием Гурвица. Этот критерий охватывает ряд различных подходов к принятию решений: от наиболее оптимистичного до наиболее пессимистичного.

Наиболее оптимистичный подход (в предположении, что y_{ij} означает «выигрыш» или «доход») состоит в выборе x^* из условия

$$\max_i \max_j y_{ij} \Rightarrow i_*; x^* = x_{i_*}. \qquad (1.3.8)$$

Аналогично при наиболее пессимистичных предположениях выбираемое решение соответствует

$$\max_i \min_j y_{ij}. \qquad (1.3.9)$$

Критерий Гурвица, называемый также <u>критерием пессимизма – оптимизма</u>, сводится к взвешенной комбинации обоих способов, устанавливая баланс между случаями предельного оптимизма и крайнего пессимизма. Если y_{ij} означает «прибыль» (т.е. соответствующие величины необходимо максимизировать), то выбирается решение из условия

$$\max_i \{\alpha \max_j y_{ij} + (1-\alpha)\min_j y_{ij}\},\ 0 \le \alpha \le 1.$$

Оценочная функция для случая «доходов» имеет, следовательно, вид:

$$y_i = \alpha \max_j y_{ij} + (1-\alpha)\min_j y_{ij} \to \max_i.$$

В том случае, когда y_{ij} представляет «затраты», оптимальное решение удовлетворяет аналогичному соотношению:

$$\min_i \{\alpha \min_j y_{ij} + (1-\alpha)\max_j y_{ij}\}.$$

При $\alpha = 1$ имеем случай предельного оптимизма (1.3.8); при $\alpha = 0$ – случай крайнего пессимизма (1.3.9). Промежуточные значения <u>показателя пессимизма-оптимизма α</u> характеризуют ту или иную склонность лица, принимающего решение, к пессимизму или оптимизму. При отсутствии явно выраженной склонности целесообразно полагать $\alpha = ½$.

В непрерывном случае, когда аргументы функционала J(x,z) не обязаны принадлежать конечным множествам, имеем:

$$\max_{x\in X}\{\alpha \max_{z\in Z} J(x,z) + (1-\alpha)\min_{z\in Z} J(x,z)\} \Rightarrow x^*,$$

или

$$J_7(x) = \alpha \max_{z \in Z} J(x,z) + (1-\alpha) \min_{z \in Z} J(x,z) \to \max_{x \in X}.$$

Аналогично для критерия Сэвиджа:

$$J_8(x) = \max_{z \in Z} r(x,z) \to \min_{x \in X},$$

где (в предложении, что функционал J требуется максимизировать)

$$r(x,z) = \max_{x \in X} J(x,z) - J(x,z).$$

Таблица 1.9

X	Z			
	z_1	z_2	z_3	z_4
x_1	5	10	18	25
x_2	8	7	8	23
x_3	21	18	12	21
x_4	30	22	19	15

Пример. Одно из предприятий, занимающееся обслуживанием населения, должно определить уровень предложения услуг так, чтобы удовлетворить потребности клиентов в течение предстоящих праздников. Точное число клиентов неизвестно, но ожидается, что оно может принять одно из четырех значений: $z_1 = 200$, $z_2 = 250$, $z_3 = 300$, $z_4 = 350$. Для каждого из этих возможных значений z_i существует наилучший уровень предложения (с точки зрения возможных затрат). Отклонения от этих уровней приводят к дополнительным затратам либо из-за превышения предложения над спросом, либо из-за неполного удовлетворения спроса.

Затраты J (в усл.ед.) приведены в табл. 1.9, где x_i означают варианты уровней предложения, среди которых надлежит найти оптимальный. Заметим, что все отраженные в табл. 1.8 уровни предложения оказываются наилучшими для соответствующих значений z_i. Так, x_1 оказывается наилучшим при $z = z_1$, x_2 – при $z = z_2$, x_3 – при $z = z_3$ и x_4 – при $z = z_4$. Таким образом, «лишних» x_i в табл. 1.9. не содержится. Применение минимаксного критерия к выбору решения позволяет получить гарантированное значение $J^* = 21$ и $x^* = x_3$. Критерий Сэвиджа приводит к матрице сожаления:

$$\{r_{ij}\} = \begin{vmatrix} 0 & 3 & 10 & 10 \\ 3 & 0 & 0 & 8 \\ 16 & 11 & 4 & 6 \\ 25 & 15 & 11 & 0 \end{vmatrix}$$

В результате минимаксной обработки матрицы $\{r_{ij}\}$ получаем $x^* = x_2$, что соответствует «сожалению», равному 8.

Критерий Гурвица при $\alpha = ½$ приводит к выбору решения $x^* = x_1$ или $x^* = x_2$. Необходимые промежуточные результаты представлены в табл. 1.10.

Таблица 1.10

X	$\min\limits_j y_{ij}$	$\max\limits_j y_{ij}$	$\alpha \min\limits_j y_{ij} + (1-\alpha) \max\limits_j y_{ij}$	Примечание
x_1	5	25	15	$\min\limits_i$
x_2	7	23	15	
x_3	12	21	16,5	
x_4	15	30	22,5	

Упражнения.

1. Примените минимаксный критерий в приведенном примере, если четвертое значение возможного числа клиентов z_4 исключено.

Ответ. Минимаксное значение равно 8 и соответствует решению x_2.

2. Примените критерий Сэвиджа, предполагая, что решение x_2 исключено.

Ответ. Минимаксное значение $r_{ij} = 10$ и соответствует выбору x_1.

3. Решите этот пример с помощью критерия Гурвица при $\alpha = 0,75$.

Ответ. Следует выбрать x_1 со значением целевой функции 10.

1.3.4 Некоторые трудности.

Рассмотренные в предыдущем разделе критерии обладают целым рядом недостатков и логических противоречий. Это довольно тонкие вопросы и мы здесь не имеем возможности дать достаточно полное изложение этих проблем. Рассмотрим только несколько примеров и наводящих соображений, позволяющих уяснить характер возможных затруднений.

<u>Пример 1</u>. Рассмотрим случай, когда ЛПР (лицо, принимающее решение) не может остановиться ни на одном из предложенных критериев: 1) максиминном, 2) Гурвица (α = ¾) и 3) недостаточного основания Бернулли при выборе альтернативы согласно заданной матрице «доходов» (табл. 1.11).

Таблица 1.11

X	Z		
	z_1	z_2	z_3
x_1	2	12	–3
x_2	5	5	–1
x_3	0	10	–2

Поэтому ЛПР решает считать альтернативу x_i предпочтительнее, чем x_j в том и только в том случае, если на это указывает большинство из трех рассмотренных критериев.

Легко проверить, что порядок полученных предпочтений имеет вид

1) $x_2 \succ x_3 \succ x_1$;

2) $x_3 \succ x_1 \succ x_2$;

3) $x_1 \succ x_2 \succ x_3$.

Таким образом, большинство критериев указывает на то, что $x_1 \succ x_2$

(критерии 2) и 3)), $x_2 \succ x_3$, $x_3 \succ x_1$:

$$x_1 \succ x_2 \succ x_3 \succ x_1$$

Получили так называемый «порочный круг» (нарушение транзитивности) – уже знакомый нам парадокс группового выбора на основе принципа большинства.

Пример 2. Ранее мы уже вводили упорядочение вида:

$x_1 \overset{p}{\succ} x_2 \leftrightarrow [\forall z \in Z : F(x_1, z) \geq F(x_2, z),$ причем хотя бы для одного z неравенство строгое].

Это свойство строгого доминирования одной строки матрицы решений (для дискретного случая) над другой. Интуитивно представляется естественным, чтобы «хороший» критерий удовлетворял следующему требованию.

Аксиома 1. Если $x' \overset{p}{\succ} x''$, то x'' не может быть оптимальным.

В то же время применение максиминного (минимаксного) критерия и критерия Гурвица к матрице доходов

Таблица 1.12

X	Z		
	z_1	z_2	z_3
x_1	0	1	3/4
x_2	0	1	1/2

приводит к оптимальности как x_1, так и x_2. А это противоречит казалось бы очевидной аксиоме 1, так как в данном случае $x_1 \overset{p}{\succ} x_2$ и x_2 не должно быть оптимальным.

Рассмотрим еще одно «очевидное» требование к «хорошему» критерию.

Аксиома 2. Добавление к матрице решений новой строки, которая доминируется одной из уже имеющихся строк, не влияет на оптимальность прежних решений.

Однако следующая жизненная ситуация показывает и правомерность подхода, связанного с нарушением аксиомы 2.

Пример 3. [Кини, Райфа]. Человек, блуждая в чужом городе в обеденное время, случайно набрел на скромный ресторан и нерешительно входит в него. Официант сообщает ему, что меню нет и что посетитель может получить либо отварную лососину за 2,5 $ либо бифштекс за 4 $. В первоклассном ресторане он выбрал бы бифштекс, но учитывая, что ему неизвестна обстановка, и учитывая разницу цен, он выбирает лососину. Вскоре официант возвращается из кухни и многословно извиняется, упрекая неразговорчивого шеф-повара, что тот упустил сказать ему, что в меню есть также жареные улитки и лягушачьи лапки, то и другое по 4,5 $. Оказывается, что наш герой питает отвращение к тому и другому и предпочел бы им лососину, тем не менее он отвечает: «Прекрасно, я переменю свой заказ на бифштекс».

Очевидно, это является нарушение вроде бы обоснованной аксиомы 2, ибо предполагается, что каждая из вновь добавленных альтернатив (улитки и лягушачьи лапки) в любом случае проигрывает уже имеющейся альтернативе –лососине, т.е. является доминируемой. Но можем ли мы сказать, что посетитель действует неразумно? Он, подобно многим другим, заключил, что лишь в «хороших» ресторанах подаются улитки и лягушачьи лапки, и поэтому риск получить плохой бифштекс, по его мнению, уменьшается.

Данные рассуждения, конечно, поддаются критике, но мы теперь уже не так уверены в бесспорности аксиомы 2.

Пример 4. Еще одно возможное возражение против весьма распространенного критерия Гурвица состоит в том, что для нижеследующей задачи выбора он дает решение, противоречащее здравому смыслу. Действительно, рассмотрим матрицу доходов

Таблица 1.13

X	Z				
	z_1	z_2	z_3	…	z_{100}
x_1	0	1	1	…	1
x_2	1	0	0	…	0

Согласно критерию Гурвица оба решения x_1 и x_2 равноценны и их оценки для любого α равны $1 - \alpha$. Однако если истинное состояние среды z_i совершенно неизвестно, то интуитивно мы бы предпочли x_1, подразумевая, что реальным состоянием «вероятнее» окажется одно из состояний $z_2 \div z_{100}$, а не z_1.

Здесь, по существу, возникают почти философские проблемы, связанные с понятием «полного незнания» и т.д. Ведь интуитивно мы все равно предположили, что вероятность реализации одного из состояний $z_2 \div z_{100}$ выше, чем у z_1.

Основной вывод заключается в том, что сложность проблемы принятия решений в значительной степени определяться самим процессом формализации задачи, например, в виде соответствующей матрицы решений. Это далеко не формальный акт и он должен выполняться опытным системным аналитиком, специалистом в конкретной предметной области.

1.3.5 Принятие решений в условиях конфликта (элементы теории игр).

Снова будем считать, что задача принятия решений сформулирована в виде задачи оптимизации

$$J(x,z) \to \max_{x \in X}, z \in Z. \qquad (1.3.10)$$

В отличие от предыдущих случаев, когда параметром z управляла «природа», здесь мы предполагаем, что параметр z управляется «разумным» противником, преследующим собственные цели. Эти цели выражаются с помощью задачи ПР, аналогичной (1.3.10):

$$I(x,z) \to \max_{z \in Z}, x \in X. \qquad (1.3.11)$$

Такого типа конфликтные задачи ПР первоначально были формализованы как задачи анализа салонных игр, что придало всей терминологии несколько легкомысленное звучание. Так, обе противоборствующие стороны называются игроками, выбираемые ими альтернативы (соответственно x и z) – <u>ходами</u>, правила выбора решений – <u>стратегиями</u>, значения функционалов J и I – <u>выигрышами</u>, а вся теория ПР с неопределенностью типа «активный партнер» – <u>теорией игр</u>. Иногда задачи ПР в условиях «природных» неопределенностей, которые мы рассматривали в предыдущем разделе, называются <u>играми против природы</u>.

Итак, пусть два субъекта А и Б, располагающие возможностью выбора соответственно элементов $x \in X$ и $z \in Z$, стремятся к достижению своих целей, представленных в виде (1.3.10), (1.3.11).

Расхождение между функционалами I и J определяет степень антагонизма игроков. В частном случае может оказаться, что $J = -I$ при любых x и z; такую ситуацию, возникающую в игре двух субъектов, называют антагонистической, строго конкурентной или игрой с нулевой суммой (J + I = 0). Однако чисто антагонистическая ситуация является в известном смысле вырожденной. Наиболее типичен конфликт, в котором интересы игроков не совпадают, но и не строго противоположны.

Легко представить себе ситуацию, когда не два, а k игроков максимизируют свои выигрыши p_i (x^1, x^2, …, x^k), i = 1, …, k. В этом случае, например для первого игрока, выбирающего решение x^1, остальные x^i будут составлять фактор неопределенности z: $p_1(x_1, z) \to \max\limits_{x^1 \in X}$, z = ($x^2$, …, x^k). Если $\sum\limits_{i=1}^{k} p_i = 0$, то мы по-прежнему говорим об игре с нулевой суммой, хотя термин «антагонистическая игра» здесь уже неприменим. Далее мы будем рассматривать только игры двух лиц.

Итак, пусть две стороны А и Б стремятся к достижению своих целей:

$$А : J(x, z) \to \max_{x \in X}, z \in Z;$$

$$Б : I(x, z) \to \max_{z \in Z}, x \in X.$$

Оба лица, принимающие решения (ЛПР), или оба «игрока», располагают возможностью выбора x и z соответственно. Далее для определенности будем полагать, что $X \subset E^n$, $Z \subset E^m$, т.е. x и z – числовые векторы соответствующих размерностей.

Таблица 1.14

X	Z	
	z_1 = Н	z_2 = П
x_1 = Н	(1, 1)	(10, 0)
x_2 = П	(0, 10)	(7, 7)

Пример такой постановки задачи уже приводился – это пример 4 из введения (дилемма заключенного). Проводя рассуждения со стороны первого игрока (игрока А), легко установить, что оба функционала J и I задаются табл. 1.14. На пересечении строки i и столбца j в табл. 1.14 стоит пара чисел (p, q), где $p=J(x_i,z_j)$, $q=I(x_i,z_j)$. В данном примере, очевидно, требуется минимизировать функционалы J и I, а не максимизировать.

Далее мы везде будем считать себя игроком А и проводить рассуждения с позиций его интересов.

В связи с тем, что исход нашего выбора решения зависит от выбора игрока Б, необходимо сделать какие-то предположения о его возможном поведении в процессе решения задачи. Правомерность подобных предположений (гипотез) впрямую зависит от характера информированности сторон о поведении другой стороны.

При принятии решений в условиях риска (а подобные задачи, как уже говорилось, могут также относиться к теории игр – игр против природы) мы, по существу, предполагали, что сторона Б («природа») действует не целенаправленно. Мы предполагали, что каждый выбор $z = z_i$ (при дискретном множестве z) характеризуется своей вероятностью, т.е. мы могли оценить частоту появления тех или иных z_i и в соответствии с этим строили свою стратегию поведения. Это одна из возможных гипотез. При игре с «осмысленным» противником, который преследует в процессе принятия своих решений вполне определенные цели, разумно прибегать к иным гипотезам, лучше отражающим существо такой задачи. По сути дела особым характером вводимых гипотез данный раздел теории ПР и выделяется в отдельную теорию – теорию игр.

Будем различать следующие основные гипотезы (случаи).

Г и п о т е з а 1. Каждый из субъектов А и Б не имеет информации о выборе, который сделан второй стороной. Дополнительные гипотезы о характере поведения второго игрока отсутствуют. В этом случае можно поступать аналогично решению задачи в условиях полной неопределенности. Это, по существу, в точности тот же случай, и мы можем воспользоваться известным принципом наилучшего гарантированного результата. Для субъекта А гарантированная оценка будет равна

$$J^* = \max_{x \in X} \min_{z \in Z} J(x, z), \tag{1.3.12}$$

а для субъекта Б

$$I^* = \max_{z \in Z} \min_{x \in X} I(x, z). \tag{1.3.13}$$

Решая задачи максимизации (1.3.12), (1.3.13), мы находим и векторы x*, z*, реализующие соответствующие гарантированные оценки.

Пример 1. Дадим графическую иллюстрацию применения принципа гарантированного результата. Пусть $J(x,z) = x^2 - z^2$, $I(x,z) = - J(x,z)$ и требуется минимизировать J и I. В результате мы имеем антагонистическую игру:

$$J(x, z) = x^2 - z^2 \to \min_{x \in X}, z \in Z;$$

$$J(x, z) = -I(x, z) = x^2 - z^2 \to \max_{z \in Z}, x \in X.$$

Будем считать, что X = Z = R есть множества всех вещественных чисел (здесь мы использовали то очевидное обстоятельство, что вместо поиска минимума функции I можно искать максимум функции – I = J). Для данного примера гарантированная оценка находится из условия

$$J^* = \min_{x} \max_{z} J(x, z).$$

Обозначим

$$\varphi(x) = \max_{z} J(x, z). \tag{1.3.14}$$

Таким образом, для вычисления одного значения функции φ для фиксированного x необходимо решить задачу оптимизации (1.3.14).

Получим

$$\varphi(x) = \max_z (x^2 - z^2) = x^2,$$

так как любой $z \neq 0$ приводит к уменьшению функции J. Теперь находим

$$\min_x \varphi(x) = \min_x x^2 = 0,$$

что достигается при x = 0. Таким образом, мы получим $J^* = 0$ и при этом $x^* = 0$. Это гарантированный результат, ибо при любом z мы будем иметь значение J не хуже (т.е. не больше), чем нуль, т.е. при любом z

$$J(x^*, z) = J(0, z) = -z^2 \leq J^0 = 0.$$

На рис. 1.20 а) представлены линии постоянного уровня функционала J (x, z) на плоскости (x, z). Напомним, что линией уровня называется геометрическое место точек на плоскости, где J = C = const. Меняя постоянную C, мы будем получать различные линии уровня. Если функция зависит более чем от двух переменных, то следует говорить не о линиях уровня, а о поверхностях уровня. На рис. 1.20 б) изображена зависимость J (x, z) в трехмерном пространстве, имеющая характерный вид «седла». Можно считать, что соответствующая поверхность «склеена» из двух видов парабол: $y_1 = x^2$, $y_2 = -z^2$. При выборе гарантирующего решения $x^* = 0$ мы при различных z будем всегда находиться на параболе A (рис. 1.20 б)), обеспечивая выполнение неравенства $J \leq J^* = 0$.

а) б)

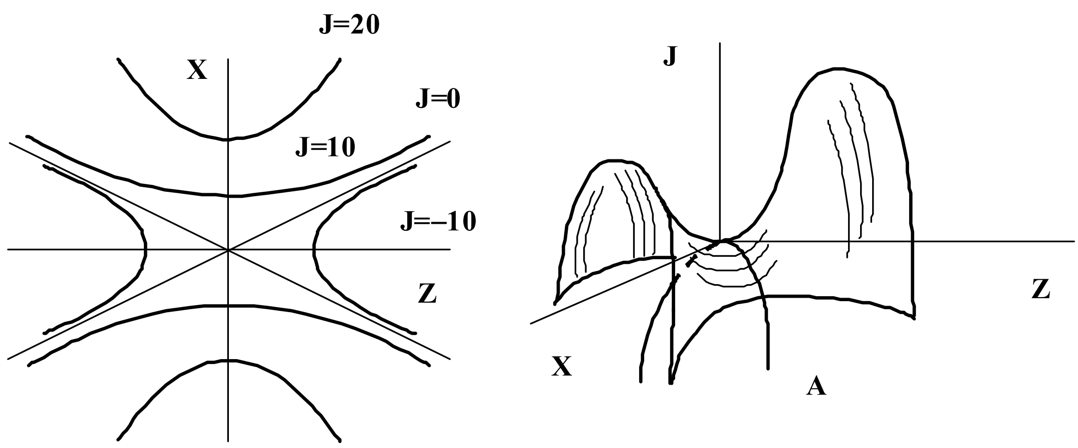


Рис. 1.19

Г и п о т е з а 2. Предполагаем, что субъект Б следует принципу максимина и выбирает z* из условия (1.3.13):

$$I^* = \max_{z \in Z} \min_{x \in X} I(x, z).$$

Тогда мы можем выбирать x согласно правилу

$$J(x, z^*) \to \max_{x \in X}, \qquad (1.3.15)$$

где z* – гарантирующее решение второго игрока. Обозначим решение задачи (1.3.15) через x**. При этом оказывается, что

$$J^{**} = J\,(x^{**}, z^*) \geq J^*, \qquad (1.3.16)$$

где J* – наша гарантированная оценка, получаемая по принципу максимина.

<u>Упражнение</u>. Доказать неравенство (1.3.16).

На примерах легко убедиться, что неравенство (1.3.16) может быть строгим, и, следовательно, следуя гипотезе 2, мы в случае ее правомерности можем получить реальный выигрыш, выбирая решение x**, а не x*.

Таблица 1.15

X	Z			
	z_1	z_2	z_3	z_4
x_1	5	– 10	9	0
x_2	6	7	8	1
x_3	8	7	15	2
x_4	3	4	– 1	4

Пример 2. Игра с нулевой суммой задана с помощью табл. 1.15, где числа на пересечении строк и столбцов означают наш выигрыш, т.е. проигрыш игрока Б – нашего соперника. Наша гарантирующая стратегия $x^* = x_3$, а гарантированная оценка $J^* = 2$. (Если бы мы выбрали другое решение, отличное от x_3, то могли бы в зависимости от действий игрока Б получить и меньшее значение выигрыша, чем 2). Аналогично для игрока Б (он в отличии от нас стремится минимизировать наш выигрыш, а тем самым и свой проигрыш) имеем $z^* = z_4$, $I^* = 4$. Действительно, игрок Б выбирает тот столбец, в котором максимальное число было бы наименьшим. В первом столбце максимальное число равно 8, во втором – 7, в третьем – 15 и в четвертом – 4. Следовательно выбирая $z^* = z_4$, игрок Б никогда не проиграет больше четырех условных единиц потерь.

Если выбирать x^{**} из условия

$$J(x, z^*) \to \max_{x \in X},$$

то мы получаем $x^{**} = x_4$, $J(x^{**}, z^*) = J^{**} = 4 > J^* = 2$. Таким образом, следуя гипотезе 2, можно, вообще говоря, получить лучший результат по сравнению с принятием решений на основе принципа гарантированного результата.

Г и п о т е з а 3. Мы теперь можем допустить, что субъект рассуждает точно так же, как и в предыдущем случае, т.е. использует не стратегию z*, а аналогичную стратегию z**. Поэтому мы можем это учесть и выбирать оптимальное решение с учетом уже этой гипотезы:

$$J(x, z^{**}) \to \max_{x \in X} \Rightarrow x^{***}, J^{***}.$$

Г и п о т е з а 4. Возможен другой сорт гипотез: мы по условиям игры знаем первый ход субъекта Б (он нам его обязан сообщить). Тогда наше поведение будет определяться стратегией в виде функции x = x (z). Мы можем ее определить в результате решения задачи оптимизации

$$J(x, z) \to \max_{x \in X}. \qquad (1.3.17)$$

Условие (1.3.17) позволяет для каждого фиксированного z определить искомое значение x, т.е. задать функцию x (z).

Для этого случая мы также можем определить гарантированный результат $\hat{J}$

$$\hat{J} = \min_{z \in Z} \max_{x \in X} J(x, z) = \min_{z \in Z} J(x(z), z).$$

Результат $\hat{J}$ будет отличаться от значения J*, найденного согласно гипотезе 1. Именно, во всех случаях будем иметь

$$\hat{J} \geq J^*. \qquad (1.3.18)$$

Таким образом, принятие гипотезы 4 вновь позволяет улучшить результат, полученный по принципу максиминного гарантированного результата.

Докажем неравенство (1.3.18), которое имеет вид

$$\min_{z \in Z} \max_{x \in X} J(x, z) \geq \max_{x \in X} \min_{z \in Z} J(x, z).$$

Для любых фиксированных x′, z′, очевидно, справедливо неравенство

$$\varphi_1(z') \geq \varphi_2(x'), \qquad (1.3.19)$$

где

$$\varphi_1(z') = \max_x J(x, z'); \quad \varphi_2(x') = \min_z J(x', z).$$

Действительно, пусть

$$\max_x J(x, z') = J(x'', z');$$

$$\min_z J(x', z) = J(x', z'').$$

Отсюда имеем следующую цепочку неравенств:

$$\varphi_1(z') = J(x'', z') \ge J(x', z') \ge J(x', z'') = \varphi_2(x').$$

Таким образом, (1.3.19) доказано. Далее, так как x', z' могут быть любыми, то их можно выбрать следующим образом:

$$x' = \arg\max_{x \in X} \varphi_2(x); \quad z' = \arg\min_{z \in Z} \varphi_2(z).$$

Подставляя эти значения в (1.3.19), приходим к (1.3.18).

В качестве примера рассмотрим игру, представленную в табл. (1.15). Для этой игры, как мы видели,

$$J^* = \max_x \min_z J(x, z) = 2$$

Для $\hat{J}$ имеем:

$$\hat{J} = \min_z \max_x J(x, z) = 4.$$

<u>Упражнение</u>. Проверьте приведенные числа.

Г и п о т е з а 5. Пусть Б знает наш первый ход. В этом случае естественно предположить, что он будет придерживаться стратегии z = z (x), которая строится в результате решения оптимизационной задачи

$$I(x, z) \to \max_{z \in Z} \Rightarrow z = z(x) \qquad (1.3.20)$$

(именно так мы поступали при принятии гипотезы 4). Принятие этих допущений, т.е. допущения о том, что мы сообщили свой ход субъекту Б, а также допущения об использовании Б стратегии z (x), позволяет нам таким образом воздействовать на выбор субъекта Б, чтобы он в максимальной степени соответствовал нашим целям. Именно, мы можем выбирать x из условия

$$J(x, z(x)) \to \max_{x \in X} \Rightarrow \tilde{x}.$$

Если максимум в соотношении (1.3.20) достигается не в одной точке z, а на некотором множестве M(x), то наш гарантированный результат $\tilde{J}$ определяется из условия:

$$\tilde{J} = \max_{x \in X} \min_{z \in M(x)} J(x, z).$$

Общим для всех рассмотренных случаев является предположение, что обе стороны, участвующие в игре, не только точно знают свои цели, но и полностью информированы о целевых функциях «противника» или партнера по игре. Для реальных конфликтных ситуаций это не всегда выполняется. Гораздо чаще мы не знаем точно целей наших партнеров, которые, в свою очередь, имеют ограниченную информацию о наших намерениях. Кроме того, необходимо учитывать и возможную сознательную дезинформацию, «блеф» со стороны каждого из игроков. Да и игроков может быть не два, а больше. Формальные модели указанных, а также других игровых ситуаций могут быть построены, но соответствующий материал выходит за рамки этой книги.

До сих пор мы рассматривали проблемы принятия решений в играх двух лиц с позиций одного из игроков. На ту же проблему можно взглянуть со стороны некоторого третьего «нейтрального» лица. Нас здесь будут интересовать некоторые характеристики решения в целом с учетом целевых функций всех игроков. Наиболее важными характеристиками являются, во-первых, <u>свойства эффективности</u> принимаемых решений по получаемым игроками «выигрышам», а во-вторых, <u>свойства устойчивости</u> решений. С позиций третьего лица – «арбитра» игра двух лиц с целевыми функциями

$$\begin{aligned} &J(x,z) \to \max_{x \in X}, \\ &I(x,z) \to \max_{z \in Z} \end{aligned} \qquad (1.3.21)$$

может трактоваться как многокритериальная (в данном случае – двухкритериальная) задача оптимизации на множестве $L = X \times Z$. Аргументом при этом является вектор $\eta \in L$, $\eta = (x, z)$, а задача (1.3.21) принимает обычный вид многокритериальной задачи:

$$\begin{aligned} &J(\eta) \to \max_{\eta \in L}; \\ &I(\eta) \to \max_{\eta \in L.} \end{aligned} \qquad (1.3.22)$$

При анализе эффективности решения задачи (1.3.22) можно снова воспользоваться уже знакомым принципом Парето – важнейшим из принципов отбора рациональных решений. Этот принцип позволяет отбросить все те решения (альтернативы выбора), которые могут быть заменены другими, обеспечивающими лучшие (в данном случае – большие) значения целевых функций всех игроков одновременно или части игроков, но без уменьшения значений целевых функций остальных субъектов, участвующих в игре. Решения, которые не могут быть указанным образом улучшены, мы и называем <u>эффективными</u> или <u>Парето-оптимальными</u>. Такие эффективные решения обладают тем свойством, что улучшать значение целевой функции одного из игроков можно только за счет других субъектов. Казалось бы, задача выбора рациональных компромиссных решений и должна решаться только в пределах множества Парето (которое в теории игр называется также «переговорным множеством»). Ведь совершенно ясно, что любое решение, находящееся вне этого множества, может быть улучшено сразу для вех игроков. Однако реальная ситуация часто оказывается значительно сложнее. Основной вопрос заключается в том, что на самом деле выбор η осуществляется не одним лицом, а несколькими. На самом деле здесь мы имеем игру, а не обычную многокритериальную задачу. Важнейшее значение приобретает другой принцип принятия решений, связанный с понятием устойчивости.

Определение. Будем называть точку $\hat{\eta} = (\hat{x}, \hat{z})$ устойчивым решением или точкой равновесия игры (1.3.21), если

$$\max_{x \in X} J(x, \hat{z}) = J(\hat{x}, \hat{z});$$

$$\max_{z \in Z} I(\hat{x}, z) = I(\hat{x}, \hat{z}).$$

При выборе устойчивого решения $\hat{\eta}$ говорят также, что достигнута ситуация равновесия.

Из приведенного определения непосредственно следует, что неустойчивость какой-либо ситуации проявляется в том, что в случае ее возникновения ей грозит распад, который обусловлен возможностями одного из игроков за счет изменения только своей стратегии улучшить свое положение за счет других. На этом основании возник так называемый принцип устойчивости Нэша (по имени автора – американского математика Джона Нэша). Он гласит, что выбор рациональной стратегии η должен производиться среди множества точек равновесия. Равновесные решения называются также оптимальными по Нэшу. Данный принцип отражает очень важное свойство коллективного решения. Именно, если оба субъекта А и Б смогли договориться о том, чтобы придерживаться выбора $x = \hat{x}$, $z = \hat{z}$, то тот субъект, который нарушает договоренность, прежде всего и пострадает: свойство устойчивости решения дает известную гарантию против нарушения договоренности.

Замечание. Все вышеизложенное справедливо и для случая N игроков, где N > 2.

Рассмотрим теперь связь двух сформулированных принципов выбора решений – принципа Парето и принципа Нэша. Возникает вопрос, насколько хороши устойчивые решения в отношении их эффективности, т.е. в отношении выигрышей, получаемых игроками в разновесных точках. Ведь каждый игрок может рассматривать выбор своего решения как принятие решения в условиях неопределенности (другой игрок выступает в качестве «природной» неопределенности или неопределенности среды). При этом можно воспользоваться принципом наилучшего гарантированного результата и выбрать соответствующую максиминную стратегию, гарантирующую ему независимо от действий другого игрока некоторый минимальный результат. Не получит ли он в таком случае больший выигрыш, чем в ситуации равновесия? Тогда все рассуждения о равновесии вообще не нужны. Справедливо, однако, следующее утверждение: в ситуации равновесия каждый из игроков получает выигрыш, не меньший, чем соответствующий гарантированный максиминный результат.

Докажем данное утверждение. Пусть $(\hat{x}, \hat{z})$ – точка равновесия игры (1.3.21) и x* – максиминная стратегия игрока А. Тогда

$$\max_{x\in X} \min_{z\in Z} J(x, z) = \min_{z\in Z} J(x^*, z).$$

Отсюда, используя определение устойчивого решения, имеем

$$\hat{J} = J(\hat{x}, \hat{z}) \geq J(x^*, \hat{z}) \geq \min_{z\in Z} J(x^*, z) = \max_{x\in X} \min_{z\in Z} J(x, z) = J^*.$$

Таким образом, доказано, что $\hat{J} \geq J^*$.

Следовательно, «максиминное возражение» не проходит и анализ равновесия принимаемых решений имеет под собой реальную основу.

Вместе с тем оказывается (и это принципиально), что устойчивое решение может не принадлежать к числу эффективных, т.е. к множеству Парето. А это уже серьезное замечание (которое ниже будет доказано с помощью соответствующего опровергающего примера). В итоге мы имеем явное противоречие между эффективностью принимаемых решений и их защищенностью от «несанкционированных» действий других игроков. Таким образом, противоречие между оптимальностью по Парето и оптимальностью по Нэшу есть противоречие между выгодностью и устойчивостью.

Только в случаях, когда устойчивые решения являются одновременно паретовскими, можно эффективно использовать принцип Нэша для решения реальных задач. В противном случае мы всегда будем выбирать между эффективностью и надежностью принимаемых решений. В этом, по-видимому, состоит одна из важных первопричин многих конфликтов и неудачных решений в человеческом обществе. Поэтому одним из важных направлений теории ПР и системного анализа является изучение систем, в которых устойчивые точки принадлежат множеству Парето.

Пример 3. Пусть к нерегулируемому перекрестку едут на высокой скорости под прямым углом друг к другу два автомобиля. У каждого из водителей есть две стратегии: 1) снизить скорость до безопасной (безопасная стратегия – стратеги Б), 2) продолжать ехать на высокой скорости (рискованная стратегия – стратегия Р). Если оба водителя будут придерживаться стратегии Б, то это приведет к благополучному исходу, оцениваемому для каждого водителя числом 1. Если оба водителя следуют стратегии Р, то происходит авария и потери каждого отражаются отрицательным числом (–9). При других комбинациях (Б, Р) или (Р, Б) исход оценивается числом 0 для снизившего скорость (за потерю времени) и числом 3 для двигающегося на высокой скорости (за экономию времени). В итоге имеем игру, представленную в табл. 1.16. Числа в таблице представляют соответствующие доходы.

Таблица 1.16

1	2	
	Б	Р
Б	(1, 1)	(0, 3)
Р	(3, 0)	(–9, –9)

В данном случае ситуации (Б, Б), (Р, Р) являются, очевидно, неустойчивыми, так как каждый из водителей может получить лучший результат за счет одностороннего изменения своего решения. Например, водитель 1 может в ситуации (Б, Б) получить лучший для себя результат, изменив стратегию на стратегию Р. В этом случае он получит выигрыш 3 вместо 1. То же справедливо в ситуации (Б, Б) и для игрока 2. Аналогично проверяется и неустойчивость ситуации (Р, Р).

Ситуации (Б, Р) и (Р, Б), напротив, обе являются устойчивыми, так как, если они возникли, ни у одного из игроков нет оснований для одностороннего изменения стратегии своего поведения.

Упражнения. 1. Покажите, что в приведенном примере оптимальной по Парето не является только ситуация (Р, Р). Таким образом, в данной задаче существуют ситуации – (Б, Р), (Р, Б), которые оптимальны одновременно и по Парето, и по Нэшу.

2. Рассмотрите снова пример 4 из введения (дилемма заключенного). Определите, какие стратегии поведения заключенных являются оптимальными по Парето, а какие – по Нэшу. Установите, что решение

(П,П) будет устойчивым по Нэшу, но не оптимальным по Парето, что и доказывает ранее сделанное замечание.

1.4 Многостадийные процессы принятия решений.

1.4.1 Постановка задачи.

Понятие многостадийного (многоэтапного, многошагового) процесса принятия решений весьма многогранно и разнообразно. Поэтому могут рассматриваться совершенно различные модели многостадийности от простых до достаточно сложных. Мы здесь остановимся на обсуждении некоторых традиционных подходов к проблеме, позволяющих уяснить главные черты и особенности многостадийных процессов принятия решений в условиях неопределенности. В частности, будем предполагать, что решаемая проблема является одноцелевой. Например, весьма часто цель всей операции заключается в максимизации «доходов» (прибыли, полезности) или минимизации «затрат». Предполагается, что получение «доходов» реализуется на каждом этапе процесса принятия решений, а затем эти «доходы» суммируются (принцип аддитивности).

Рассматриваемая далее модель многостадийного процесса принятия решений предполагает наличие некоторого графа, носящего название «дерева решений» и, по существу, описывающего, как можно попадать из заданного множества начальных вершин в заданное множество конечных вершин графа. При этом с каждой вершиной графа ассоциируется некоторое состояние S_i , в котором находится объект принятия решений, а дуги, выходящие из вершины, соответствуют возможным переходам из одного состояния в другое в зависимости от принимаемых решений.

На рис. 1.21 дан пример так называемого детерминистского дерева решений.

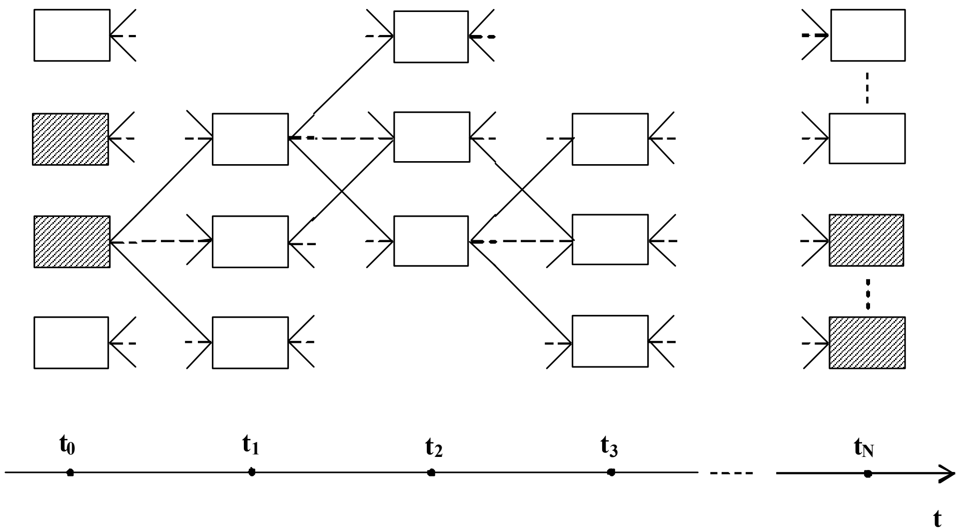


Рис. 1.20

Здесь и далее предполагается, что процесс разворачивается во времени и движение по графу осуществляется слева направо. Допустимые начальные и конечные вершины заштрихованы. Считается, что каждая ветвь графа имеет свой вес – вещественное число – означающее соответствующие локальные «затраты» на переход в другое состояние. Основная задача состоит в оптимальном выборе начальной вершины (из множества допустимых) и пути из нее в любую из допустимых конечных вершин. Оптимальность понимается в смысле построения допустимого пути, реализующего минимальные суммарные затраты (задача выбора минимального пути на графе). В частном случае множества допустимых начальных и конечных вершин могут быть одноэлементными.

В приведенном примере граф содержит только так называемые основные или «решающие» вершины (рис. 1.22).

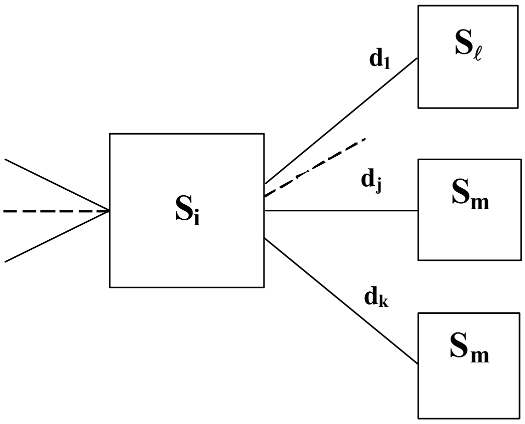


Рис. 1.21

В каждую такую вершину можно попасть различными способами, что показывается наличием нескольких дуг, входящих в вершину. При этом считается, что система (объект принятия решений) находится в определенном фазовом состоянии S_i, а число состояний конечно. Из вершины S_i исходит несколько дуг графа, соответствующих различным решениям, которые могут быть приняты в данном состоянии. Выбор конкретной альтернативы d_i приводит к переходу системы в новую «решающую» вершину (новое состояние).

Более сложная ситуация возникает, когда выбор конкретного решения d_i определяет не конкретное новое состояние системы, а задает некоторую лотерею на множестве возможных новых состояний (плотность распределения вероятности) (рис. 1.23).

Фактически в конечномерном случае (который и рассматривается) это означает, что после выбора d_i мы попадаем в некоторую «случайную» вспомогательную вершину q_{ij} и далее переходим в одно из возможных для данного этапа состояний S_ℓ, …, S_m, S_n в соответствии с заданными вероятностями p_ℓ, …, p_m, p_n (рис. 1.23), где

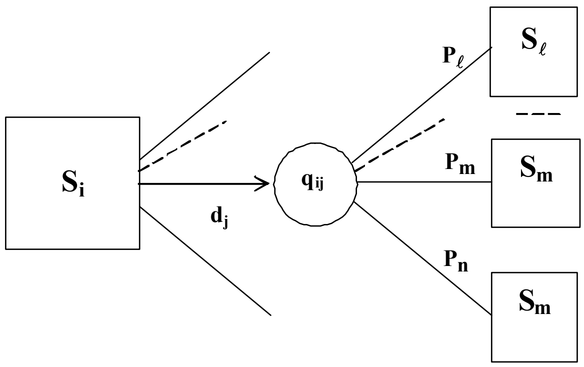


Рис. 1.22

$$\sum_{i=\ell,\dots,m,n} p_i = 1 \;.$$

Это случай так называемой вероятностной неопределенности. В случае полной неопределенности структура рис. 1.23 сохраняется, но стрелки, исходящие из вершины q_{ij} уже не будут иметь весов (соответствующие вероятности отсутствуют).

В пределах одного и того же графа (дерева решений), описывающего конкретную ситуацию, могут реализоваться все возможные виды переходов.

Перейдем теперь к методологии решения сформулированных многоэтапных задач принятия решений.

1.4.2 Детерминистский случай. Метод Беллмана.

Основная схема и главная идея метода Беллмана применительно к рассматриваемой проблематике достаточно проста и естественна. Рассмотрим конкретный пример детерминистского дерева решений (рис. 1.24)

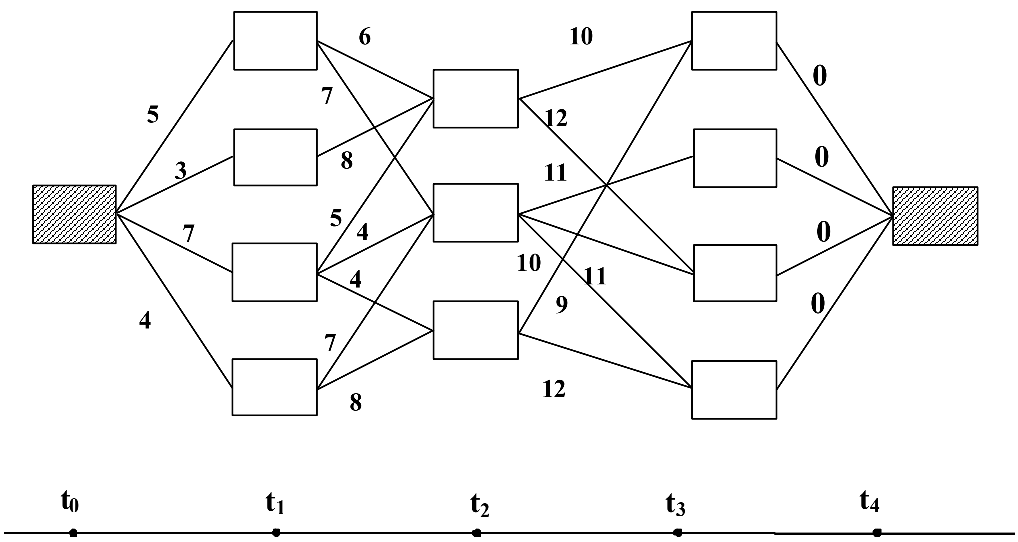


Рис. 1.23

На рис. 1.24 одна начальная и одна конечная вершина. Легко видеть, что, по существу, конечными являются все четыре вершины, соответствующие моменту времени t_3. Однако с помощью введения фиктивной вершины для t_4 удалось свести задачу к графу с одной конечной вершиной. Точно так же можно поступать и с начальными вершинами, если их несколько. Числа у дуг графа означают трудоемкости (затраты), обеспечивающие переход от одной «решающей» вершины к другой. Дуги на промежутке $[t_3, t_4]$ имеют нулевые веса, что и означает, что все вершины, отвечающие t_3, являются, по существу, конечными. Главная задача заключается в выборе оптимального в смысле суммарных затрат пути, соединяющего начальную и конечную вершины.

Основная вычислительная идея метода Беллмана состоит из двух моментов: во-первых, задача поиска оптимального пути начинает решаться с конца, а во-вторых, исходная задача погружается в множество аналогичных задач с различными начальными вершинами и одной и той же конечной вершиной. При этом предполагается, что в качестве начальной вершины последовательно выступают все без исключения вершины графа.

Реализуем метод Беллмана для приведенного примера. Будем продвигаться по графу справа налево, выставляя определенные числа внутри каждой из вершин (помечая вершины) и указывая оптимальные направления движения из каждой вершины с помощью одной или нескольких стрелок. При этом числа внутри квадратиков-вершин будут означать всегда одно и то же – это суммарные затраты, которые получаются при движении из данной вершины, выбранной в качестве начальной, по оптимальному пути (т.е. это наименьшие из возможных затрат). Например, если мы имеем ситуацию, изображенную на рис. 1.25 а), где вершины 2, 3, 4 уже помечены, и надо пометить вершину 1, то будем рассуждать следующим образом (все вершины на рис. 1.25 для удобства объяснения пронумерованы в правом верхнем углу).

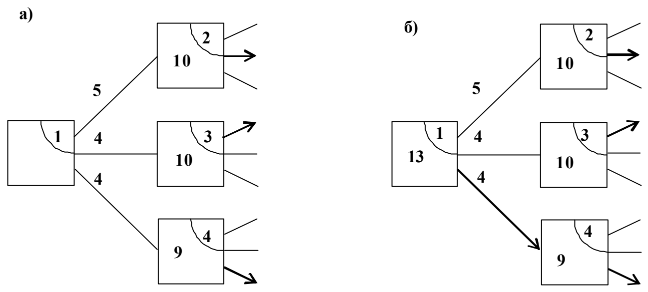


Рис. 1.24

Если в качестве начальной взять вершину 1, то поиск оптимального пути из нее сводится к сравнению трех чисел: 5 + 10 = 15, 4 + 10 = 14, 4 + 9 = 13. Наименьшее из этих чисел – 13 и, следовательно, именно число 13 мы запишем в вершине 1, а стрелочкой соединим 1 и 4 вершины (рис. 1.25 б)). Действительно, по построению число 9 (полученное на предыдущем этапе) означает минимально возможные потери при движении из вершины 4, как из начальной в фиксированную конечную вершину. Затраты в 4 единицы необходимы для перехода из вершины 1 в вершину 4. В итоге мы и получаем число 13. В двух других возможных случаях мы имеем большие затраты и, следовательно, оптимальный маршрут из вершины 1 лежит через вершину 4.

Продвигаясь справа налево, мы, обрабатывая последовательно вертикальные слои вершин, пометим весь граф (рис. 1.26).

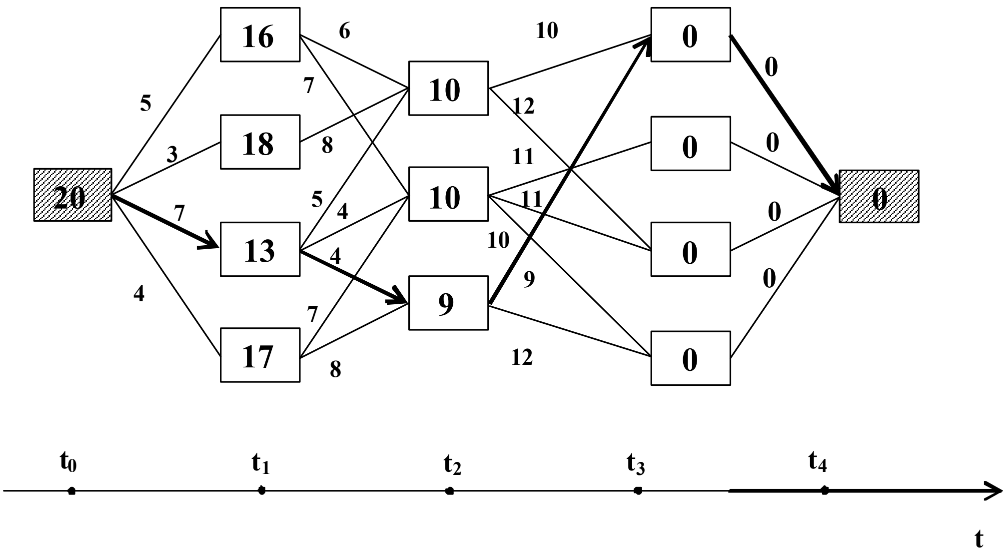


Рис. 1.25

Для восстановления искомого оптимального пути достаточно пройти теперь уже слева направо в направлении стрелок, начиная из уже помеченной начальной вершины (оптимальный путь на рис. 1.26 выделен). Число 20 означает минимально возможные затраты и получается уже на первом этапе после пометки графа. Оптимальный путь, очевидно, может быть и не единственным.

Согласно методу Беллмана одновременно находятся все возможные оптимальные пути.

Можно видеть, что проведенная процедура позволяет находить абсолютный глобальный минимум. Предыдущие рассуждения легко преобразуются к строгому доказательству данного утверждения. В то же время важно понимать, что глобально оптимальная стратегия не есть суперпозиция локально оптимальных стратегий. Скажем, при попытке построения оптимального пути на графе рис. 1.25 при движении слева направо и выборе каждый раз стрелок с наименьшим весом мы, конечно, терпим неудачу и получаем результат:

$3 + 8 + 10 = 21,$

что больше 20.

Пример. Можно дать содержательную интерпретацию многоэтапной задачи принятия решений, представленной графом на рис. 1.25. Основная задача может быть сформулирована следующим образом. Некоторая фирма для реализации проекта должна осуществить постройку здания с привлечением субподрядчиков. На этапе $[t_0, t_1]$ необходимо выполнить нулевой цикл и возвести фундамент. Свои услуги для выполнения этого этапа предложили четыре фирмы, и стоимость работ составляет 5, 3, 7, 4 условных единиц соответственно. Возникает вопрос, какую фирму выбрать? На втором этапе $[t_1, t_2]$ на построенном фундаменте необходимо возвести стены и кровлю. Имеются три фирмы, согласные выполнить указанный объем работ. Однако при этом возникают определенные требования к качеству и конструкции фундамента. Не на любом фундаменте фирма имеет возможность возводить свои стены из своего материала. Поэтому на графе на промежутке $[t_1, t_2]$ каждый квадратик t_1 соединяется не с каждым квадратиком t_2. Даны только допустимые в указанном выше смысле соединения. Веса у дуг означают, как и прежде, стоимость работ.

Точно так же на этапе $[t_2, t_3]$ мы имеем четыре фирмы, осуществляющие внутренние и отделочные работы и завершающие строительство здания.

Основная задача состоит в экономии затрат по привлечению субподрядчиков. Выше эта задача была решена методом Беллмана.

1.4.3 Многостадийные процессы в условиях неопределенности.

Методику решения многостадийных задач принятия решений в условиях неопределенности мы поясним с помощью конкретного примера. В основе по-прежнему лежит метод Беллмана, а также те методы раскрытия неопределенностей, которые обсуждались в предыдущих главах.

Пример. Торговая фирма должна выполнить оптовые закупки товара у внешнего производителя с последующей его перепродажей в течение года в своих торговых точках. Фирма должна принять решение о закупке крупной партии товара или небольшой партии. Впоследствии, если была закуплена небольшая партия, можно докупить товар у производителя (может быть по новым оптовым ценам), а при первоначальной покупке крупной партии существует опасность убытков из-за возможного невысокого спроса на этот товар на внутреннем рынке. Таким образом, решение, в основном, определяется будущим спросом, который заранее достоверно неизвестен. Кроме того, предполагается, что спрос со временем может измениться. По условиям контракта дополнительные закупки товара фирма сможет выполнить лишь через 4 месяца после начала календарного года при условии, что вначале была закуплена небольшая партия. Вопрос о дополнительных закупках встанет, если установится достаточно высокий спрос на товар. Необходимо обеспечить правильные решения как при первоначальной закупке, так и возможной дополнительной закупке товара с целью обеспечения максимальной ожидаемой прибыли, получаемой в течение одного года.

Ежемесячная торговая прибыль, получаемая фирмой в каждой из возможных ситуаций, представлена в табл. 1.17.

Таблица 1.17

	Z_1 (высокий)	Z_2 (низкий)
x_1 (малая партия)	50	40
x_2 (крупная партия)	200	60

В данном модельном примере мы предположили, что спрос может быть либо «высоким», либо «низким». В принципе возможен более подробный подход с более точными градациями спроса. Кроме того, предполагается, что суммарный торговый ежемесячный доход от продажи докупленной через четыре месяца продукции будет несколько меньше, чем при первоначальной закупке крупной партии и составит 180 у.е. в месяц при высоком спросе и 40 у.е. – при низком спросе. (Причины этого могут быть различными, в том числе связанными с условиями дополнительной аренды складских помещений, изменением закупочных оптовых цен и т.п.).

Затраты на закупку крупной и мелкой партий товара соответственно составляют 1000 и 200 у.е., а затраты на возможную дополнительную закупку товара (через четыре месяца) равны 840 у.е.

Будем считать, что проведенные маркетинговые исследования показали, что вероятность высокого спроса на данный товар составляет 0.75, а низкого, соответственно, – 0.25.

Требуется рекомендовать руководству торговой компании такое решение проблемы, чтобы в итоге обеспечить максимальный ожидаемый объем прибыли через год. В данном случае, очевидно, что раз уровень спроса достоверно неизвестен, то речь идет о задаче принятия решений в условиях неопределенности. В частности, можно говорить о максимизации математического ожидания объема прибыли (ожидаемой прибыли). В действительности, в каждом отдельном случае, расчет может не совпадать с реально полученными результатами, но «в среднем», при многократном повторении описанной ситуации выбора, расчетные значения прибыли дадут хорошую оценку для фактически полученной средней прибыли. Если же описанная выше ситуация с торговой компанией является уникальной (единичной), то, по-видимому, целесообразно при выработке решения использовать не критерий математического ожидания, а, например, критерий Гурвица или Сэвиджа. Для полной гарантии расчеты должны быть основаны на максиминном критерии. Применение принципа гарантированного результата в любом случае полезно, по крайней мере на первой стадии исследования, с целью предварительной оценки имеющихся потенциальных возможностей и выбора окончательного критерия.

В силу модельного характера рассматриваемой проблемы мы далее не будем вдаваться в детали соответствующих экономических интерпретаций. Представим сформулированную задачу в виде дерева решений (рис. 1.27).

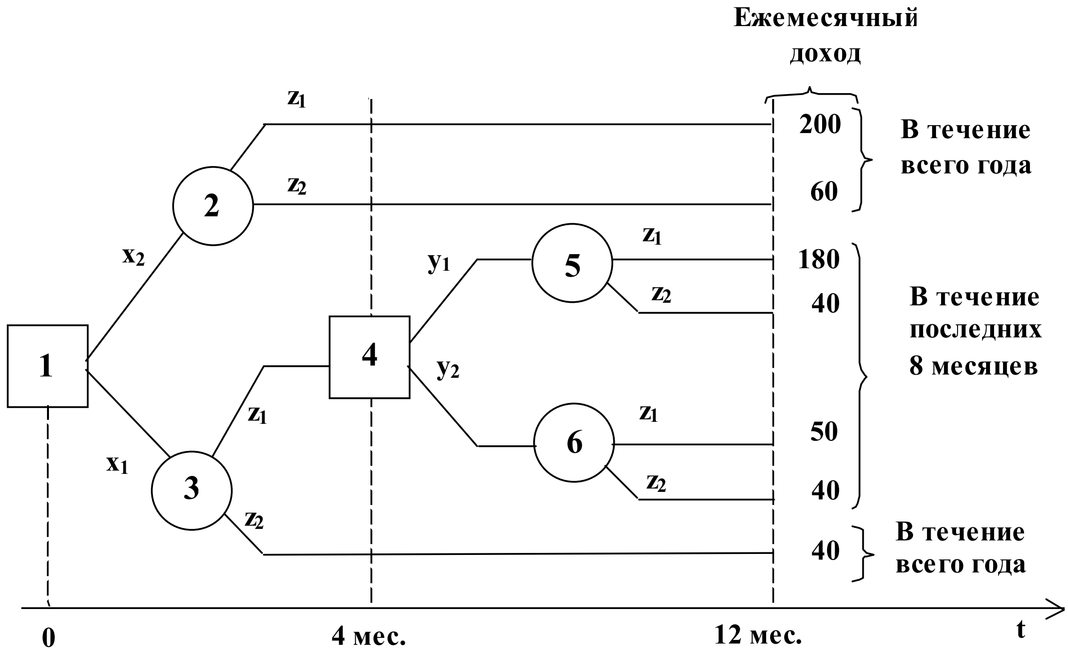


Рис. 1.26

Здесь квадратиками изображены «решающие» вершины, а кружками – вспомогательные вершины, описывающие неопределенное состояние среды. Внутри указаны номера вершин графа. Обозначения x_1, x_2, z_1, z_2 соответствуют табл. 1.17, а переменные y_i означают следующее:

y_1 – решение о дополнительных закупках товара через четыре месяца;

y_2 – решение об отказе от дополнительных закупок.

Дадим решение задачи, воспользовавшись критерием математического ожидания для раскрытия неопределенностей.

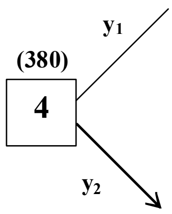


Рис. 1.27

Согласно общей рецептуре метода Беллмана решение задачи начинается движением по «решающим» вершинам графа справа налево. Таким образом, вначале обрабатывается «решающая» вершина 4 (рис. 1.28).

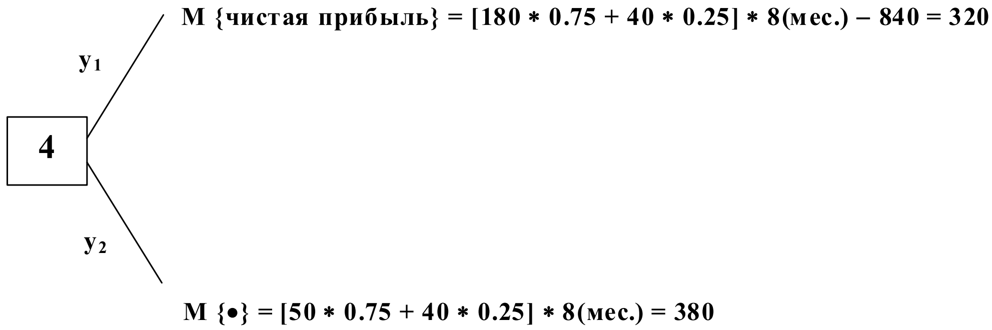


Рис. 1.28

Из полученных двух чисел 380 оказывается большим (мы максимизируем «доходы»). Этим числом помечается вершина 4, а стрелка совпадает с направлением y_2 (рис. 1.29).

Далее переходим к вершине 1 (рис. 1.30).

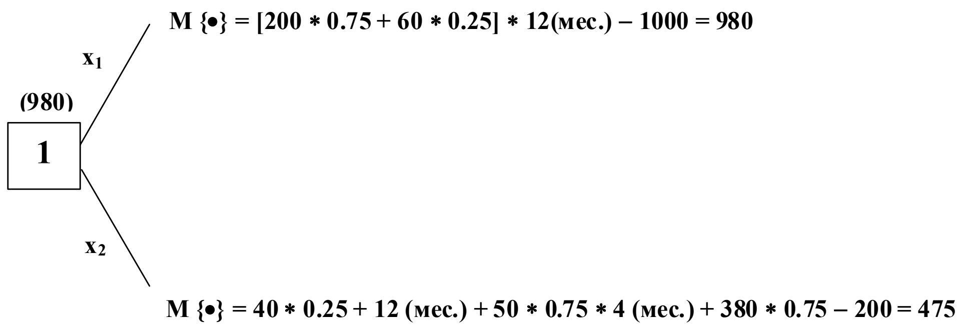


Рис. 1.29

Основной вывод состоит в том, что с позиций критерия математического ожидания выгоднее в вершине 1 идти по направлению x_2, т.е. сразу закупать крупную партию товара (при данных числовых характеристиках задачи). Ожидаемая прибыль составит при этом 980 условных единиц в год.

При обработке вершины 4 мы фактически имели дело с «решающей» матрицей, представленной в табл. 1.18, где

$600 = 180 * 8 - 840$; $-520 = 40 * 8 - 840$;

$400 = 50 * 8$; $320 = 40 * 8$,

Таблица 1.18

Y	Z	
	z_1 (0.75)	z_2 (0.25)
y_1	600	– 520
y_2	400	320

а матрица является матрицей доходов (в ней представлена суммарная прибыль за последние восемь месяцев).

В вершине 1 имеем «решающую» матрицу («доходов»), представленную в табл. 1.19, где

$380 = 50 * 4 + 380 - 200$; $280 = 40 * 12 - 200$;

$1400 = 200 * 12 - 1000$; $-280 = 60 * 12 - 1000$.

Таблица 1.19

X	Z	
	z_1 (0.75)	z_2 (0.25)
x_1	380	280
x_2	1400	– 280

Решим теперь ту же самую задачу, используя принцип гарантированного результата. В вершине 4 по-прежнему имеем матрицу «доходов», представленную в табл. 1.18. Используя принцип гарантированного результата, заключаем, что оптимальным решением является решение y_2, так как наихудший возможный результат при этом равен 320 единицам «дохода», а при выборе y_1 можем получить потери в объеме 520 единиц. С вершиной 4 теперь ассоциируется число 320, а стрелка пойдет в направлении y_2.

Далее переходим к вершине 1 с матрицей табл. 1.20 (она уже отличается от матрицы табл. 1.19).

Таблица 1.20

X	Z	
	z_1	z_2
x_1	320	280
x_2	1400	-280

Здесь:

$320 = 50 * 4 + 320 - 200.$

По критерию гарантированного результата лучшей оказывается альтернатива x_1 с гарантированным «доходом» в 280 у.е. в год. Следовательно, первоначально рекомендуется закупить небольшую партию товара. Если сразу установится высокий спрос на товар (и продержится четыре месяца), то мы окажемся в «решающей» вершине 4 и получим прибыль в $50 * 4 + 320 - 200 = 320$ у.е. за год. Причем через четыре месяца согласно принципу гарантированного результата рекомендуется не делать дополнительных закупок товара (решение y_2 считается оптимальным в вершине 4).

Таким образом, при наличии ситуации неопределенности на различных этапах многошаговой проблемы принятия решений метод Беллмана позволяет указывать оптимальные стратегии поведения в любой «решающей» вершине (т.е. в любом состоянии, в котором может оказаться реальная система). Все эти стратегии представляют для пользователя несомненный интерес, ведь из-за наличия неопределенностей заранее достоверно невозможно установить, какая траектория развития системы реализуется в действительности.

Представленная методика решения многостадийных задач естественным образом обобщается на более сложные многоальтернативные деревья решений.

1.5 Методы многокритериального выбора на основе дополнительной информации пользователя.

Решением многокритериальной задачи, сформулированной в разд. 1.2, является соответствующее множество Парето – множество недоминируемых по Парето альтернатив. Это множество может оказаться достаточно обширным, а пользователя обычно интересует выбор какого-то одного «наилучшего» варианта или небольшого их числа. Если какая-либо дополнительная информация о задаче отсутствует, то множество Парето – это лучшее, что можно предложить. Однако при наличии дополнительной информации о системе предпочтений пользователя могут быть развиты различные методы сужения исходного множества альтернатив – более сильные, чем методы, основанные на доминировании по Парето. Весьма часто исходной информацией для таких методов выступает само множество Парето и ставится задача его сужения с целью выбора одной или нескольких альтернатив в качестве окончательного результата. Некоторые возможные подходы к решению этой проблемы рассмотрены далее.

1.5.1 Адаптивные процедуры выбора

Эти методы основаны на гипотезе о существовании некоторой «функции потерь» $u(x)$, определенной на исходном множестве альтернатив X:

$$u : X \to R,$$

где R – множество вещественных чисел. Основная задача состоит в выборе одного из элементов $x^* \in X$, такого, что

$$x^* = \arg\min_{x \in X} u(x)$$.

Можно вместо «функции потерь» ввести аналогичную «функцию полезности» и ставить задачу ее максимизации. В ряде случаев мы так и будем поступать. Это стандартная задача нелинейного программирования о поиске минимизатора. Отличие рассматриваемой ситуации от типовой задачи нелинейного программирования заключается в следующем. Мы здесь предполагаем, что функция u(x) описывает цель операции выбора, и предпочтения лица, принимающего решение (ЛПР) устроены очень просто: чем меньше значение функции потерь, тем лучше. Главное предположение состоит в том, что функция u(x) считается заранее неизвестной (в противном случае мы просто воспользовались бы методами конечномерной скалярной оптимизации для выбора наилучшей альтернативы).

Далее от ЛПР, решающего многокритериальную задачу выбора, мы будем требовать не оценки значения u(x) для конкретного $x \in X$ (что достаточно сложно), а только способности сравнения двух альтернатив по их векторным оценкам. Как будет ниже показано, этого оказывается достаточно, чтобы реализовать, например, такой метод поиска минимума нулевого порядка как метод Нелдера-Мида. При этом ЛПР выступает в качестве своеобразного измерительного устройства, но ему, как было отмечено, не требуется указывать значения u(x) (что очень важно), а только фиксировать: хуже, лучше, одинаково. Здесь, конечно, существуют свои трудности, связанные, например, с возможной противоречивостью ответов ЛПР, но они преодолимы и мы здесь на них не будем останавливаться.

Рассматриваемые адаптивные процедуры часто оказываются более эффективными и легко реализуемыми, чем так называемые апостериорные процедуры, связанные с явным восстановлением функции u(x) (теория полезности). Действительно, построение u(x) в явном виде позволяет, в частности, упорядочить все альтернативы, что не требуется для решения исходной задачи выбора. В результате проделывается «лишняя» работа, вызывающая необходимость ЛПР отвечать на многочисленные достаточно сложные и неестественные для него вопросы.

Метод Нелдера-Мида

Метод Нелдера-Мида – НМ-метод (или метод деформируемого многогранника) – является стандартным методом нелинейного программирования и изучается в соответствующих курсах по теории конечномерной скалярной оптимизации. Однако мы здесь для полноты изложения кратко укажем основные элементы этого метода, что при необходимости может оказаться и достаточным для написания соответствующей программы.

НМ-метод решает задачу поиска минимизатора x* некоторой заданной функции u:

$$X \xrightarrow{u} R, X \subset R^n,$$
$$x^* = \arg\min_{x \in X} u(x).$$

Прообразом НМ-метода явился симплексный метод Спендли, Хекста и Химсворта, основная идея которого состоит в следующем.

В пространстве поиска R^n строится равносторонний многогранник (регулярный симплекс) с количеством вершин равным n + 1 (для n = 2 – это будет равносторонний треугольник). Далее выясняется, какая из вершин симплекса является наихудшей в смысле значения функции u(x). Для этого можно вычислить u(x) во всех вершинах (если функция u(x) задана аналитически или алгоритмически). Но можно производить попарные сравнения вершин, пользуясь категориями «больше», «меньше», «равно», как указывалось ранее.

Найденная наихудшая вершина заменяется на новую вершину, которая является отражением наихудшей вершины относительно центра тяжести оставшихся вершин. Получается новый симплекс, где вся процедура повторяется. В результате симплекс передвигается по пространству поиска в сторону искомого минимизатора функции u(x) (рис. 1.31)

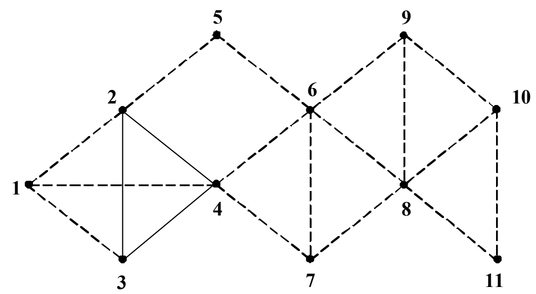


Рис. 1.30

На самом деле в таких процедурах принимаются специальные меры, предотвращающие циклы (циклические движения), а также используются правила уменьшения размера симплекса.

Описанный метод испытывает определенные трудности в связи с отсутствием механизма ускорения поиска в перспективных направлениях, а также в связи с продвижением вдоль искривленных оврагов и хребтов функции u(x). В НМ-метод внесены соответствующие усовершенствования, и симплекс получает возможность изменять свою форму, вытягиваться, сжиматься и, таким образом, не будет оставаться симплексом. Поэтому для него используется более подходящее название – деформируемый многогранник, а сам метод часто называется методом деформируемого многогранника.

Основные операции НМ-метода:

1. <u>Отражение</u> – проектирование худшей вершины x^h через центр тяжести x^c оставшихся вершин (рис. 1.32):

$x^r = x^c + \alpha(x^c - x^h), \ \alpha > 0.$

Здесь x^c – не вершина, а точка центра тяжести вершин 2, 3.

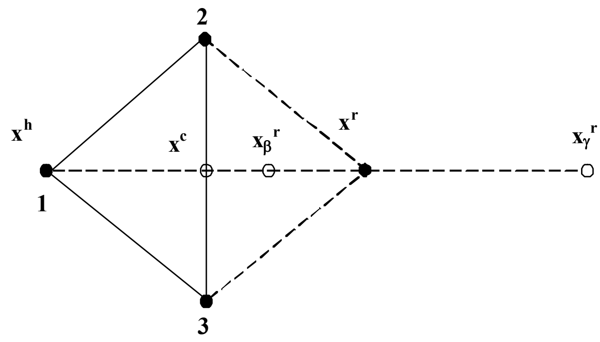


Рис. 1.31

2. Растяжение. Если значение функции u(x) в точке x^r оказывается лучше, чем в лучшей вершине из списка {1, 2, 3}, то выполняется растяжение в γ раз ($\gamma > 1$ – коэффициент растяжения), и точка x^r заменяется на точку x^r_γ.
3. Сжатие. Если в отраженной вершине x^r значение функции u(x) хуже, чем во всех других вершинах (кроме x^h), то производится сжатие (с коэффициентом $0 < \beta < 1$) и точка x^r заменяется на точку x^r_β.
4. Редукция. Если в отраженной вершине значение функции u(x) хуже, чем в точке x^h, то весь многогранник сжимается в 2 раза относительно лучшей вершины x^b (рис. 1.33).

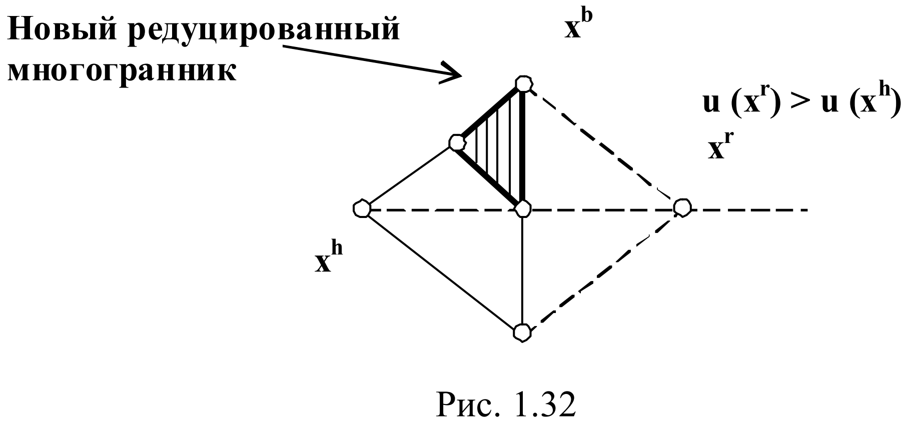


Рис. 1.32

5. В остальных случаях операции 2, 3, 4 не производятся, а сам процесс продолжается для нового многогранника $\{2, 3, x^r\}$.

6. Окончание процесса производится, когда выполняется условие приблизительного равенства значений функции в вершинах текущего многогранника и в центре тяжести многогранника без учета худшей вершины. Могут быть использованы и другие условия окончания процесса.

Реализация адаптивной процедуры выбора на основе НМ-метода

Итак, решается следующая основная задача:

$$f_i(x) \to \min_{x \in X}, \quad (i = 1, ..., m), \quad X \subset R^n,$$

где X – множество альтернатив; через F = f (X) обозначим множество допустимых оценок.

Основная гипотеза: в многокритериальной задаче, в которой необходимо выбрать единственную альтернативу x* из всех возможных (или небольшое число «наилучших» альтернатив), необходимо предположить, что существует единственная, хотя, возможно, и неизвестная ЛПР цель операции, описываемая скалярной функцией u(x), причем

$$x^* = \arg\min_{x \in X} u(x).$$

Будем предполагать также, что эта функция u(x) «согласована» с векторным отображением $f = (f_1, ..., f_m)$ в том смысле, что

$$\forall \arg\min_{x \in X} u(x) \in P(X),$$

где P (X) – множество Парето заданной многокритериальной задачи.

Из теории многокритериальной оптимизации известно (см. разд. 1.2.), что при определенных предположениях о выпуклости множества достижимости F найдутся такие весовые коэффициенты

$$\alpha_i^* \ge 0, i = 1,...,m; \quad \sum_{i=1}^{m} \alpha_i^* = 1,$$

что линейная функция

$$J(x) = \sum_{i=1}^{m} \alpha_i^* f_i(x)$$

достигает своего минимума в точке x*, так как последняя является точкой Парето. Поэтому нет никакой разницы, определять ли x* или соответствующие весовые коэффициенты α*.

Согласно этой основной идее процесс выбора будет протекать следующим образом.

Каждому $\alpha = (\alpha_1, ..., \alpha_m)$ будет соответствовать точка $x(\alpha) \in P(X)$, полученная как решение задачи минимизации линейной свертки

$$J(x) = \sum_{i=1}^{m} \alpha_i f_i(x) \to \min_{x \in X} \tag{1.5.1}$$

любым методом скалярной оптимизации. Векторная оценка $f(x(\alpha))$ будет одновременно оценкой α. А далее во множестве

$$A = \{\alpha = (\alpha_1,...,\alpha_m) \mid \alpha_i > 0, \sum_{i=1}^{m} \alpha_i = 1\},, A \subset R^m$$

реализуется НМ-метод, позволяющий последовательно находить:

$\alpha^* \to x^* = x(\alpha^*) \to f^* = f(x^*)$.

Таким образом, в отличие от возможного «прямого» подхода, когда процедура НМ-метода реализуется непосредственно во множестве $X \subset R^n$, мы здесь работаем в пространстве весовых коэффициентов R^m, причем, как правило, m << n. Правда, мы вынуждены решать вспомогательные оптимизационные задачи в пространстве R^n при минимизации функционала линейной свертки (1.5.1) для каждого пробного α. Но эти задачи решаются не в режиме диалога с пользователем и поэтому могут быть решены с меньшими временными затратами. Наиболее трудоемкая «диалоговая» часть процедуры выбора реализуется в пространстве R^m существенно меньшей размерности и в этом главный выигрыш построенной «косвенной» процедуры.

Важно отметить, что предлагаемый «косвенный» подход оказывается реализуемым и в случае, если X является множеством (конечным или бесконечным) объектов произвольной природы, т.е. требование $X \subset R^n$ является в данном случае, вообще говоря, непринципиальным. Важно лишь, чтобы для каждого из элементов $x \in X$ можно было вычислить соответствующую векторную оценку f (x) согласно заданному отображению

$$X \xrightarrow{f} R^m.$$

Если абстрактные объекты из X являются непараметризованными (т.е. с ними не ассоциируются какие-либо числовые векторы), то обычно множество X оказывается конечным и вспомогательные задачи минимизации (1.5.1) решаются простым перебором вариантов. При этом диалоговый НМ-метод по-прежнему реализуется в числовом непрерывном пространстве весовых коэффициентов R^m, как это и было описано выше.

Дополнительное преимущество рассмотренной косвенной реализации по сравнению с прямой реализацией НМ-метода во множестве $X \subset R^n$ заключается в том, что здесь мы гарантированно осуществляем выбор строго в пределах множества Парето и поэтому гарантируется эффективность получаемых решений, независимо от системы предпочтений ЛПР.

Легко видеть также, что вместо линейной свертки (1.5.1) можно использовать более эффективную свертку Джоффриона, реализующую принцип лексикографического упорядочения, не требуя свойства выпуклости множества достижимости F.

1.5.2 Выбор на основе метода t-упорядочения

Пусть решается детерминистская многокритериальная задача

$$f_i(x) \to \max, \;\; f_i : X \to R, \;\; i = \overline{1, m}, \qquad (1.5.2)$$

где X – произвольное абстрактное множество. В данном разделе будем предполагать, что все критериальные функции f_i отражают «полезность» объекта с позиций различных критериев и являются соизмеримыми в том смысле, что значения каждой критериальной функции изменяются в одних и тех же пределах [a, b]:

$$\forall x \in X : 0 \le a \le f_i(x) \le b, \;\; i = \overline{1, m}. \qquad (1.5.3)$$

Таким образом, мы предполагаем, что оценочные шкалы критериев являются числовыми и одинаковыми.

Отметим, что предположение о соизмеримости критериев является существенным и требует для каждой решаемой задачи отдельного обоснования и исследования.

Требование, связанное с необходимостью приведения всех числовых шкал к единому промежутку выглядит весьма невинно и формально достигается с помощью, например, следующих простых преобразований:

$$\bar{f}_i(x) = a + (b - a)\frac{f_i(x) - \min f_i}{\max f_i - \min f_i}. \qquad (1.5.4)$$

Здесь $\max f_i$, $\min f_i$ – максимальные и минимальные значения f_i соответственно. Новые оценочные функции $\bar{f}_i$ будут изменяться уже в пределах заданного промежутка [a, b]. При этом наименее предпочтительный по любому из частных критериев вариант получит оценку «a», а наиболее предпочтительный – оценку «b». Часто полагают a = 0, b = 1. Могут использоваться и другие (может быть, нелинейные) формулы нормировки, аналогичные (1.5.4).

Многие полагают, что проблемы соизмеримости числовых критериев вовсе не существует. Достаточно воспользоваться любыми соотношениями типа (1.5.4). Однако, это, к сожалению, не так. Рассмотрим конкретный пример, показывающий, что отношение доминирования, устанавливаемое в пространстве нормированных оценок, неинвариантно (из-за изменения нижних и верхних границ) относительно изменения рассматриваемой совокупности объектов X.

Пусть для трех объектов $X = \{x_1, x_2, x_3\}$ имеем следующие оценки по трем частным критериям $f = (f_1, f_2, f_3)$:

$f(x_1) = (10, 10, 3)$; $f(x_2) = (8, 8, 10)$; $f(x_3) = (0, 0, 0)$.

Тогда для преобразованных оценок при $[a, b] = [0, 1]$ получим

$\overline{f}(x_1) = (1,1,0.3)$; $\overline{f}(x_2) = (0.8,0.8,1)$; $\overline{f}(x_3) = (0,0,0)$.

Если предположить, что все три частные критерия равноправны и нас интересует средняя оценка для каждого объекта, то тогда

$$\overline{f}_1(x_1) + \overline{f}_2(x_1) + \overline{f}_3(x_1) = 2.3 < \overline{f}_1(x_2) + \overline{f}_2(x_2) + \overline{f}_3(x_2) = 2.6$$

и, следовательно, объект x_2 оказывается более предпочтительным, чем объект x_1 (по «средней полезности»).

Но если исключить объект x_3, как наихудший по всем трем критериям, то получим обратное утверждение. Действительно, в этом случае

$\overline{f}(x_1) = (1,1,0)$; $\overline{f}(x_2) = (0,0,1)$

и

$$\overline{f}_1(x_1) + \overline{f}_2(x_1) + \overline{f}_3(x_1) = 2 > \overline{f}_1(x_2) + \overline{f}_2(x_2) + \overline{f}_3(x_2) = 1.$$

Итак, мы получили обескураживающий результат. Если исходная совокупность объектов X содержит три объекта x_1, x_2, x_3, то оказывается, что объект x_2 лучше, чем объект x_1. Если же исключить из рассмотрения (худший по всем критериям) объект x_3, то получим, что объект x_1 оказывается лучше, чем объект x_2. Безобидная на первый взгляд процедура нормировки оказывается на деле весьма сложной, так как приводит к явно антиинтуитивным результатам. Можно пытаться исправить ситуацию и, например, фиксировать верхние и нижние границы критериальных функций раз и навсегда (для данной задачи) независимо от конкретного набора объектов и их оценок. При этом мы получим однозначность, но ситуация в целом вряд ли улучшится, так как установленные границы, по существу, и будут определять соответствующее отношение доминирования, и в этом смысле произвол сохраняется.

Итак, пусть критериальные функции (или просто – критерии) соизмеримы и удовлетворяют условиям (1.5.3). В качестве примера изначально соизмеримых критериев можно привести систему школьных оценок по нескольким предметам f_i.

<u>Определение 1</u>. Нормированные критерии f_i и f_j называются равноценными, (что записывается в виде $f_i = f_j$), если всякие две векторные оценки Z, W, где

$$\begin{aligned}&Z=(z_1,\dots,z_i,\dots,z_j,\dots,z_m), Z=f(x), x\in X\\&W=(z_1,\dots,z_i+\delta,\dots,z_j-\delta,\dots,z_m)\end{aligned} \qquad (1.5.5)$$

одинаковы по предпочтительности при любом δ (большим или меньшим нуля), удовлетворяющим неравенствам:

$a \le z_i + \delta \le b$, $a \le z_i - \delta \le b$.

Легко видеть, что суммы частных оценок в позициях i, j у векторных оценок Z, W совпадают.

Таким образом, если, например, два школьника оцениваются по четырем предметам и имеют оценки (которые необходимо максимизировать)

$$Z = (5, 4, 4, 3),\ W = (5, 5, 3, 3) \qquad (1.5.6)$$

то при условии равноценности критериев f_2, f_3 приведенные векторные оценки будут одинаковы по предпочтительности, так как 4 + 4 = 5 + 3, а остальные оценки совпадают.

Следовательно, если критерии f_i, f_j равноценны, то можно «забрать» δ единиц у частной оценки z_j в (1.5.5) и «передать» их частной оценке z_i. При этом получим векторную оценку, одинаковую с исходной по предпочтительности.

Если в приведенном примере (1.5.6) считать, что оценка Z предпочтительнее, чем W, то естественно предположить, что критерий f_3 важнее критерия f_2. Дадим соответствующее определение.

<u>Определение 2</u>. Критерий f_i более важен, чем критерий f_j (что записывается в виде $f_i > f_j$), если векторная оценка

$$Z = (z_1, ..., z_i, ..., z_j, ..., z_m)$$

менее предпочтительна, чем оценка

$$W = (z_1, ..., z_i + \delta, ..., z_j - \delta, ..., z_m),$$

где

$$\delta \in \{\delta > 0 \mid z_i + \delta \le b,\ a \le z_j - \delta\}.$$

Таким образом, перенос δ единиц ($\delta > 0$) с частной оценки z_j на частную оценку z_i приводит к улучшению ситуации, если $f_i > f_j$.

Введенные определения 1, 2 показывают, как может интерпретироваться дополнительная ординальная (порядковая) информация пользователя (ЛПР) об относительной важности частных критериев, на основе которой и происходит сокращение множества Парето решаемой многокритериальной задачи. Важно понимать, что одна и та же ординальная информация, задаваемая в виде цепочки равенств и неравенств, например, вида

$$f_i > f_j = f_k > f_\ell,$$

в других процедурах выбора может получать совершенно иную интерпретацию, что, в свою очередь, может приводить и к другим системам предпочтений, и к другим результатам выбора. Соответствующие вопросы рассмотрены в этой главе далее.

Приведенные в определениях 1, 2 интерпретации ординальной информации ЛПР позволяют строить отношения доминирования более сильные, чем отношение Парето (что и приводит к сужению последнего, а это наша основная задача). Рассмотрим пример.

Пусть

$$Z = (1, 0.5, 0.1, 0.2);$$

$$W = (0.4, 0.9, 0.1, 0.2)$$

и пусть утверждается, что критерий f_1 важнее, чем f_2 : $f_1 > f_2$.

Эти векторные оценки, очевидно, несравнимы по Парето. Рассмотрим оценку

$$W' = (0.8, 0.5, 0.1, 0.2),$$

полученную из W с помощью «переноса» числа 0.4 со второй позиции в первую. Имеем, согласно определению 2

$$W' \succ W,$$

так как более важному критерию f_1 в оценке W′ соответствует большая оценка (0.8 вместо 0.4). И, так как имеем

$$Z \overset{P}{\succ} W'$$

(доминирование по Парето), то естественно считать

$$Z \succ W' \succ W$$

и в результате

$$Z \succ W.$$

Таким образом, оценки Z, W оказываются уже сравнимыми, и оценка W может быть отброшена. Мы здесь везде предполагаем, что рассматриваемые отношения доминирования являются транзитивными.

Методы упорядочения альтернатив, основанные на рассмотренной процедуре «переноса» (transfer) с учетом ординальной информации пользователя, будем называть методами t-упорядочения.

Рассмотрим укрупненный алгоритм, реализующий метод t-упорядочения.

В качестве исходной информации для алгоритма t-упорядочения принимается множество S высказываний пользователя (ЛПР) об относительной важности частных критериев вида:

$$S = \{f_i = f_j; \ldots; f_q > f_p\}.$$

Мы будем предполагать, что множество S скорректировано следующим образом. Во-первых, необходимо проверить и при необходимости обеспечить непротиворечивость высказываний из S, может быть, с помощью проведения дополнительного диалога с пользователем для уточнения его системы предпочтений. Во-вторых, необходимо расширить множество S за счет добавления новых высказываний, являющихся транзитивными следствиями уже имеющихся (выполнить операцию транзитивного замыкания). Именно, если мы, например, уже ввели два высказывания

$$f_i > f_j;\ \ f_j > f_k,$$

то естественно ввести новое высказывание

$$f_i > f_k$$

и т.п.

Пусть теперь $Z = (z_1, \ldots, z_m)$; $W = (w_1, \ldots, w_m)$ – две векторные оценки, которые необходимо сравнить с учетом дополнительной скорректированной информации S.

Если эти оценки сравнимы по Парето, то задача решена. В противном случае вектор Z фиксируется, а по вектору W формируются следующие два множества (может быть, бесконечные, даже если исходные множества оценок конечны).

1. WE – множество W-эквивалентных векторов (включающее сам вектор W), полученное из W с помощью всех возможных переносов δ между парами равноценных критериев. Следовательно, множество WE строится с учетом всех данных типа $f_i = f_j$ из S.

2. WI – множество W-улучшенных векторов, каждый из которых получен с помощью возможных переносов δ с учетом всех данных

$f_i = f_j$, $f_q > f_p$.

При этом предполагается, что переносы согласно информации $f_q > f_p$ производятся только с целью улучшения вновь полученного вектора и, по крайней мере, одна такая улучшающая передача выполнена для любого вектора из WI.

Далее новое отношение предпочтения $\overset{t}{\succ}$ строится следующим образом:

$$Z \overset{t}{\succ} W \leftrightarrow [\exists W' \in WE : Z \overset{p}{\succ} W'] \vee [\exists W'' \in WI : Z \underset{\sim}{\overset{p}{\succ}} W''] . \qquad (1.5.7)$$

Здесь через $\left(\underset{\sim}{\overset{p}{\succ}} \right)$ обозначено нестрогое предпочтение вида

$$Z \underset{\sim}{\overset{p}{\succ}} W \leftrightarrow \forall i \in [1:m] : z_i \geq w_i .$$

Определение (5.7) имеет весьма простой смысл. Если вектор Z строго лучше, чем некоторый вектор W′, эквивалентный W, или нестрого лучше, чем некоторый вектор W″, который, в свою очередь, строго лучше, чем W, то полагаем, что Z строго лучше W.

«Прямой» метод построения рассмотренного отношения доминирования $\left(\overset{t}{\succ}\right)$ в общем случае является практически нереализуемым из-за непреодолимых алгоритмических трудностей в общем случае и значительных вычислительных затрат в случаях когда само построение алгоритмически возможно. Уже отмечалось, в частности, что множества WE, WI практически всегда будут содержать бесконечное число элементов и о каких-то процедурах простого перебора не может быть и речи. Поэтому для реализации построения отношения доминирования $\left(\overset{t}{\succ}\right)$ был предложен иной подход, реализованный в системе Quick Choice.

Частным случаем изложенного метода t-упорядочения является метод, предложенный В.В. Подиновским (далее – метод П-упорядочения). Он основан на том, что мы имеем возможность формировать множества, аналогичные WE и WI с помощью перестановок численных значений оценок между равноценными и неравноценными критериями. Например, пусть дана векторная оценка

$Z = (5, 10, 6)$

и известно, что $f_1 > f_2$. Тогда по Подиновскому оценка

$Z' = (10, 5, 6)$,

полученная из Z перестановкой чисел 5, 10, будет признана лучшей, чем Z, так как на место более «важного» критерия f_1 пришло большее значение (10 вместо 5). Если бы критерии f_1 и f_2 были равноценными, то оценки Z, Z′

считались бы эквивалентными.

Очевидно, в методе П-упорядочения множества WE, WI оказываются конечными и могут быть практически построены без особых вычислительных проблем.

Недостатком метода П-упорядочения является его недостаточная «мощность». Например, пусть ставится задача сравнения двух векторных оценок

$W = (7, 9, 6)$,

$Z = (5, 10, 6)$

при наличии ординальной информации $f_1 > f_2$. Эти оценки, очевидно, несравнимы по Парето. Несравнимы они и по методу П-упорядочения (никакие перестановки численных значений оценок между f_1, f_2 не приводят к их сравнимости по Парето). В то же время легко видеть, что согласно методу t-упорядочения для

$Z' = (6, 9, 6)$,

полученной из Z с помощью переноса $\delta = 1$ со второй позиции в первую, мы имеем

$$Z' \succ Z,\ \ W \succ Z' \succ Z$$

и, следовательно,

$$W \overset{t}{\succ} Z.$$

В то же время с позиций «физического смысла» метод t-упорядочения представляется столь же естественным, что и метод Подиновского.

1.5.3 Задачи с малым числом критериев и альтернатив

Решается многокритериальная задача

$$f_i(x) \to \max, \quad i = 1, ..., m, \tag{1.5.8}$$

где $x \in X = \{x_1, x_2, ..., x_n\}$.

Таким образом, X – конечное множество, содержащее n элементов x_i. Предполагается, что числа m, n относительно невелики, так как именно они в рассматриваемых ниже методах будут определять трудоемкость диалога с пользователем в реальном времени по извлечению дополнительной информации о задаче.

Ниже будут рассмотрены метод Саати и метод Коггера и Ю.

Проблема ранжирования объектов по «важности». Матрица попарных сравнений

Пусть задано конечное множество объектов

$P = \{p_1, p_2, ..., p_m\}$

и нам необходимо построить вектор

$\alpha = (\alpha_1, \alpha_2, ..., \alpha_m)$

с неотрицательными вещественными компонентами α_i, такими, что

$$\sum_{i=1}^{m} \alpha_i = 1$$

Числа α_i будем интерпретировать как весовые коэффициенты, определяющие «важность» или «полезность» объектов p_i. Чем большее значение α_i соотносится с объектом p_i, тем выше «полезность» этого объекта. Например, в исходной многокритериальной задаче (1.5.8) ранжироваться могут частные критерии f_i, а также непосредственно альтернативы x_i из X.

Основным объектом в рассматриваемых методах является треугольная матрица S, называемая здесь матрицей попарных сравнений:

$$S = \begin{bmatrix} 1 & \alpha_{12} & \alpha_{13} & \dots & \alpha_{1m} \\ 0 & 1 & \alpha_{23} & \dots & \alpha_{2m} \\ 0 & 0 & 1 & \dots & \alpha_{3m} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 1 & \alpha_{m-1,m} \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Элемент $\alpha_{ij} = \alpha_i / \alpha_j$ матрицы интерпретируется как коэффициент превосходства i-го объекта p_i над j-ым объектом p_j из множества P. Если $\alpha_i / \alpha_j > 1$, то объект p_i «важнее» объекта p_j и т.д.

Предполагается, что пользователь или ЛПР имеет возможность отвечать на вопросы типа: «Во сколько раз объект p_i превосходит объект p_j по важности?»

Коэффициенты α_{ij} могут выбираться пользователем из фиксированной балльной шкалы, например (Саати):

$$\left\{ \frac{1}{9}, \frac{1}{8}, \frac{1}{7}, \frac{1}{6}, \frac{1}{5}, \frac{1}{4}, \frac{1}{3}, \frac{1}{2}, 1, 2, 3, 4, 5, 6, 7, 8, 9 \right\}.$$

Для облегчения работы пользователя целые числа шкалы могут получать соответствующую смысловую интерпретацию, например:

1 – равная важность;

3 – слабое превосходство;

5 – сильное превосходство;

7 – очень сильное превосходство;

9 – абсолютное превосходство;

2, 4, 6, 8 – промежуточные случаи.

Очевидно, что шкала должна содержать (и содержит) соответствующие обратные значения.

Методы Саати и Коггера и Ю

Задача состоит в отыскании вектора весовых коэффициентов

$\alpha = (\alpha_1, \alpha_2, ..., \alpha_m)$

по известной матрице попарных сравнений S.

Согласно методу Саати по треугольной матрице S строится следующая полнозаполненная матрица $\bar{S}$:

$$\bar{S} = \begin{bmatrix} 1 & \alpha_{12} & \alpha_{13} & \dots & \alpha_{1m} \\ \alpha_{21} & 1 & \alpha_{23} & \dots & \alpha_{2m} \\ \dots & \dots & \dots & \dots & \dots \\ \alpha_{m1} & \alpha_{m2} & \alpha_{m3} & \dots & 1 \end{bmatrix},$$

где элементы нижней треугольной части α_{ij} $(i > j)$ матрицы удовлетворяют соотношениям

$\alpha_{ij} = 1/\alpha_{ji}$.

Легко доказать, что искомый вектор

$\alpha = (\alpha_1, \alpha_2, ..., \alpha_m)$

является собственным вектором матрицы $\bar{S}$, соответствующим максимальному собственному числу матрицы $\lambda = m$ и может быть найден как решение системы уравнений:

$$\bar{S}\alpha = \lambda_{max}\alpha. \qquad (1.5.9)$$

Существует единственное решение данной системы линейных алгебраических уравнений, удовлетворяющее условию

$$\sum_{i=1}^{m} \alpha_i = 1$$

Если матрица системы (1.5.9) задана неточно, то предлагается численно определять ее максимальное собственное число и соответствующий собственный вектор.

Собственно метод Саати сводится к следующему.

Предполагается, что частные критерии f_i не обязательно являются числовыми функциями и могут иметь качественный неформальный характер. В этом случае для каждого частного критерия f_i ставится задача ранжирования объектов $\{x_1, x_2, \ldots, x_n\}$ с построением на основе диалога с пользователем соответствующей матрицы попарных сравнений и определением вектора весов

$$\alpha^i = (\alpha_1^i, \ldots, \alpha_n^i).$$

Полученные числа α_j^i интерпретируются как значения $f_i(x_j)$, $j = \overline{1,n}$. Таким образом, каждая альтернатива получает уже числовую оценку по каждому из частных критериев.

Далее осуществляется аналогичная операция по ранжированию самих частных критериев по важности с построением вектора весов

$$\beta = (\beta_1, \ldots, \beta_m).$$

В качестве оптимальной альтернативы (их может быть несколько) выбираем

$$x^* = \arg\max_i J(x_i),$$

где

$$J(x_i) = \sum_{k=1}^{m} \beta_k f_k(x_i), \quad i = \overline{1,n}; \quad f_k(x_i) = \alpha_i^k.$$

Метод Коггера и Ю отличается от метода Саати тем, что для нахождения вектора весовых коэффициентов ранжируемых объектов используется не система уравнений (5.9), а система вида

$$TS\alpha = \alpha, \tag{1.5.10}$$

где S – треугольная матрица попарных сравнений, а

$T = \mathrm{diag}\,[1/m, 1/m\text{-}1, \ldots, 1]$.

В данном случае матрица системы – треугольная, что облегчает решение задачи.

Обсуждение

При практическом использовании рассмотренных методов возникает целый ряд вопросов.

В стандартных вариантах алгоритмов предполагается, что все элементы матрицы попарных сравнений S определяются независимо в результате диалога, что может приводить к противоречивости ответов пользователя в смысле нарушений очевидных равенств вида

$$\alpha_{ik}\alpha_{kj} = \alpha_{ij}\left(\frac{\alpha_i}{\alpha_k}\cdot\frac{\alpha_k}{\alpha_j} = \frac{\alpha_i}{\alpha_j}\right).$$

Если, как это и рекомендуется авторами, продолжать решать системы (5.9), (5.10), то непонятен смысл механизма обработки заведомо противоречивой информации.

Таким образом, процедура независимой оценки элементов α_{ij} вызывает определенные сомнения и трудности.

С другой стороны, если полагать, что мы должны в конечном итоге получить непротиворечивую информацию от пользователя – может быть, с помощью дополнительных уточняющих вопросов – то тогда можно поступать значительно проще.

В этом случае можно вообще отказаться от решения каких-либо систем уравнений. Действительно, для непосредственного определения

$\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_m)$

достаточно, например, получить от пользователя цепочку коэффициентов превосходства вида

$$\alpha_{12}, \alpha_{13}, \ldots, \alpha_{1m}$$

или

$$\alpha_{12}, \alpha_{23}, \ldots, \alpha_{m-1,m}. \qquad (1.5.11)$$

При этом существенно сокращается трудоемкость диалога и достигается непротиворечивость информации.

Могут быть предложены процедуры получения от пользователя избыточной информации для уточнения его оценок на основе определенных механизмов усреднения, но это – предмет отдельного рассмотрения.

Продолжим обсуждение. Ясно, что применение фиксированной балльной шкалы Саати со смысловой интерпретацией в определенной степени провоцирует пользователя на дачу противоречивых в указанном выше смысле ответов. Действительно, пусть, например, объект p_1 абсолютно превосходит объект p_2 и $\alpha_{12} = 9$ согласно шкале. Пусть также объект p_2 абсолютно превосходит объект p_3 и, следовательно, $\alpha_{23} = 9$. Спрашивается, какое число назовет пользователь в качестве α_{13} – тоже 9 или 81? Последнее число в шкале отсутствует.

По-видимому, в подобных процедурах целесообразно использовать так называемые «транзитивные» шкалы типа:

слабое превосходство = a,

сильное превосходство = aa, (1.5.12)

очень сильное превосходство = aaa,

абсолютное превосходство = aaaa и более.

В этом случае мы в значительной степени будем избавлены от весьма неожиданных согласно шкале Саати утверждений типа: если объект p_1 слабо превосходит объект p_2 ($\alpha_{12} = 3$), а объект p_2, в свою очередь, слабо превосходит объект p_3 (α_{23} =3), то объект p_1 абсолютно превосходит объект p_3 (α_{13} =9)!

Согласно шкале (1.5.12) комбинация двух слабых превосходств дает сильное превосходство, два сильных превосходства дают абсолютное превосходство и т.д. Получаемые согласно этой шкале результаты имеют весьма понятную смысловую интерпретацию. Выбор конкретного значения для базы шкалы «а» - это отдельный вопрос. Можно, например, положить а = 1.5 или а = 2 с получением следующих двух шкал:

Таблица 1.21

Смысловая интерпретация	а = 1.5	а = 2
слабое превосходство	1.5	2
сильное превосходство	2.25	4
очень сильное превосходство	3.38	8
абсолютное превосходство	5.06 и более	16 и более

Простой алгоритм выбора

В данном разделе предлагается простой алгоритм выбора на основе информации (1.5.11). Поясним принципиальную схему метода на конкретном примере.

Рассмотрим проблему выбора научного руководителя студентом старшего курса. Глобальный показатель «качества», характеризующий правильность выбора, будем связывать с <u>общим удовлетворением работой</u> в конкретной научной группе. Этот показатель является достаточно расплывчатым и неопределенным, поэтому используются соответствующие критерии-заместители, и задача трансформируется к некоторой многокритериальной задаче. Будем рассматривать следующие частные критерии оптимальности, характеризующие в совокупности исходный глобальный показатель:

1) Перспективность проводимых в группе исследований с позиций последующего трудоустройства (f_1).
2) Личный интерес студента к проводимым исследованиям (f_2).
3) Возможность получения дополнительной заработной платы в процессе обучения (f_3).
4) Связь научной группы с конкретными фирмами, принимающими на работу молодых специалистов (f_4).
5) Профессионализм, характер и человеческие качества лично научного руководителя (f_5).
6) Состав научной группы и отношения в коллективе (f_6).

Предполагается, что все введенные частные показатели необходимо максимизировать, т.е. большему значению каждого показателя будет соответствовать более желаемое состояние для студента.

Рассмотрим ситуацию выбора, когда имеется три потенциальных научных руководителя, обозначенных буквами A, B, C.

Будем следовать предлагаемому простому алгоритму выбора с транзитивной шкалой и базой a = 2 (см. табл. 1.21). Построим вектор весов для сформулированных частных критериев. Необходимо задать пользователю пять вопросов, и определить в результате вектор коэффициентов превосходства (1.5.11). Будем считать, что по результатам диалога были получены следующие данные:

$\alpha_{12} = 2$, $\alpha_{23} = 4$, $\alpha_{34} = 1/4$, $\alpha_{45} = 1$, $\alpha_{56} = 4$.

Здесь равенство $\alpha_{12} = 2$, например, означает, что частный критерий f_1 в два раза превосходит по «важности» критерий f_2 и т.д.

Воспользовавшись соотношением

$$\alpha_{ij} = \alpha_i/\alpha_j$$

и условием нормированности вектора α, непосредственно получаем:

$\alpha_1 = 0.364$; $\alpha_2 = 0.182$; $\alpha_3 = 0.045$; $\alpha_4 = 0.182$; $\alpha_5 = 0.182$; $\alpha_6 = 0.045$.

Далее переходим к процедуре вычисления значений частных критериев оптимальности, соответствующих трем вариантам A, B, C.

Вначале с помощью того же самого подхода ранжируем варианты A, B, C по критерию f_1 (перспективность исследований). Пусть пользователь указал следующие значения коэффициентов превосходства:

$$\alpha_{12}^1 = 1; \alpha_{23}^1 = 1/2 .$$

Соответствующий вектор весов α^1 имеет компоненты

0.250, 0.250, 0.500,

которые интерпретируются как значения функции f_1 для трех вариантов A, B, C:

f_1 (A) = 0.250; f_1 (B) = 0.250; f_1 (C) =0.500.

Аналогично определяем значения остальных частных критериев для вариантов A, B, C:

$$\alpha_{12}^2 = 2; \alpha_{23}^2 = 2 ;$$

f_2 (A) = 0.571; f_2 (B) = 0.286; f_2 (C) =0.143.

$$\alpha_{12}^3 = 1; \alpha_{23}^3 = 1 ;$$

f_3 (A) = 0.333; f_3 (B) = 0.333; f_3 (C) =0.333.

$$\alpha_{12}^4 = 1/2; \alpha_{23}^4 = 1/4 ;$$

f_4 (A) = 0.091; f_4 (B) = 0.182; f_4 (C) =0.727.

$$\alpha_{12}^5 = 4; \alpha_{23}^5 = 2 ;$$

f_5 (A) = 0.727; f_5 (B) = 0.182; f_5 (C) =0.091.

$$\alpha_{12}^6 = 1/2; \alpha_{23}^6 = 2 ;$$

$f_6(A) = 0.250$; $f_6(B) = 0.500$; $f_6(C) = 0.250$.

Воспользовавшись методом линейной свертки,

$$J(x_i) = \sum_{k=1}^{6} \alpha_k f_k(x_i),$$

получим значения обобщенного критерия оптимальности для трех вариантов $x_1 = A$, $x_2 = B$, $x_3 = C$:

$$J(A) = 0.364 * 0.250 + 0.182 * 0.571 + 0.045 * 0.333 + 0.182 * 0.091 +$$
$$+ 0.182 * 0.727 + 0.045 * 0.250 = 0.370.$$

$$J(B) = 0.364 * 0.250 + 0.182 * 0.286 + 0.045 * 0.333 + 0.182 * 0.182 +$$
$$+ 0.182 * 0.182 + 0.045 * 0.500 = 0.247.$$

$$J(C) = 0.364 * 0.500 + 0.182 * 0.143 + 0.045 * 0.333 + 0.182 * 0.727 +$$
$$+ 0.182 * 0.091 + 0.045 * 0.250 = 0.383.$$

Следовательно, наиболее перспективным с позиций применяемого метода признается выбор руководителя С. Однако видно, что выбор А оказывается почти столь же хорошим.

Замечание. Ясно, что если нет оснований считать множество достижимости рассматриваемой многокритериальной задачи выпуклым, целесообразно вместо линейной свертки в качестве обобщенного критерия использовать свертку Джоффриона, основанную на комбинации линейной и максиминной сверток.

Легко подсчитать, что в предложенном методе общее число сравнений объектов по важности, выполняемых пользователем, равно

$$N_1 = m - 1 + m(n - 1),$$

где m – число частных критериев, n – количество альтернатив.

В стандартных процедурах Саати и Коггера и Ю число сравнений равно

$$N_2 = \frac{m^2 - m}{2} + m\frac{n^2 - n}{2},$$

что может быть существенно больше N_1. Так для $m = 6$, $n = 10$ имеем:

$N_1 = 59$, $N_2 = 285$.

Кроме того, как уже указывалось, при использовании предлагаемого подхода нет необходимости решать какие бы то ни было линейные системы, и искать собственные числа матриц.

1.5.4 Метод ограничений

Простым и часто применяемым методом сжатия множества Парето является метод ограничений.

Решается стандартная многокритериальная задача

$$\begin{aligned} &f_i(x) \to \max, x \in D, \\ &f_i : D \to R, i = 1,...,m, \end{aligned} \tag{1.5.13}$$

где D – произвольное абстрактное множество.

На первом этапе одним из известных методов строится множество Парето P(D). Далее метод ограничений реализуется в соответствии со следующей последовательностью шагов.

Шаг1. Пользователю предлагается назначить нижние допустимые границы t_i для всех m критериальных функций:

$$f_i(x) \geq t_i, \ i = 1, ..., m. \tag{1.5.14}$$

Предполагается, что указанные значения критериальных функций дают удовлетворяющие пользователя варианты.

Шаг2. Строится подмножество множества Парето, состоящее из точек, удовлетворяющих неравенствам (1.5.14):

$D_t = \{x \in P(D) \mid f_i(x) \geq t_i, \ i = 1, ..., m\}$, $D_t \subset P(D)$.

Шаг3. Если D_t – пустое множество, то пользователю предлагается ослабить требования с помощью уменьшения какого-то из чисел t_i. Далее переходим к шагу 2. Если D_t непусто – переходим к шагу 4.

Шаг4. Выбирается $\forall\ x \in D_t$ и предъявляется пользователю в качестве кандидата на «решение» задачи (1.5.13). Если решение удовлетворяет пользователя, то процесс завершается. В противном случае переходим к шагу 5.

Шаг5. Пользователю предлагается назначить новую (увеличенную) нижнюю границу по одному из критериев и осуществляется переход к шагу 2.

Как правило, в процессе диалога пользователь получает дополнительную информацию о задаче в виде диапазонов изменения векторных оценок для элементов множества Парето или множеств D_t. Иногда целесообразно иметь информацию о так называемой «идеальной» точке, соответствующей лучшим возможным значениям по всем критериям (такая точка реально обычно не существует, но позволяет оценить множество допустимых верхних границ). Кроме того, в процессе решения «хорошая» СППР должна информировать пользователя о структуре множеств P(D), D_t. Например, для случая конечного множества D пользователь на шаге 4 возможно захочет уточнить свой выбор (даже если он его первоначально и удовлетворил) с помощью просмотра других точек из D_t, удовлетворяющих той же системе ограничений. Должна быть обеспечена также возможность возврата назад к прежним вариантам и возможность выбора новых начальных условий, новых критериев и т.д.

Метод ограничений целесообразно использовать на завершающей стадии процесса выбора, например, после применения метода t-упорядочения. Тогда в качестве исходного для метода ограничений множества будет использоваться не P(D), а некоторое его подмножество, что сократит наиболее трудоемкую диалоговую часть процедуры выбора.

Непосредственно в методе ограничений какая-либо ординальная информация не используется, а сокращение исходного множества альтернатив производится в процессе поступления дополнительной информации от пользователя в виде последовательности наборов нижних границ $\{t_i\}$.

1.5.5 Рандомизированные стратегии принятия решений

В данном разделе мы рассмотрим еще один подход к интерпретации ординальной информации пользователя об относительной важности частных критериев оптимальности многокритериальной задачи (1.5.13).

Основная идея заключается в следующем. Рассматривается какая-либо скалярная свертка (обычно линейная) векторного критерия оптимальности

$f = (f_1, ..., f_m)$ с весовыми коэффициентами $\{\alpha_i\}$. Например, вводится следующий «обобщенный», «глобальный», «сводный» и т.п. критерий:

$$J(x,\alpha) = \langle f(x),\alpha \rangle = \sum_{i=1}^{m} \alpha_i f_i(x), \qquad (1.5.15)$$

где $\alpha=(\alpha_1, ..., \alpha_m)$ – вектор неотрицательных весовых коэффициентов, удовлетворяющих условию

$\sum \alpha_i = 1$.

Далее, поступающая от пользователя ординальная информация вида

$f_j \succ f_k$ или $f_\ell \sim f_p$

интерпретируется с помощью соответствующих неравенств:

$$\alpha_j \geq \alpha_k \text{ или } \alpha_\ell = \alpha_p. \qquad (1.5.16)$$

В результате исходное множество весовых коэффициентов

$$A = \{\alpha = (\alpha_1, ..., \alpha_m) | \alpha_i \geq 0, \ \sum \alpha_i = 1\}$$

сужается до некоторого подмножества $\overline{A} \subset A$. Для любого $\alpha \in \overline{A}$ выполняются все неравенства (1.5.16). В частном случае может быть $\overline{A} = A$.

Считаем, что α – случайный вектор с равномерной плотностью распределения во множестве $\overline{A}$. Тогда значение $J(x,\alpha)$ для любого фиксированного x также будет случайной величиной. Оптимальное значение x^* предлагается выбирать из условия

$$x^* = \arg\max_x M\{J(x, \alpha)\}, \qquad (1.5.17)$$

где $M\{\cdot\}$ – знак математического ожидания.

Описание основной идеи закончено. Дадим некоторые комментарии.

Если используется линейная свертка (1.5.15), то

$$M\{J(x, \alpha)\} = \langle f(x), M\{\alpha\} \rangle$$

и задача сводится к определению математических ожиданий весовых коэффициентов, одних и тех же для любого x. Полученные математические ожидания $M\{\alpha_i\}$ будут играть роль весовых коэффициентов в стандартном методе линейной свертки (см. разд. 1.2.3.) со всеми вытекающими особенностями. В частности, при невыпуклой структуре множества f(D) заведомо (независимо от ординальной информации пользователя) выпадают из рассмотрения эффективные решения, лежащие на невыпуклых участках границы. Последнее замечание определяет существенный и определяющий недостаток рассмотренного подхода. Однако, если f (D) – выпуклое множество, то отмеченных трудностей не возникает.

Пример. Рассмотрим многокритериальную задачу выбора (1.5.13) с m = 2. Пусть множество достижимости f (D) невыпукло и представлено на рис. 1.34.

Множество Парето в пространстве оценок, очевидно, совпадает с отрезком границы [a, c]. При условии «одинаковой важности» критериев f_1, f_2 пользователь в качестве окончательного решения, вполне возможно, выбрал бы точку «b», где $f_1 = f_2$. Однако, как следует из рассмотрения в разд. 1.2.3., при любых наборах весовых коэффициентов α с помощью конструкции (1.5.15) могут быть получены только точки «a» и «c». Точка «b» не будет решением ни при каких видах ординальной информации пользователя.

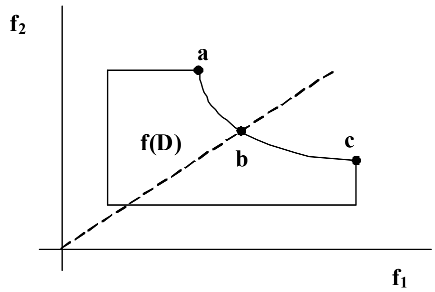


Рис. 1.33

Основная проблема при численной реализации данного метода заключается в организации процедуры «вбрасывания» заданного количества m-мерных случайных точек в построенную, обычно достаточно сложную, область $\overline{A}$. В простейших случаях результат может быть получен аналитически.

<u>Пример.</u> Пусть решается многокритериальная задача (1.5.13) с m = 2 и пусть из ординальной информации, заданной пользователем, получим $\alpha_1 \geq \alpha_2$. Тогда множества A и $\overline{A}$ могут быть представлены с помощью рис. 1.35.

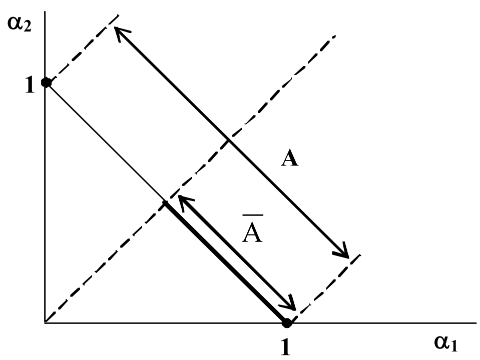


Рис. 1.34

В данном случае, очевидно,

$$M\{\alpha_1\} = 0.75;\; M\{\alpha_2\} = 0.25.$$

(Это координаты середины отрезка $\overline{A}$).

Таким образом, мы имеем более чем однозначную интерпретацию достаточно слабого утверждения пользователя о том, что $\alpha_1 > \alpha_2$. В этом также заключается определенная особенность метода: он, может быть, излишне категоричен.

В общем случае произвольной структуры множества f (D), когда его выпуклость не может быть гарантирована, целесообразно в соответствии с рекомендациями разд. 1.2.3. использовать метод лексикографической оптимизации:

$$J(x,\alpha) = \left\{ \min_i \alpha_i f_i(x), \sum_{i=1}^{m} f_i(x) \right\} \to \max_{x \in D} \tag{1.5.18}$$

или

$$J(x,\alpha) = \sum_{i=1}^{m} f_i(x) \to \max_{x \in X(\alpha)},$$
$$X(\alpha) = \operatorname{Arg} \max_{x \in D} \min_i \alpha_i f_i(x).$$

При этом оптимальная точка x* также находится согласно выражению (1.5.17).

Простейшая прямая реализация изложенного метода на основе свертки (1.5.18) состоит в следующем. Пусть множество D – конечно. Для каждого $x \in D$ генерируется последовательность α^1, ..., α^N векторов $\alpha \in \overline{A}$ в соответствии с равномерной плотностью распределения вероятностей. Для каждого α^k из этой последовательности вычисляется значение J (x, α^k) по формуле (1.5.18). Далее вычисляется среднее значение J как оценка для M {J(x, α)} при данном x. В соответствии с полученными значениями M {J (x, α)} для всех x выбирается наилучшая точка x* согласно представлению (1.5.17).

1.6 КОММЕНТАРИЙ

Мы изложили некоторые из важнейших проблем, входящих в теорию принятия решений. Основное содержание этой дисциплины схематично представляется в следующем виде:

1) математическое описание – создание модели ситуации выбора;

2) анализ неопределенностей, формализация понятия цели, формирование критериев и целевых функций;

3) решение возникающих оптимизационных и других математических задач.

Приведенная последовательность действий достаточно условна, так как указанные разделы тесно переплетаются в процессе решения конкретной практической задачи. Однако основные моменты исследования здесь отражены.

Построение модели ситуации выбора, – а с него начинается любое исследование – требует глубокого понимания специфики процесса. Мы по существу рассмотрели два языка, на которых может быть сформулирована проблема ПР: язык бинарных отношений и критериальный язык описания выбора. Язык бинарных отношений является более общим по сравнению с критериальным языком, поскольку не требует численной оценки качества каждой отдельно взятой альтернативы. Напротив, критериальный язык применяется, если сравнение альтернатив сводится к сравнению соответствующих им чисел. При этом мы допускали многокритериальность, т.е. возможность оценки альтернативы с помощью не одного числа, а нескольких.

Еще более общим языком, который нами ранее не рассматривался, является язык функций выбора. Пусть U – фиксированная совокупность непустых подмножеств множества альтернатив A. Функцией выбора (на U) называется отображение C, сопоставляющее всякому множеству $B \in U$ подмножество $C(B) \subseteq B$ (т.е. подмножество «выбранных», «наиболее предпочтительных» альтернатив). В частном случае, если задано отношение предпочтения R, функцию выбора можно определить равенством $C(B) = Max_R B$. В этом случае множество C (B) совпадает с ядром – множеством максимальных элементов из B по отношению R. В результате мы приходим к уже изученной постановке задаче ПР.

Следовательно, задачи, сформулированные на языке бинарных отношений, описываются также с помощью функций выбора. Обратный вопрос о возможности введения бинарного отношения предпочтения по заданной функции выбора оказывается значительно более тонким.

Учитывая, что в настоящее время в приложениях наиболее часто применяется критериальный язык описания предпочтений, можно считать, что следующая важнейшая группа проблем – это формирование критериев и целевых функций (функционалов). Эти проблемы, как мы видели, решаются в тесной связи с методами преодоления различных видов неопределенностей на основе тех или иных гипотез.

Сделаем несколько общих, но важных для практики замечаний относительно применения оптимизационного подхода в теории ПР.

Критерий оптимальности – это, вообще говоря, некоторое правило, позволяющее отличать «оптимальные» решения от «неоптимальных». Простейший критерий связан, например, с заданием на множестве альтернатив некоторого функционала $J : A \to E$ и с утверждением, что оптимальными признаются такие альтернативы из множества A, которые доставляют максимум функционалу J на A.

Функционал J при этом называется целевым функционалом (или целевой функцией). Довольно часто допускается вольность речи и сам функционал J называется критерием. При этом обычно говорят о задаче максимизации (или минимизации) «критерия J».

Важно также отличать цель принятия решений от отражающего эту цель критерия (или критериев). Приведем пример. Общую цель службы скорой медицинской помощи можно сформулировать как «доставка больного в больницу в наилучшем состоянии, возможном при данных обстоятельствах». Эту цель достаточно трудно формализовать. Поэтому при анализе систем скорой помощи используется критерий «время реагирования» (в технических системах – это «время отклика»). Этот критерий определяется как время, истекшее между получением вызова скорой помощи и прибытием машины к больному. Другой критерий, используемый при исследовании эффективности работы службы скорой помощи, – это «время доставки» – время между получением вызова и прибытием больного в больницу. Смысл введения этих критериев состоит в предположении, что более краткие времена «реагирования» и «доставки» способствуют достижению общей цели системы скорой помощи, хотя полностью ее и не отражают (успех дела, в частности, зависит от квалификации персонала, его умения оказать эффективную немедленную помощь и т.д.).

Изложенная ситуация оказывается достаточно общей: критерии и отвечающие им целевые функции характеризуют цель лишь косвенно, иногда лучше, иногда хуже, но всегда приближенно. В этой связи обычно говорят о критериях-заместителях, т.е. таких критериях, которые лишь косвенно характеризуют степень достижения связанной с ними цели. В сущности, все критерии являются «заместителями», так как ничто не поддается абсолютно точному измерению.

Все предыдущее изложение, по-видимому, в достаточной степени прояснило тот факт, что среди используемых в теории ПР математических методов особую роль играют методы решения оптимизационных задач. Они составляют фундамент системы математического обеспечения проблем принятия решений. Соответствующие вопросы, касающиеся проблемы конечномерной скалярной оптимизации, будут изложены во второй части книги.

2 ЭКСПЕРТНЫЕ СИСТЕМЫ ПРИНЯТИЯ РЕШЕНИЙ.

2.1 Введение.

Экспертная система (ЭС) представляет из себя компьютерную программу, позволяющую автоматизировать достоверные рассуждения человека-эксперта в конкретной предметной области. ЭС – это диалоговая система; содержание и форма диалога соответствует «беседе» эксперта с «заказчиком» или пользователем системы с целью получения экспертных заключений по обсуждаемой проблеме. В результате такой беседы человек-эксперт приходит к определенным выводам и рекомендациям, позволяющим ответить на основной вопрос пользователя. В частности, пользователя может интересовать проблема выбора решения из заданного множества альтернатив. К такому же результату мы приходим, общаясь и с ЭС, которая моделирует поведение человека-эксперта. В качестве традиционного примера предметной области, где целесообразно применение ЭС является область медицинской диагностики. Основываясь на данных анализов и внешних симптомах проявления болезни, а также на основе имеющейся дополнительной информации ЭС должна указать наиболее правдоподобный диагноз (из заданного множества диагнозов), моделируя рассуждения человека-эксперта, в данном случае – лечащего врача или врача-диагноста.

Основная цель, достигаемая при использовании ЭС, состоит в тиражировании знаний высококвалифицированных экспертов. Это приводит к удешевлению процесса экспертизы (обращение к высококвалифицированному человеку-эксперту не всегда возможно и стоит достаточно дорого), а также, вообще говоря, к повышению достоверности и надежности результатов экспертизы. Последний аспект особенно ярко проявляется в задачах принятия решений в условиях критических ситуаций, когда требуется быстро и безошибочно указать способ поведения реального объекта (например, человека). Как правило, в жестких временных рамках даже высококвалифицированный человек-эксперт (например, офицер наведения ракет в системах противовоздушной обороны) подвержен влиянию различных психологических факторов, затрудняющих процесс выработки рациональных решений. Экспертная же система за счет своего быстродействия и отсутствия влияния нежелательных человеческих факторов позволяет быстро и непредвзято оценить результаты анализа обстановки и выработать разумную ответную реакцию.

При этом важно понимать, что построение и последующее применение ЭС возможно только при условии наличия эксперта (или группы экспертов), знания которого (или которых) удалось формализовать с помощью соответствующей «базы знаний». Помимо знания основных фактов и данных из конкретной предметной области эксперт владеет также своей логикой рассуждений, которая также должна быть отражена при построении ЭС (механизм вывода).

В настоящее время ситуация такова, что существующие и проектируемые ЭС направлены на принятие решений в достаточно узких предметных областях, что, вообще говоря, соответствует практике использования живых экспертов.

Далее в части 3 будут рассмотрены основные сведения по внутреннему устройству ЭС. Приводимых сведений будет достаточно не только для понимания смысла работы реальных ЭС, имеющихся на рынке современных информационных технологий, но и для создания своих собственных ЭС, предназначенных для решения возникающих специальных задач. Изложенный материал, конечно, не охватывает всего многообразия способов представления знаний и организации процедуры логического вывода. Многие вопросы не рассмотрены. Однако приводимые сведения, по мнению автора, могут явиться основой для более глубокого освоения предмета с помощью многочисленной специальной литературы.

2.1.1 Назначение и области применения экспертных систем.

В настоящее время системы поддержки принятия решений в виде экспертных систем (ЭС) широко используются в различных областях. Появилась и развивается специальная индустрия по разработке и внедрению ЭС.

Основное назначение ЭС состоит в решении неформализованных задач выбора, являющихся трудными для традиционных методов математического анализа и традиционных методов программирования.

Наибольшее распространение ЭС получили в таких областях, как

проектирование заказных интегральных схем;

автоматизация программирования на основе применения современных CASE-систем и окружений разработки больших программных проектов;

военные приложения;

здравоохранение;

риэлтерская деятельность по подбору и продаже объектов недвижимости, рынок недвижимости;

финансовый рынок и рынок ценных бумаг;

автоматизированное комплексирование заказных компьютерных систем, в том числе офисных;

принятие решений в кризисных ситуациях;

охрана правопорядка;

современные информационные образовательные технологии, контроль знаний обучающихся;

задачи планирования и рационального распределения ресурсов и т.д.

По своему смыслу многие ЭС, применяемые в вышеперечисленных областях, могут быть отнесены к одному из следующих классов.

Диагностирующие и управляющие системы. Основная задача диагностики может быть в общем виде сформулирована следующим образом. Пусть S – некоторая диагностируемая система (рис. 2.1).

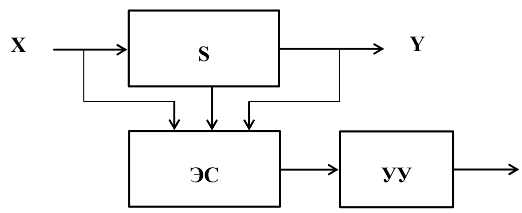


Рис. 2.1

X – входные, а Y – выходные сигналы диагностируемой системы. На основе анализа в реальном времени информации о входных и выходных сигналах, а также информации о внутреннем состоянии системы S диагностирующая экспертная система ЭС должна делать заключения о «правильности» функционирования S. При возникновении «нештатных», критических ситуаций они должны фиксироваться ЭС. Кроме того, должны определяться места «неисправностей» в системе S и выдаваться рекомендации устройству управления УУ (это может быть техническое устройство, человек, группа лиц, государственный орган и т.п.) с целью вывода S из кризисной ситуации. Таким образом, данная ЭС будет выполнять функции управления.

Как уже указывалось, под S может пониматься, в частности, пациент некоторого лечебного учреждения, филиал некоторого банка, автомобиль с компьютером на борту или большой программный комплекс, функционирующий в реальном времени. Таким образом, здесь речь идет, по существу, о некоторой системе мониторинга (слежения). В этом случае диагностика состояния S и соответствующая интерпретация поступающей информации происходят в реальном времени. ЭС осуществляет сигнализацию о выходе параметров слежения системы S за допустимые пределы, анализирует возможные причины и выдает советы о целесообразной реакции на сложившуюся ситуацию. Подобные системы мониторинга применяются в медицине, экономике, военной области, в системах управления ядерными реакторами и т.д.

В виде соответствующих ЭС существуют и применяются статические диагностирующие системы, например, системы по определению и устранению неисправностей автомобиля при обращении пользователя на станцию технического обслуживания.

Прогнозирующие системы. Прогнозирующие ЭС могут также применяться в различных областях. Основная задача прогнозирующей ЭС заключается в анализе развития ситуации (некоторой системы S) за определенный отрезок времени и выдачи соответствующих выводов и прогнозов о правдоподобных путях развития этой ситуации в будущем. Например, введение некоторых новых законов в стране требует предварительного анализа возникающих последствий и оценки желательности этих последствий. Точно так же анализ определенных изменений на бирже позволяет указать возможное развитие ситуации в будущем, что может повлиять на принимаемые в настоящий момент решения. Безусловно, существуют прогнозирующие системы, основанные на регулярных математических методах, связанных, в частности, с методами анализа временных рядов, однако в ряде случаев прогноз может быть осуществлен только на основе знаний экспертов в конкретной предметной области и это область применимости ЭС.

<u>Планирующие системы.</u> Предназначены для создания плана реализации последовательности действий для достижения поставленных целей. Примерами ЭС, занятых планированием, могут служить системы формирования плана проведения боевой операции в заданных условиях или системы создания плана действий при комплексировании сложной информационной (например, телекоммуникационной) системы по основным требованиям, сформулированным заказчиком.

В качестве примера планирующей экспертной системы можно также привести ЭС, обслуживающую какое-то количество рабочих станций (терминалов) в торговом зале и позволяющую покупателям компьютеров спланировать покупку – выбрать в диалоговом режиме конфигурацию компьютера, в наибольшей степени соответствующую целям и финансовым возможностям каждого отдельного покупателя.

Аналогичные ЭС могут применяться при продаже любых достаточно сложных объектов и услуг, например, на рынке недвижимости.

<u>Интерпретирующие (анализирующие) системы</u> осуществляют анализ («расшифровку») поступающей информации о состоянии некоторой системы или объекта и затем дают описание реальной ситуации на стандартном для данной прикладной области языке. Например, военные интерпретирующие системы могут использоваться для идентификации целей на основе данных радиолокационной разведки.

Можно и далее продолжить описание областей применения и функций ЭС. Ясно, что вышеприведенная классификация является в достаточной степени условной и неполной, и одна и та же ЭС может быть описана (проинтерпретирована!) с различных позиций и отнесена сразу к нескольким классам. Однако бесспорно, что области практического применения ЭС могут быть связаны с такими ключевыми словами, как интерпретация, управление, диагностика, прогноз, проектирование, планирование, наблюдение, отладка, ремонт, обучение и т.д.

2.1.2 Структура экспертной системы.

ЭС содержит следующие основные компоненты:

- база знаний;
- механизм вывода (средство компьютерного мышления).

Основной процесс заключается в применении механизма вывода к исходным знаниям с целью получения результирующих знаний, представляющих интерес для пользователя ЭС.

Кроме основных компонентов ЭС включает дополнительные подсистемы, обеспечивающие: общение с пользователем, перенос знаний от эксперта в компьютерную программу, объяснение и обоснование результатов вывода и т.д.

Типовая структура ЭС и схема взаимодействия участников процесса построения и использования ЭС представлены на рис. 2.1.2. Собственно ЭС обведена пунктиром.

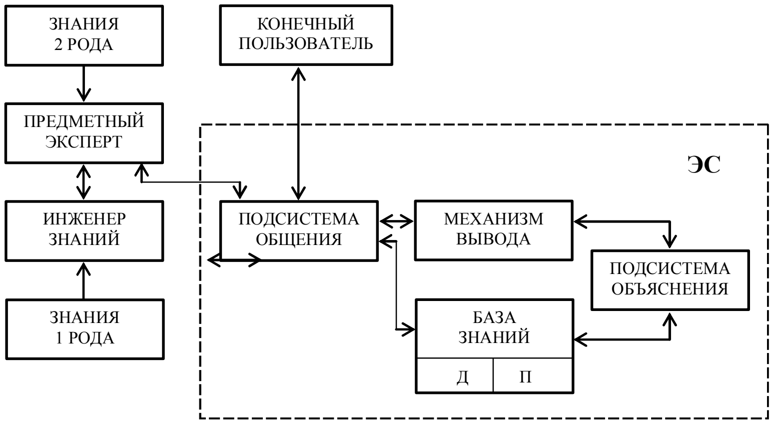


Рис. 2.1

Знания, которыми владеет эксперт в конкретной предметной области делятся на **декларативные** (Д) и **процедурные** (П). Декларативные знания (или **факты**) дают описание фактов и явлений внешнего мира, относительно которых можно установить, есть они в наличии или нет. Например, у больного температура может быть повышенной или нормальной (есть температура или нет, как обычно говорят). Данный конкретный человек может иметь высшее образование или не иметь и т.п.

Процедурные знания заключаются в правилах манипулирования фактами для получения заключений, приводящих к новым знаниям (как декларативным, так, возможно, и процедурным). Весьма распространенная форма представления процедурных знаний связана с уже упоминавшейся в этой книге **продукцией** ЕСЛИ ... ТО... . Например, запись

ЕСЛИ А ТО В,

где А и В – факты, позволяет по факту А установить наличие факта В, если указанная продукция (элемент множества процедурных знаний) присутствует в базе знаний ЭС.

На рис. 2.2. представлены знания первого и второго рода. К знаниям первого рода относятся общезначимые, общеизвестные декларативные и процедурные знания, например, отражающие законы сохранения в физике.

Такие знания доступны не только эксперту в данной предметной области (предметному эксперту), но и, в частности, инженеру знаний. Знания второго рода являются в определенном смысле более ценными. Они включают различные know-how, эмпирические и интуитивные соображения, которыми владеет данный предметный эксперт. Так же как и знания первого рода, эти знания могут быть как декларативными, так и процедурными.

Предметный эксперт или просто эксперт – это человек, являющийся признанным специалистом в конкретной предметной области и умеющий (а главное – желающий) ясно объяснить свои методы, приемы и стратегии решения проблем. В ЭС, как правило, моделируются знания одного или нескольких экспертов, а также используются дополнительные доступные знания.

Инженер знаний (инженер по знаниям) должен являться глубоким специалистом в области ЭС и, в частности, он должен владеть полной информацией о конкретной ЭС, так как именно он осуществляет перенос знаний эксперта в ЭС при ее построении и настройке на конкретную предметную область. Инженер знаний общается с экспертами и форматирует полученные знания для их введения в базу знаний ЭС. Кроме того, желательно, чтобы инженер знаний имел достаточно высокий научный и интеллектуальный потенциал для быстрой адаптации к конкретной предметной области. Инженер знаний участвует в разработке конкретной ЭС и в ее последующем сопровождении.

Конечный пользователь использует ЭС по прямому назначению – для получения ответов на свои вопросы из области компетентности данной ЭС. Важное обстоятельство заключается в том, что пользователь, в отличие от инженера знаний, может не быть специалистом в области информатики.

Он общается с ЭС через подсистему общения на языке, максимально приближенном к профессиональному языку в конкретной области экспертизы. Тем более от него не требуется знаний в области программирования вообще и программирования ЭС, в частности.

Как устроены базы знаний и механизм вывода (иногда говорят – машины логического вывода) будет далее показано при рассмотрении конкретных ЭС.

2.1.3 Основные классы и виды экспертных систем.

Классификацию объектов производят по каким-либо признакам. В данном случае мы кратко на эвристическом уровне укажем основные виды ЭС в зависимости от методов представления знаний.

Методы, основанные на правилах – продукционные экспертные системы. Как уже отмечалось, продукционные системы в качестве базы процедурных знаний имеют набор продукций (правил) вида ЕСЛИ А ТО В, где А и В – элементы множества декларативных знаний. (В ряде случаев могут быть использованы и более сложные продукционные структуры). Здесь А называется условием, а В – следствием. Если факт А породил В, то теперь уже В может выступать как условие в новой продукции и т.д. Организуется, так называемая, цепочка логического вывода, которая заканчивается фактом или фактами, играющими роль результатов экспертизы.

С помощью правил-продукций в отличие от традиционных приемов программирования удается реализовать более гибкую стратегию организации ветвления программы (передачи управления), которое должно управляться самими данными. Кроме того, при этом достигается логическая цельность и ясность программы, что важно как для понимания ее работы во время создания самого программного продукта, так и для последующей его модификации в процессе эксплуатации.

Более подробно способы представления знаний в продукционных ЭС, а также различные механизмы вывода рассмотрены в следующей главе.

Фреймы. Под фреймовыми системами понимаются ЭС, основанные на специальных методах представления знаний в виде объектов и отношений между объектами. Представление знаний, основанное на фреймах, является по сравнению с продукционными методами альтернативным способом структурирования, хранения и обработки знаний.

По существу речь идет о хорошо известной технологии объектно-ориентированного программирования, применяемой для целей создания ЭС. Основными понятиями являются понятия структуры, объекта, слота, атрибута, значения.

Структура дает общее описание объекта с указанием списка атрибутов как имен слотов (мест хранения информации). При этом объект выступает как некоторая конкретизация структуры, содержащая уже конкретную информацию (значения) по всем слотам.

Организация представления знаний на основе объектно-ориентированного подхода имеет ряд особенностей, важнейшим из которых является принцип наследования. Во фреймовой ЭС предметная область описывается некоторой иерархической фреймовой структурой, в которой фреймы, занимающие более низкое положение в иерархии, наследуют свойства фреймов более высокого уровня. Это, в частности, позволяет экономить память компьютера (так как исключается дублирование при описании свойств объектов), а также уменьшать вероятность возникновения ошибок и противоречий в системе знаний.

К недостаткам фреймовых систем обычно относят их относительно высокую сложность и низкое быстродействие. Кроме того, достаточно сложным оказывается процесс изменения принятой иерархической структуры (родовидовой иерархии).

Более подробное рассмотрение данных вопросов выходит за пределы данной книги.

Прочие методы. Существует множество других концепций построения ЭС, например, семантические сети и нейлоровские диагностирующие системы. И, если семантические сети тесно связаны с идеями фреймового представления знаний, и далее не рассматриваются, то нейлоровские системы основаны на принципиально ином подходе и далее будут рассмотрены подробно вплоть до описания работающей ЭС.

2.2 Продукционные экспертные системы.

2.2.1 Основные компоненты продукционной экспертной системы.

В соответствии со структурой типовой ЭС (рис. 2.2) рассмотрим особенности представления знаний и механизмы вывода в продукционных системах. Основные компоненты

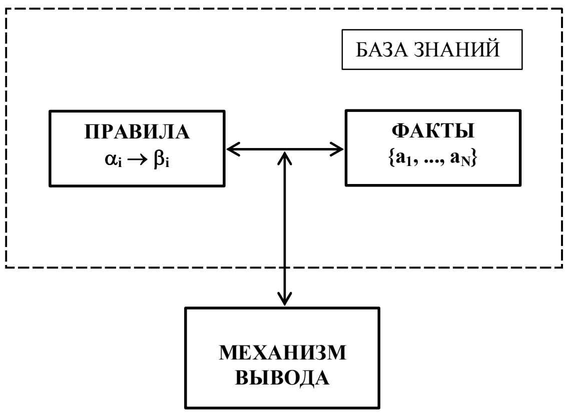


Рис. 2.2

продукционной ЭС изображены на рис. 2.3.

Механизм вывода часто называется интерпретатором правил или планировщиком. Правила – продукции $\alpha_i \rightarrow \beta_i$ интерпретируется с помощью конструкции:

$$\text{ЕСЛИ } \alpha_I \text{ ТО } \beta_I.$$

Здесь α_i и β_i могут достаточно сложным образом зависеть от фактов a_i. Например, можем иметь продукцию вида

$$a_1 \wedge a_3 \wedge a_5 \wedge a_9 \rightarrow a_{10},$$

где $\wedge$ – знак конъюнкции (логическое «и»).

При решении задач оптимизации с помощью некоторой диалоговой системы оптимизации или пакета прикладных программ для выбора метода конечномерной оптимизации без ограничений может применяться ЭС, содержащая, например, следующее правило:

ЕСЛИ	Решаемая задача = задача конечномерной оптимизации без ограничений,
И	Количество переменных = 2,
ТО	Рекомендуемый метод = метод вращения осей Розенброка.

Механизм вывода в продукционной ЭС может быть построен в соответствии с различными принципами. Рассмотрим основную идею.

Мы имеем базовое (неизменное) для данной ЭС множество возможных фактов

$$A = \{a_1, ..., a_n\},$$

формируемое при создании (разработке) ЭС.

При этом важно понимать, что никакие новые факты не могут быть получены в результате работы ЭС. Основная и единственная задача ЭС –

устанавливать определенные связи между фактами для конкретной ситуации, интересующей пользователя ЭС.

Будем различать два подмножества A_0, A_1

$$A = A_0 \cup A_1, \ \ A_0 \cap A_1 = \varnothing$$

исходного множества A. Множество A_1 будет называться множеством констатированных (или помеченных) фактов, а множество A_0 – множеством непомеченных фактов. Иногда множество A_1 называется **рабочим полем** ЭС.

В начале работы ЭС множество A_1 содержит некоторое количество фактов (исходная информация), например,

$$A_1 = \{a_1, a_2, a_3\}.$$

Далее происходит последовательное пополнение множества A_1 за счет элементов множества A_0. Интерпретатор правил сопоставляет левые части продукций

$$\alpha_i \rightarrow \beta_i$$

с имеющимися во множестве A_1 фактами и выполняет то правило, левая часть которого α_i согласуется с фактами из A_1 (оказывается истинной). В результате множество A_1 пополняется за счет фактов, констатируемых в правой части продукции β_i.

Пример.

$$A_1 = \{a_1, a_2, a_3\};$$

найдена продукция

$$P_k: a_1 \wedge a_2 \wedge a_3 \rightarrow a_4 \wedge a_5.$$

Левая часть оказывается согласованной с множеством A_1, продукция выполняется, и новое множество A_1 имеет состав

$$A_1 = \{a_1, a_2, a_3, a_4, a_5\}.$$

Далее процесс повторяется – проверяется согласованность с A_1 уже оставшихся продукций.

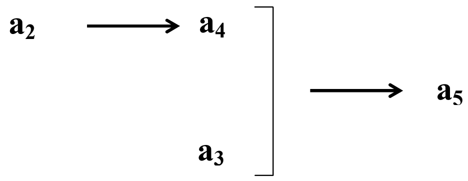


Рис. 2.3

Основная задача продукционной ЭС заключается в определении последовательности правил-продукций, позволяющей по исходным фактам получить интересующий пользователя факт.

Процесс сопоставления левых частей продукций α_i с множеством констатированных фактов A_1 порождает **цепочку вывода** или **цепочку рассуждений**. Эту цепочку можно изобразить графически (рис. 2.4).

Представленная на рисунке цепочка соответствует последовательному применению продукций

$$a_2 \to a_4$$

$$a_4 \wedge a_3 \to a_5$$

к исходному множеству $A_1 = \{a_1, a_2, a_3\}$. Возможно, новый факт a_5 и есть результат, интересующий пользователя.

В реальных продукционных ЭС начальное множество A_1 (множество известных пользователю фактов) формируется не однократно, в начале решения задачи, а последовательно, изменяясь в процессе диалога. Таким образом, пополнение A_1, происходит в реальном времени и не только в результате выполнения последовательности продукций, но и как следствие диалога с пользователем. Так, например, результат выполнения некоторого правила может означать некоторое взаимодействие с внешней средой, в том числе и вопрос к пользователю, ответ на который приводит к пополнению множества A_1.

2.2.2 Прямая и обратная цепочки вывода

Выше мы рассмотрели процесс модификации множества A_1 в соответствии с **прямой цепочкой вывода**. Основная идея заключалась в поиске новых фактов (новой информации) в направлении стрелок, разделяющих левые и правые части правил:

$$\alpha_i \to \beta_i.$$

Рассмотрим теперь более подробно на конкретном примере, как происходит вывод при прямой и обратной стратегии вывода. Мы здесь будем предполагать, что интерпретатор правил будет анализировать список правил сверху вниз и выполнять первое же правило, левая часть которого согласуется с множеством текущих фактов A_1.

Пример прямой цепочки вывода. Пусть

$$A_1 = \{a_1, a_2, a_3, a_5, a_7, a_8\},$$

список продукций:

$$P_1 : a_6 \wedge a_2 \to a_9,$$

$$P_2 : a_3 \wedge a_4 \to a_6,$$

$$P_3 : \quad a_1 \to a_4.$$

Шаг 1. Просматриваем список продукций сверху вниз, сопоставляя левые части с элементами множества A_1. В результате выполнения P_3 имеем:

$$A_1 = \{a_1, a_2, a_3, a_4, a_5, a_7, a_8\},$$

$$A_1 := A_1 \cup a_4.$$

(Добавлен элемент a_4).

Шаг 2. Снова просматриваем список правил сверху вниз. На этот раз соблюдаются условия для выполнения продукции P_2:

$$A_1 := A_1 \cup a_6 = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8\}$$

(здесь добавление a_6 осуществляется к последнему варианту множества A_1, полученного на шаге 1).

Шаг 3. Выполняется продукция P_1 и

$$A_1 := A_1 \cup a_9 = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9\}.$$

Цепочка соответствующих рассуждений построена на рис. 2.2.3.

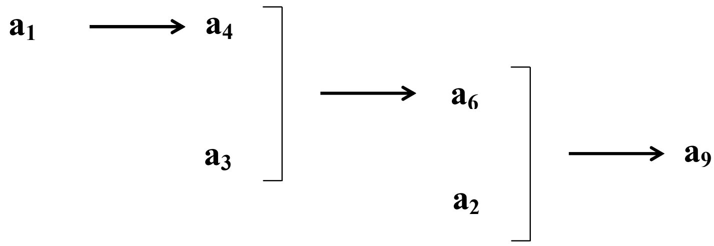


Рис. 2.4

Основной результат: было выведено, что наряду с исходными фактами (ситуациями) существуют также ситуации a_4, a_6, a_9. Например, исходные факты могут быть результатом анализов, а факты a_4, a_6, a_9 (или только a_9!) – возможные диагнозы (виды заболеваний).

Предположим теперь, что мы хотим использовать ЭС с этой же самой базой знаний, чтобы установить, существует ли ситуация a_9 (например, болен ли человек конкретной болезнью?).

На первый взгляд кажется, что мы уже установили этот факт. Да, это так в данном примере. Но мы попутно установили и другие новые факты (a_4 и a_6), которые теперь нас не интересуют. И если реальная база знаний содержит не три правила, а сотни и тысячи, то в результате прямой цепочки рассуждений будет проделана огромная лишняя (в смысле задачи установления конкретного факта) работа, связанная с построением большого числа вполне справедливых цепочек вывода и ситуаций, не имеющих отношения к искомому результату.

Поэтому в подобных случаях более выгодной может оказаться **обратная цепочка вывода**. Возвращаясь к предыдущему примеру, заметим, что основная цель состоит в доказательстве существования ситуации (факта) a_9 и мы теперь будем выполнять только те правила, которые относятся к установлению этого факта.

Пример обратной цепочки вывода. Пусть

$$A_1 = \{a_1, a_2, a_3, a_5, a_7, a_8\},$$

$$P_1 : a_6 \wedge a_2 \rightarrow a_9,$$

$$P_2 : a_3 \wedge a_4 \rightarrow a_6,$$

$$P_3 : \qquad a_1 \rightarrow a_4.$$

<u>Шаг1</u>. Системе сообщается, чтобы она подтвердила (установила) существование ситуации a_9. Вначале проверяется наличие a_9 во множестве A_1. Если его нет, то в списке правил P_i имеется правило вида

$$\alpha_i \rightarrow a_9 \cup \beta_i.$$

Система находит правило

$$P_1 : a_6 \wedge a_2 \rightarrow a_9$$

и решает, что теперь необходимо установить наличие фактов a_2 и a_6, чтобы вывести a_9. (Таких правил, вообще говоря, может быть несколько и процесс разветвляется).

<u>Шаг 2</u>. Имеем $a_2 \in A_1$, $a_6 \notin A_1$. Находим правило

$$P_2 : a_3 \wedge a_4 \rightarrow a_6$$

и задача сводится к установлению a_3, a_4.

<u>Шаг 3</u>. Имеем $a_3 \in A_1$, $a_4 \notin A_1$. Находим правило

$$P_3 : a_1 \rightarrow a_4$$

и задача сводится к установлению a_1. Но $a_1 \in A_1$ и поэтому задача установления a_9 решена.

ЭС ставит диагноз: a_9. Основная цель – факт существования a_9 – достигнута.

Как уже указывалось, ЭС является диалоговой системой и поэтому получение начальных данных и сам процесс вывода сопровождается диалогом с пользователем.

В только что рассмотренных примерах диалог присутствует (может присутствовать) на этапах установления наличия тех или иных фактов в базе знаний (во множестве A_1). Например, для обратной цепочки вывода целесообразны следующие вопросы пользователю:

Существует ли a_2? (Шаг 2).

Ответ: да ($a_2 \in A_1$).

Существует ли a_6? (Шаг 2).

Ответ: нет ($a_6 \notin A_1$).

Существует ли a_3? (Шаг 3).

Ответ: да ($a_3 \in A_1$).

Существует ли a_4? (Шаг 3).

Ответ: нет ($a_4 \notin A_1$).

Существует ли a_1?

Ответ: да ($a_1 \in A_1$).

Ясно, что если некоторые факты являются общезначимыми (знания 1-го рада), то они автоматически присутствуют в базе знаний (в A_1) и соответствующие вопросы не задаются.

2.2.3 Простая диагностирующая экспертная система.

При построении ЭС, как мы видели, возникает проблема организации диалога с пользователем. Диалог должен быть организован таким образом, чтобы задаваемые вопросы поступали к пользователю в нужное время и выглядели бы естественными для сложившейся в процессе вывода ситуации. Поэтому техника ведения диалога должна быть тщательно продумана на стадии создания ЭС. Ниже излагается один из известных (Г.С. Поспелов) подходов к созданию диагностирующих продукционных ЭС. Соответствующие интерпретации могут быть весьма разнообразными и поэтому область применимости обсуждаемых конструкций оказывается достаточно широкой.

Задано множество фактов

$$A = \{ a_{ij}\} \cup \{q_i\} = (a_1, a_2, ..., a_n),$$

состоящее из элементов двух типов. Элементы a_{ij} определяют обычные декларативные знания из конкретной предметной области. Элементы q_i определяют вид взаимодействия с внешней средой и в данном случае представляют из себя вопросы пользователю в виде альтернативного меню:

$$q_i = a_{i_1},...,a_{i_s}$$

Некоторые из q_i имеют другой смысл – смысл результирующих заключений или диагнозов, оформленных в виде соответствующих сообщений пользователю.

Продукции в данной системе имеют вид

$$a_{ij} \rightarrow q_m = \{a_{m_1},...,a_{m_k}\}.$$

Все множество фактов и продукций организованы в некоторую систему, представленную в виде графа «или». Фрагмент такого диагностирующего графа с вершинами диагнозами q_9, q_{10}, q_{11}, q_{12} (терминальными вершинами) представлен на рис. 2.6

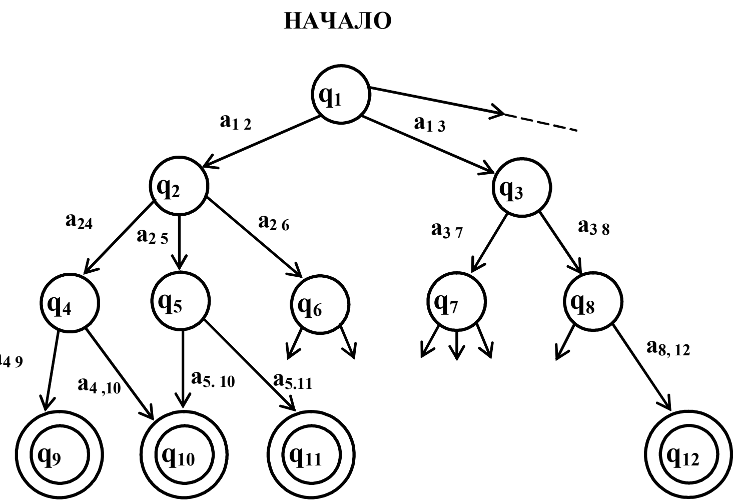


Рис. 2.5

Принцип работы такой ЭС заключается в следующем. После обращения пользователя к ЭС мы попадаем в вершину q_1, инициирующую вопрос пользователю в виде соответствующего альтернативного меню:

$$q_1 = \{a_{12}, a_{13}, ...\}.$$

Допустим, на вопрос системы: «Какой из фактов a_{12}, a_{13}, ... имеет место?» пользователь ответил: a_{12}. В результате мы заносим a_{12} в рабочее поле и попадаем в новую вершину – вопрос q_2:

$$q_2 = \{a_{24}, a_{25}, a_{26}\},$$

где ситуация повторяется. В конце концов, мы оказываемся в одной из терминальных вершин, где пользователь получает сообщение о результате, характеризующим выбранный ЭС диагноз.

При такой структуре ЭС достаточно просто может быть реализована **подсистема объяснений**, как важнейшая составная часть любой ЭС. Для этого достаточно каждой вершине q_i графа сопоставить соответствующий текст, описывающий мотивации выбора в данной вершине. А далее происходит вывод этих текстов при движении снизу (от результата) вверх в соответствии с реализованной цепочкой рассуждений. Для ответа на вопросы пользователя типа «Почему q_i, а не q_j» система, двигаясь по цепочке вывода снизу вверх, определяет место разветвления (какую-то вершину q_k) путей, ведущих к q_i и q_j. Пояснительный текст, ассоциированный с вершиной q_k и должен содержать ответ на вопрос пользователя «Почему q_9, а не q_{12}» система ответит «Потому, что a_{12}, а не a_{13}», а на вопрос «Почему q_9, а не q_{11}» получим ответ «Потому, что a_{24}, а не a_{25}» и т.п.

Указанная структуризация базы знаний ЭС в данном случае оказывается более естественной и логически оправданной, чем, например, непосредственное использование продукций, построенных в соответствии с различными путями, ведущих от начала процесса к каждой из терминальных вершин, например:

$$a_{12} \wedge a_{24} \wedge a_{49} \rightarrow q_9,$$

$$a_{12} \wedge a_{24} \wedge a_{4,10} \rightarrow q_{10},$$

$$a_{12} \wedge a_{25} \wedge a_{5,10} \rightarrow q_{10}.$$

Мы рассмотрели лишь «принципиальную схему» соответствующей ЭС. Ясно, что в действительности приходится решать много достаточно сложных вспомогательных задач. В частности, важная проблема заключается в разработке и реализации механизма пополнения и модификации знаний, представленных в виде такого диагностирующего графа.

2.2.4 Формальное представление продукционной экспертной системы.

Формальные модели продукционных ЭС играют большую роль не только для изучения конкретных ЭС, но также и при построении новых

ЭС. В частности, на основе таких формализмов могут изучаться вопросы эффективности различных механизмов вывода, вопросы непротиворечивости и полноты знаний и т.д. Рассмотрим один из возможных подходов.

Будем предполагать, что база знаний ЭС состоит из конечного набора правил:

$P = \{P_1, ..., P_m\}$,

$$P_i : a_{i_1} \wedge a_{i_2} \wedge ... \wedge a_{i_k} \to a_j \qquad (2.2.1)$$

и мысленно возможного конечного набора фактов (ситуаций)

$$A = \{a_1, ..., a_n\}.$$

Часто можно считать, что в правой части импликации (2.2.1) находится один факт а. Например, при наличии нескольких фактов, объединенных логическим «И», можно соответственно увеличить число импликаций с одной и той же левой частью. Как мы уже видели, в общем случае продукции могут иметь более сложную структуру.

Задача продукционной ЭС – определение цепочки правил, позволяющей получить (подтвердить) интересующий пользователя факт (конечно, из А). Как мы видели, процедуры построения этой цепочки могут быть различными – прямой вывод, обратный вывод. Возможен и смешанный вывод. Но в любом случае нас интересует результирующая цепочка от исходных данных к выводимому факту.

Рассмотрим ниже процедуру прямого вывода.

С учетом информации, поступающей от пользователя, каждое правило устанавливает новый факт $a_m \in A$, расширяя тем самым набор установленных фактов, находящихся в рабочем поле A_1.

Применимость любого следующего правила зависит только от состояния рабочего поля A_1 (с учетом фактов, введенных пользователем).

Множество A_1 рассматривается при этом как состояние самой ЭС, а продукции являются операторами, изменяющими это состояние.

Состояние ЭС описывается с помощью вектора состояния

$$x = (x^1, ..., x^i, ..., x^n),$$

где:

n – количество элементов базового множества фактов A;

$x_i = 1$, если $a_i \in A_1$ и

$x_i = 0$, если $a_i \in A \backslash A_1$ (факт не установлен).

Продукция

$$P_i : a_{i_1} \wedge a_{i_2} \wedge ... \wedge a_{i_k} \to a_j$$

приводит систему из состояния x в новое состояние, если она применима, т.е. если

$$x_{i_1} \cdot x_{i_2} \cdot ... \cdot x_{i_k} = 1.$$

В этом случае получаем новый вектор состояния

$$y = (y_1, \ldots, y_n), \quad P_i\, x = y,$$

где $y_k = x_k$ при $k \neq j$; $y_j = 1$. (Если уже имели $a_j \in A_1$, то $y = x$ и несмотря на применимость продукции состояние не меняется).

Если

$$x_{i_1} \cdot x_{i_2} \cdot ... \cdot x_{i_k} = 0$$

(что соответствует случаю, когда хотя бы один из фактов $a_{i_1}, a_{i_2}, ..., a_{i_k}$ не находится в рабочем поле A_1), то

$$P_i\, x = x,$$

т.е. состояние не меняется. (Продукция неприменима).

Пример. Пусть имеется список фактов

$$A_1 = \{a_1, a_2, ..., a_6\}$$

и три продукции:

$P_1 : a_2 \wedge a_3 \to a_5$,

$P_2 : a_2 \wedge a_4 \wedge a_5 \to a_6$,

$P_3 : a_6 \to a_1$.

Начальное состояние системы

$$x^0 = (011100).$$

Рассмотрим изменение состояния системы при последовательном применении указанных продукций:

$$x^1 = P_1\, x^0 = (011110),$$

$$x^2 = P_2\, x^1 = (011111),$$

$$x^3 = P_3\, x^2 = (111111).$$

Легко видеть, что, например, продукция вида $a_1 \wedge a_2 \to a_3$ неприменима в состоянии x^0 и, следовательно, не меняет этого состояния. Продукция же $a_2 \wedge a_3 \to a_4$ применима, но также не меняет состояния x^0.

Указанный формализм позволяет трактовать процесс вывода заключений в продукционной ЭС как процесс эволюции некоторой динамической системы в дискретном времени t:

$$x^{t+1} = P_t\, x^t,$$

где $\{x^1, x^2, ...\}$ – последовательность состояний ЭС, а $\{P_1, P_2, ...\}$ – последовательность примененных продукций.

Предполагается, что на каждом шаге вывода перед поиском очередной продукции в базе процедурных знаний P система имеет возможность задать вопрос пользователю (так, как это было описано выше при рассмотрении примеров) для установления дополнительных фактов из A. Стратегия ведения диалога с пользователем определяется при создании конкретных ЭС, настроенных на ту или иную предметную область.

С учетом указанной интерпретации процедуры вывода и привлекая концепцию динамической системы, можно проводить достаточно полный анализ функционирования продукционных ЭС.

Сама цель работы ЭС формализуется как оценка возможности перехода динамической системы из заданного множества начальных состояний x_0 (в частном случае это может быть одно состояние) в некоторое целевое множество

$$T = \left\{ x^i \middle| x_{i_1} \vee x_{i_2} \vee \ldots \vee x_{i_k} = 1 \right\}$$

состояний, в которых оказывается установленным хотя бы один из фактов заданного множества B:

$$B = \left(a_{i_1}, a_{i_2}, \ldots, a_{i_k} \right), B \subset A.$$

При этом могут быть привлечены важные факты из теории управляемых динамических систем с целью оптимизации траектории вывода. С помощью данного аппарата решается также важная проблема порядка выбора правил из базы P на каждом шаге вывода, а также проблемы противоречивости знаний. Однако более детальное рассмотрение этих вопросов выходит за рамки этой книги и мы отсылаем читателя к специальной литературе (см., в частности, работы Г.С. Поспелова и И.Г. Поспелова).

2.3 Представление и использование нечетких знаний

Одна из проблем, возникающая при создании ЭС, состоит в учете неточности и ненадежности любой информации. Аналогичные проблемы возникают и в других областях компьютерного моделирования, например, в методах численного анализа. Там ставится задача оценки погрешности численного результата при известных погрешностях исходных данных и неточности реализации самого алгоритма (вычислений). Иначе говоря, определяется характер «переноса» погрешностей на результат. Точно так же в ЭС возникает задача оценки степени ненадежности наших выводов и рекомендаций при неточности исходной информации и информации, поступающей в результате диалога с пользователем.

Далее мы рассмотрим один из способов рассуждения при наличии неопределенности, основанный на формуле Байеса теории вероятностей.

2.3.1 Элементы теории вероятностей.

Напомним необходимые базовые понятия элементарной теории вероятностей.

Пусть σ – некоторый комплекс условий, при осуществлении которого могут происходить события из множества

$$S = \{A, B, C, \dots\}.$$

Множество S называется системой случайных событий.

Если появление события A обязательно сопровождается появлением события B, то говорят, что A влечет B:

$$A \subset B.$$

Если $A \subset B$ и $B \subset A$, то говорят, что события A и B равносильны:

$$A = B.$$

С позиций теории вероятностей такие события неразличимы.

Событие, состоящее в наступлении обоих событий A и B, называется произведением событий A и B и обозначается как AB или $A \cap B$.

Событие, состоящее в наступлении хотя бы одного из событий A и B называется суммой событий A и B и обозначается как A + B или $A \cup B$.

Событие, состоящее в том, что A происходит, а B не происходит, называется разностью событий A и B или A – B.

Два события A и $\overline{A}$ называются противоположными, если

$$A + \overline{A} = U, A\overline{A} = V,$$

где U – достоверное событие (происходит всегда при реализации комплекса условий σ); V – невозможное событие.

Два события A и B называются несовместимыми, если их совместное появление невозможно: AB = V.

При построении математической теории вероятностей исходят из базового понятия пространства (множества) элементарных событий, отождествляя его с достоверным событием U. Природа элементов этого множества заранее не оговаривается. Различные подмножества точек из U и образуют систему событий S.

Пример. Пусть комплекс условий σ состоит в том, что внутри единичного квадрата наудачу выбирается точка. Тогда все множество точек, расположенных в указанном квадрате, образует множество U. Пусть теперь A и

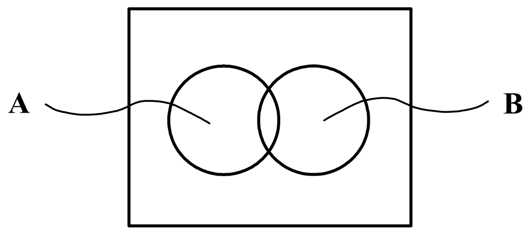


Рис. 2.6

В – события, состоящие в попадании точки внутрь левого и правого круга, соответственно (рис. 2.7)

Тогда события A, B, $\overline{A}$, $\overline{B}$, A + B, AB, U, V могут быть представлены в виде соответствующих заштрихованных подмножеств множества U (рис. 2.8).

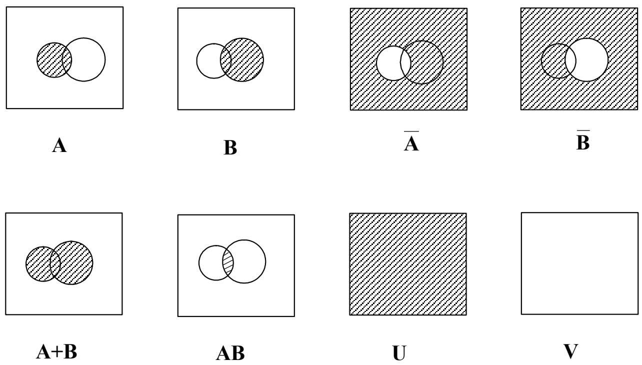


Рис. 2.7

Каждому случайному событию A ставится в соответствие неотрицательное число P (A) $\in$ [0, 1] – его вероятность.

В вышеприведенном примере вероятность события равна площади соответствующей фигуры.

Справедливы следующие основные законы теории вероятностей:

1. $P(\overline{A}) = 1 - P(A)$, где P(A) – вероятность наступления события A, а $P(\overline{A})$ – вероятность его ненаступления.

2. P(A + B) = P(A) + P(B) – P(AB).

3. P(AB) = P(A) · P(B/A) = P(B) · P(A/B).

Здесь P(B/A) означает вероятность появления события B при условии наступления события A (условная вероятность).

4. P(U) =1, P(V) = 0.

Говорят, что событие A не зависит (независимо) от события B, если

$$P(A/B) = P(A).$$

Данное равенство означает, что наступление события B не изменяет вероятности наступления события A. Оказывается, что если A не зависит от B, то и B не зависит от A. Таким образом, в случае независимости событий А и В имеем

$$P(AB) = P(A) \cdot P(B).$$

Выведем теперь основное для наших целей соотношение.

Пусть А и В – два события. Тогда в силу вышеприведенных соотношений получим

$$P(A/B) = \frac{P(AB)}{P(B)}, \quad P(B/A) = \frac{P(AB)}{P(A)}$$

или, исключая P(AB),

$$P(A/B) = \frac{P(B/A)P(A)}{P(B)}.$$

Кроме того, так как события А и $\overline{A}$ несовместны, и одно из них обязательно происходит, имеем

$$P(B) = P(B/A)\,P(A) + P(B/\overline{A})\,P(\overline{A}).$$

В результате имеем искомую **формулу (теорему) Байеса**:

$$P(A/B) = \frac{P(B/A)P(A)}{P(B/A)P(A) + P(B/\overline{A})P(\overline{A})}.$$

2.3.2 Байесовский подход.

Одним из исчислений неопределенностей в теории экспертных систем является теория вероятностей и теорема Байеса, в частности. С помощью формулы Байеса удается накапливать информацию, поступающую из различных источников, с целью подтверждения или неподтверждения определенной гипотезы (диагноза).

Пусть имеется некоторая гипотеза Н и некоторая априорная вероятность того, что гипотеза Н истинна. Эта вероятность Р(Н) либо задается в самом начале как исходное данное, либо является результатом предыдущих преобразований. Далее предполагается, что появляется некоторое свидетельство Е, относящееся к данной гипотезе и мы хотим на основе этой информации уточнить априорную вероятность истинности гипотезы Н. Согласно формуле Байеса имеем:

$$P(H/E) = \frac{P(E/H)P(H)}{P(E/H)P(H) + P(E/\overline{H})P(\overline{H})}.$$

Прокомментируем, следуя К. Нейлору, эту формулу на простом примере из области медицинской диагностики.

Пусть некоторый человек подозревается в заболевании гриппом. Следовательно, в данном случае гипотеза Н состоит в том, что он болен гриппом. Можно считать, что в медицинских учреждениях на основе статистических данных известна априорная вероятность Р(Н) того, что пациенты в данное время года и в данной местности заболевают гриппом. Пусть Е означает свидетельство высокой температуры у данного конкретного пациента. Легко видеть, что формула Байеса позволяет получить Р(Н/Е) (вероятность гриппа при наличии высокой температуры). Чтобы воспользоваться формулой Байеса для этого случая необходимо знать вероятности:

Р(Е/Н) – вероятность температуры при гриппе;

$P(E/\overline{H})$ – вероятность температуры при отсутствии гриппа.

Мы предполагаем, что обе эти вероятности нам известны. Они также получаются при обработке имеющихся статистических данных. Ясно, что все три числа Р(Н), Р(Е/Н), $P(E/\overline{H})$ могут быть получены заранее и имеют универсальный характер, не зависящий от данных по конкретному пациенту.

Теперь, замечая, что

$$P(\overline{H}) = 1 - P(H),$$

мы можем воспользоваться формулой Байеса – все числа в правой части известны.

Пусть

$$P(H) = 0.001;\ \ P(\overline{H}) = 1 - P(H) = 0.999;$$

$$P(E/H) = 1.0;$$

$$P(E/\overline{H}) = 0.01.$$

Тогда по формуле Байеса получим

$$P(H/E) \cong 0.009.$$

Таким образом, вероятность заболевания гриппом при поступлении свидетельства о высокой температуре увеличилась и составила 0.009 по сравнению с 0.001 (исходная априорная вероятность).

Принципиальная схема работы байесовской ЭС состоит в следующем.

Первоначально мы имеем априорную вероятность P(H), которая хранится в базе знаний. Но, получив свидетельство E и пересчитав эту вероятность по формуле Байеса, мы можем записать ее на место P(H). Получение очередного свидетельства приводит к новому обновлению (увеличению или уменьшению) этой вероятности. Каждый раз текущее значение этой вероятности будет считаться априорным для применения формулы Байеса. В конечном итоге, собрав все сведения, касающиеся всех гипотез (например, диагнозов болезней), ЭС приходит к окончательному заключению, выделяя наиболее вероятную гипотезу в качестве результата экспертизы.

Далее мы уточним и детализируем приведенную принципиальную схему.

2.4 Нейлоровские диагностирующие системы.

Разработанная К. Нейлором концепция построения ЭС, основана на вышеизложенной общей байесовской схеме. Основные принципы, реализованные в данной ЭС, включают:

введение верхних и нижних порогов для вероятностей гипотез;

учет неопределенностей, заключенных в реакции пользователей;

введение цен свидетельств, определяющих сценарий диалога с пользователем.

Рассмотрим эти вопросы более подробно.

2.4.1 Элементы механизма логического вывода.

Первое усложнение, которое вводится в общую схему байесовского подхода, связано с использованием верхних и нижних порогов для вероятностей отдельных гипотез. Если вероятность P(H) после учета всех свидетельств превосходит верхний порог M1(H):

$P(H) > M1(H)$,

то гипотеза H принимается как основа для возможного заключения. Если же

$$P(H) < M2(H),$$

где M2(H) – нижний порог, то гипотеза H отвергается как неправдоподобная.

Есть основания устанавливать верхние и нижние пороги M1, M2 индивидуально для каждой гипотезы, в соответствии с имеющейся в ЭС базой знаний и максимально возможных уровней вероятностей гипотез с учетом всех принципиально возможных свидетельств. Например, можно полагать

$$M1(H) = 0.9\ PMAX(H),$$

$$M2(H) = 0.5\ M1(H),$$

где PMAX(H) – максимально возможная вероятность, достижимая для данной гипотезы, при условии, что все свидетельства, имеющиеся в базе знаний и связанные с этой гипотезой, будут подтверждены пользователем в пользу гипотезы H. Величины PMAX(H) для всех H, так же как и M1(H), M2(H), очевидно, могут быть вычислены заранее и также включены в базу знаний ЭС.

Учет неопределенности, заключенной в ответах пользователя на вопросы ЭС является важным моментом в организации диалога. В идеале мы могли бы предположить, что на вопрос ЭС пользователь отвечает либо «да», либо «нет» (есть температура у пациента, нет температуры и т.д.), т.е. выполняется данное свидетельство E или не выполняется. Более реалистичной является ситуация, когда пользователь по какой-либо причине либо хочет уклониться от ответа (если, например, он слишком сложен и не соответствует квалификации пользователя в данной предметной области), либо стремится дать не слишком определенный ответ. Например, если задается вопрос о наличии повышенной температуры у пациента, то необходимо дать возможность пользователю проранжировать степень повышения температуры, например, в соответствии с 11-балльной шкалой:

– 5 соответствует НЕТ

0 соответствует НЕ ЗНАЮ

+ 5 соответствует ДА.

Присутствуют и все промежуточные целые значения шкалы от – 5 до + 5.

В результате каждое свидетельство E будет оцениваться по этой шкале на основании ответа пользователя $R \in \overline{R}$

$\overline{R} = \{-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5\}$.

В соответствии с байесовским подходом, после обработки очередного свидетельства E мы вычисляли вероятность P(H/E) и заменяли ею предыдущую вероятность P(H). Теперь мы должны предложить способ вычисления не P(H/E), а P(H/R). Это может быть выполнено следующим образом. Во-первых, случай R = 5 должен соответствовать вероятности P(H/E), вычисленной по формуле Байеса. Во-вторых, вариант R = –5 должен соответствовать величине $P(H/\overline{E})$. Последняя вероятность может быть найдена из соотношений:

$$P(H) = P(H/E)\,P(E) + P(H/\overline{E})\,P(\overline{E}),$$

$$P(\overline{E}) = 1 - P(E)$$

$$P(E) = P(E/H)\,P(H) + P(E/\overline{H})\,P(\overline{H}).$$

Здесь величины P(E/H), P(H), $P(E/\overline{H})$, P(E), P(H/E) предполагаются заданными.

В третьих, случай R = 0 (НЕ ЗНАЮ), очевидно, не должен изменять априорную вероятность P(H) и поэтому здесь имеем:

P(H/R) = P(H).

Мы получили три характерные точки на графике P(H/R) как функции R. Промежуточные значения P(H/R) предлагается восстанавливать с помощью линейной интерполяции. Таким образом, для любого $R \in \overline{R}$ мы получаем соответствующее значение P(H/R), что и требовалось. При этом максимальное значение P(H/R) будет равно P(H/E) и соответствовать случаю R = 5, как и указывалось выше.

2.4.2 Цены свидетельств – косвенная цепочка рассуждений.

Последнее усложнение, которое вносится в базовую схему байесовского подхода, связано с проблемой управления логическим выводом.

Проблема состоит в следующем. В базе знаний имеем конечное множество гипотез

$H_1, H_2, ..., H_n$

и конечное множество свидетельств (вопросов)

$E_1, E_2, \dots, E_S$.

Каждой гипотезе H_i соответствует свое подмножество ассоциированных с ней свидетельств. Согласно стратегии прямой цепочки рассуждений (мы обсуждали этот подход при рассмотрении продукционных ЭС) можно, в принципе, последовательно прорабатывать весь список возможных вопросов и затем выбирать наиболее вероятную гипотезу. При этом, однако, возникают две проблемы.

Во-первых, процедура такого перебора может потребовать больших затрат реального времени, так как мы не знаем, на какой H_i остановится в конце концов ЭС, и поэтому вопросы, ассоциированные с H_i, могут поступить пользователю, например, в конце диалога. Во-вторых, такой способ ведения диалога психологически не соответствуют желанию пользователя установить диагноз, т.е. выбрать конкретную гипотезу H. У пользователя создается впечатление, что ЭС бессистемно «прыгает» из одной области в другую, перемежая «общие» вопросы (которые на самом деле желательны лишь в начале диалога) с очень частными.

Поэтому, как мы уже знаем, часто используется обратная цепочка рассуждений: от гипотез – к подтверждающим или опровергающим свидетельствам. В этом случае мы осуществляем перебор не в множестве свидетельств E, а в множестве гипотез H. По своему поведению такая ЭС выглядит более целенаправленной, особенно при проверке конкретного диагноза. Но и здесь в общем случае (когда диагноз не известен и не ставится задача подтверждения или опровержения конкретного диагноза) неясно, в каком именно порядке следует перебирать гипотезы H_i, а также привлекать свидетельства для подтверждения тех или иных гипотез.

Далее мы рассмотрим предложенную К. Нейлором процедуру логического вывода, которая не может быть отнесена ни к прямой, ни к обратной цепочке рассуждений. Она условно может быть названа **«косвенной цепочкой рассуждений».**

Рассматриваемый подход сводится к назначению цены каждого свидетельства E_i, которая отражает важность данного свидетельства в процессе логического вывода. Далее при построении диалога каждый раз выбираются вопросы (свидетельства) с наибольшими ценами. В процессе вывода цены свидетельств все время пересчитываются в зависимости от получаемых текущих результатов. По-видимому, именно так ведут себя специалисты-эксперты, задавая те вопросы, которые представляются им наиболее важными в текущей ситуации.

В качестве цены свидетельства E предлагается использовать выражение

$$C(E) = \sum_{i=1}^{n} \left| P(H_i / E) - P(H_i / \overline{E}) \right|,$$

где:

$$P(H/E) = \frac{P(E/H)P(H)}{P(E)}, \; P(E) = P(E/H)\,P(H) + P(E/\overline{H})\,P(\overline{H})$$

соответствует формуле Байеса, а выражение для вероятности $P(H/\overline{E})$

$$P(H/\overline{E}) = \frac{(1 - P(E/H)) \cdot P(H)}{1 - P(E)}$$

получается из предыдущей формулы заменой E на $\overline{E}$ и на основании очевидных равенств

$$P(\overline{E}/H) = 1 - P(E/H),$$

$$P(\overline{E}) = 1 - P(E).$$

Таким образом, C(E) определяется как полная сумма максимально возможных изменений вероятностей по всем n гипотезам, имеющимся в базе знаний.

Далее система выбирает свидетельство E с максимальной ценой C(E) и задает соответствующий вопрос пользователю. После получения ответа (в 11-балльной шкале) пересчитываются все вероятности $P(H_i)$ – они заменяются на $P(H_i/R)$ в соответствии с вышеизложенным механизмом линейной интерполяции. После формирования нового массива $P(H_i)$ заново пересчитываются все цены свидетельств и процесс повторяется.

Отметим, что в процессе вывода все вероятности P(E/H), $P(E/\overline{H})$ остаются неизменными – меняется только массив P(H). Из формулы Байеса видно, что

$$\lim P(H/E) \; = \; 0 \;\text{ при } P(H) \to 0,$$

$$\lim P(H/\overline{E}) = \; 0 \;\text{ при } P(H) \to 0$$

(при неизменных P(E/H), $P(E/\overline{H})$). Следовательно, цены свидетельств, относящиеся к таким маловероятным гипотезам, будут автоматически падать, что естественно: желательно задавать вопросы, относящиеся к более перспективным гипотезам. И наоборот, цены свидетельств, относящиеся к более вероятным гипотезам, будут расти.

Легко видеть, что введенная формула для C(E) по существу совпадает с, может быть, более «естественной» формулой

$$\overline{C}(E)=\sum_{i=1}^{n}\left\{\left|P(H_i/E)-P(H)\right|+\left|P(H_i/\overline{E})-P(H)\right|\right\}.$$

<u>Пример.</u> Пусть P(H) = 0.001; P(E/H) = 0.9; P(E/$\overline{H}$) = 0.01. Тогда P(H/E) $\cong$ 0.0083; P(H/$\overline{E}$) $\cong$ 0.0001. В результате имеем:

$$C(E)=\left|0.0083-0.0001\right|=0.0082;$$
$$\overline{C}(E)=\left|0.0083-0.001\right|+\left|0.0001-0.001\right|=0.0073+0.0009=0.0082.$$

Возможное уточнение изложенного метода ведения диалога на основе использования цен свидетельств может быть связано с механизмом «взвешивания» свидетельств. При этом на шаге L + 1 больший вес получают свидетельства, ассоциируемые с гипотезами, имеющими отношение к запрошенному на шаге L свидетельству E_{L-1}. Соответствующие связи представлены на рис. 2.9. Согласно этому рисунку, на шаге L свидетельства E_S, …, E_{S+6} получит большие цены свидетельств, чем непосредственно получаемые по формуле для C(E). В результате шансы на выбор в качестве E_L одного из указанных свидетельств E_S, …, E_{S+6} возрастают. Указанный подход позволяет внести

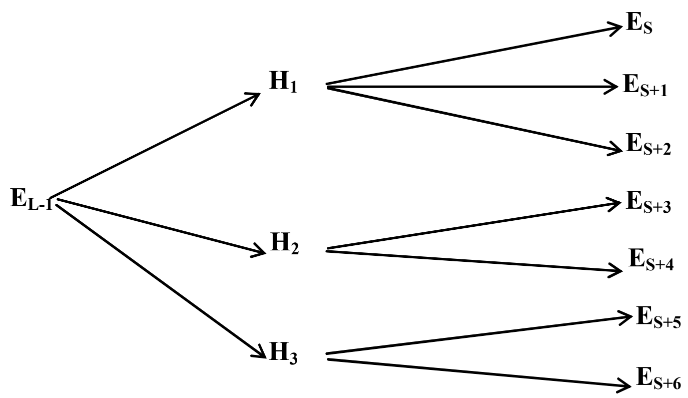


Рис. 2.9

определенную «инертность» в работу ЭС, что благоприятно отражается на психологическом состоянии пользователя.

Действительно, в этом случае ЭС перестает вести себя «нервозно» и перепрыгивать с одной группы гипотез на другую даже при малых различиях в соответствующих ценах свидетельств. Реализуется более естественный и оправданный ход диалога, что с точки зрения пользователя создает эффект более «внимательного» отношения ЭС к предмету экспертизы.

2.4.3 Правила остановки.

Мы переходим к важному вопросу определения момента окончания работы ЭС. Ясно, что для этого, вообще говоря, достаточно перебрать все свидетельства и все гипотезы. Однако окончание процесса экспертизы обычно наступает значительно раньше.

В рассматриваемой диагностирующей ЭС с каждой гипотезой Н связываются следующие пять чисел:

P(H) – текущая вероятность истинности гипотезы Н;

$P_{max}(H)$ – текущая максимально достижимая вероятность для гипотезы Н в предположении, что все оставшиеся свидетельства будут свидетельствовать в пользу Н;

$P_{min}(H)$ – текущая минимальная вероятность гипотезы Н (все оставшиеся свидетельства говорят о том, что скорее $\overline{H}$, чем Н);

M1(H), M2(H) – верхний и нижний пороги, введены в п. 15.1 как величины, пропорциональные PMAX(H).

Как уже указывалось, PMAX(H) означает максимально достижимую в данной ЭС вероятность истинности H (если все без исключения свидетельства свидетельствуют в пользу H). В самом начале процесса, очевидно, имеем:

$$P_{max}(H) = PMAX(H),$$

затем величина $P_{max}(H)$ будет, вообще говоря, меньше, чем PMAX(H). Последняя величина является константой для данной базы знаний.

С учетом указанных пяти величин возможны следующие условия остановки работы ЭС.

1. **Определение наиболее вероятной гипотезы**. Если в некоторый момент обработки списка свидетельств в процессе работы ЭС оказывается, что для какой-то гипотезы H_k выполняются условия:

$P_{min}(H_k) > P_{max}(H_i)$ для $\forall i \neq k$,

то, очевидно, гипотеза H_k оказывается наиболее вероятной и продолжать процесс экспертизы бессмысленно. Ответ пользователю – H_k.

2. **Определение правдоподобной гипотезы**. Иначе говоря, такой гипотезы H, для которой в данный текущий момент времени выполнилось условие

$$P_{min}(H) > M1(H),$$

где M1(H) – критерий верхнего порога. Данный критерий остановки является в определенном смысле основным и реализуется чаще других. Обычно пользователя интересуют все такие гипотезы и поэтому необходимо дополнительно проверить условие

$P_{max}(H) < M1(H)$

для всех остальных гипотез. Если это условие не выполнено, то процесс работы ЭС может быть продолжен для пополнения списка правдоподобных заключений.

3. **Отсутствие заключения** – констатация факта, что ЭС не может принять решения. Данная ситуация возникает, если мы имеем в какой-то момент времени

$P_{max}(H) < M2(H)$ для всех H.

(Ясно, что для корректности базы знаний мы должны потребовать выполнение условия PMAX(H) > M2(H) для всех H). Кроме этого, мы вынуждены констатировать отсутствие заключения, если при остановке согласно условиям п.1, мы имеем для наиболее вероятной гипотезы H_k

$$P(H_k) < M1(H_k).$$

Мы прекращаем обсуждение возможных условий остановки. Ясно, что для реальных ЭС все они должны быть согласованы между собой и с базой знаний ЭС. Необходимо также при создании новых баз знаний осуществлять проверку их корректности и логической непротиворечивости, чтобы процесс экспертизы, например, не закончился в самом начале.

2.4.4 Структура базы знаний и алгоритм логического вывода

База знаний рассматриваемой диагностирующей ЭС содержит записи, касающиеся знаний о конкретных диагнозах (гипотезах) и знаний о соответствующих симптомах (свидетельствах). Формат записи для описания конкретной гипотезы H (формат 1) может иметь следующий вид:

$$\text{НАЗВ. ГИП.}; P; S; (j_1;\ p_1^+; p_1^-); \ldots; (j_S;\ p_S^+; p_S^-).$$

Здесь НАЗВ. ГИП. – название гипотезы H; P = P(H) – априорная (исходная) вероятность данной гипотезы; S – число свидетельств E_i, относящихся к данной гипотезе; j_k – номер свидетельства; $p_k^+ = P(E_{j_k} / H)$ – вероятность выполнения свидетельства для данной гипотезы; $p^- = P(E_{j_k} / \overline{H})$ –

вероятность выполнения свидетельства E_{jk} при неверности данной гипотезы H.

Знания о свидетельствах могут быть представлены в следующем формате (формате 2):

№ СВИДЕТЕЛЬСТВА; НАЗВАНИЕ СВИДЕТЕЛЬСТВА; ЗАДАВАЕМЫЙ ВОПРОС.

Замечание. Одни и те же номера свидетельств j_k могут, очевидно, присутствовать в описаниях различных гипотез, но вот соответствующие вероятности p_k^+, p_k^- при этом будут, скорее всего, разными.

Пример. Рассмотрим фрагмент медицинской базы знаний.

Описание гипотезы (формат 1):

ГРИПП; 0.001; 2; (1; 1; 0.01); (2; 0.9; 0.1)

(при этом круглые скобки могут отсутствовать – мы их оставили для наглядности записи).

Описание свидетельств (формат 2):

1; ТЕМПЕРАТУРА; ЕСТЬ ЛИ У ВАС ВЫСОКАЯ ТЕМПЕРАТУРА?

2; НАСМОРК; ЕСТЬ ЛИ У ВАС НАСМОРК?

Дадим расшифровку приведенных записей. Априорная вероятность того, что некто болен гриппом, равна 0.001. В данной базе знаний с гриппом связаны два (2) симптома. Первый симптом – высокая температура. Вероятность высокой температуре здесь положена равной 1. Вероятность высокой температуры при отсутствии гриппа равна 0.01.

Второй симптом – насморк. Вероятность насморка при гриппе равна 0.9. Вероятность насморка при отсутствии гриппа равна 0.1.

На симптом № 1 (температура) могут быть ссылки и из записей для других гипотез (болезней), но вероятности p^+, p^- в тройках (1, p^+, p^-) будут уже другими.

Подытожим все наше рассмотрение нейлоровских байесовских ЭС в виде следующего простого алгоритма:

<u>Шаг 1.</u> Сформировать массив $P(H_i)$ априорных вероятностей для всех гипотез. Для этого просмотреть базу знаний (формат 1) и извлечь P для каждой из гипотез (это второй элемент записи в формате 1).

<u>Шаг 2.</u> Сформировать массив цен свидетельств $C(E_j)$:

$$C(E_j) = \sum_{i=1}^{n} \left| \frac{P(E_j / H_i) P(H_i)}{P(E_j / H_i) P(H_i) + P(E_j / \overline{H}_i) P(\overline{H}_i)} - \right.$$
$$\left. - \frac{(1 - P(E_j / H_i) P(H_i)}{1 - P(E_j / H_i) P(H_i) - P(E_j / \overline{H}_i) P(\overline{H}_i)} \right|,$$

где j – номер свидетельства, определяет выбор троек (j, p^+, p^-) из базы знаний (формат 1). При этом

$$p^+ = P(E_j / H_i); \; p^- = P(E_j / \overline{H}_i).$$

Очевидно, в сумму для $C(E_j)$ входят только те слагаемые, которые соответствуют гипотезам H_i, содержащим в своем описании (формат 1) тройки с номером j.

<u>Шаг 3.</u> Определить свидетельство E_m с максимальной ценой:

$$E_m : m = \arg\max_j C(E_j).$$

(Здесь же может быть реализована процедура определения E_m с учетом результата дополнительного взвешивания цен свидетельств, как это было указано в п.15.2).

Шаг 4. Задать пользователю вопрос, хранящийся в базе знаний (формат 2) для найденного на шаге 3 свидетельства E_m . Ответ пользователя R_m должен быть дан по шкале от –5 до +5.

Шаг 5. Пересчитать массив $P(H_i)$ в соответствии с полученным на шаге 4 ответом пользователя:

$$P(H_i) := P(H_i/R_m)$$

(см. п. 2.4.1).

Шаг 6. С учетом полученных на шаге 5 новых значений для элементов массива $P(H_i)$ пересчитать элементы массива цен свидетельств $C(E_j)$.

Шаг 7. Вычислить для каждой гипотезы H_i значения $P_{max}(H_i)$, $P_{min}(H_i)$ (см. п. 15.3).

Шаг 8. Определить число PM:

$$PM = \max_i P_{min}(H_i)$$

– это наибольший из возможных достижимых минимумов вероятностей для всех гипотез.

Шаг 9. Проверить, существует ли такой номер k, для которого $P_{max}(H_k) > PM$.

Если «ДА», то перейти к шагу 3. Если «НЕТ», то выбрать гипотезу (гипотезы) H_m:

$$m = \arg\max_i P(H_i)$$

(наиболее вероятный результат).

Шаг 10. Выдать в качестве результата гипотезу (гипотезы) H_m и вызвать подпрограмму объяснений, которая представляет пользователю протокол с описанием всех выводов, проделанных ЭС. Протокол снабжается необходимыми комментариями.

Как уже указывалось ранее, могут быть реализованы и более сложные критерии окончания работы ЭС.

2.4.5 Пример базы знаний.

Рассмотрим теперь ставший уже классическим пример базы знаний для ЭС, используемой для ремонта автомобилей (К. Нейлор). Пример носит чисто иллюстративный характер и не претендует на использование в реальных задачах.

Гипотезы H_i:

СЕВШИЙ АККУМУЛЯТОР; 0.1; 5; (1; 0; 0.99); (2; 0.7; 0.05); (4; 0.2; 0.5); (5; 0; 0.99); (6; 1; 0.01)

НЕТ БЕНЗИНА; 0.05; 2; (2; 1; 0.01); (6; 0.9; 0.02)

ОТСЫРЕЛ РАСПРЕДЕЛИТЕЛЬ ЗАЖИГАНИЯ; 0.01; 3; (3; 0.9; 0.1); (4; 0.25; 0.5); (6; 0.9; 0.02)

ЗАМАСЛЕНЫ СВЕЧИ; 0.01; 2; (4; 0.01; 0.5); (6; 0.9; 0.02)

Свидетельства (симптомы) E_i:

1; ФАРЫ ГОРЯТ; ГОРЯТ ЛИ ФАРЫ?

2; УКАЗАТЕЛЬ БЕНЗИНА НА НУЛЕ; ЕСТЬ ЛИ БЕНЗИН?

3; АВТОМАШИНА ОТСЫРЕЛА; НЕ СТОЯЛА ЛИ АВТОМАШИНА ДОЛГО ПОД ДОЖДЕМ?

4; АВТОМАШИНА НЕДАВНО ПРОШЛА ТЕХОБСЛУЖИВАНИЕ; ПРОХОДИЛА ЛИ НЕДАВНО АВТОМАШИНА ТЕХОБСЛУЖИВАНИЕ?

5; СТАРТЕР КРУТИТСЯ; КРУТИТСЯ ЛИ СТАРТЕР?

6; АВТОМАШИНА НЕ ЗАВОДИТСЯ; АВТОМАШИНА НЕ ЗАВОДИТСЯ?

По данной базе знаний можно сформировать массив $C_1(E_i)$ исходных цен свидетельств. Они приведены в табл. 15.5.1.

Таблица 15.5.1.

№	Свидетельство	$C_1(E)$	$C_2(E)$
1	Фары горят	0.9174	0.9991
2	Указатель бензина на нуле	1.4151	1.2135
3	Автомашина отсырела	0.0822	0.7554
4	Автомашина недавно прошла техобслуживание	0.1376	0.8153
5	Стартер крутится	0.9174	0.9991
6	Автомашина не заводится	2.3807	0.0000

Согласно вычисленным исходным ценам свидетельств $C_1(E_i)$ первым будет всегда задаваться вопрос, связанный со свидетельством E_6 (у него максимальная цена 2.3807):

АВТОМАШИНА НЕ ЗАВОДИТСЯ?

При ответе пользователя «ДА» (R = 5) мы можем пересчитать массив $P(H_i)$ и вычислить новые цены свидетельств $C_2(E_i)$. Они приведены в табл. 15.5.1. С учетом новых цен свидетельств будет задан следующий вопрос, связанный со свидетельством E_2 (у него максимальная цена 1.2135). При этом сообщение о том, что автомашина все-таки не заводится привело к существенному увеличению цен и остальных свидетельств, что выглядит весьма естественно.

3 ПРИМЕРЫ СИСТЕМ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ.

3.1 Quick choice - система многокритериального выбора вариантов.

3.1.1 Область применения системы

Система Quick Choice (QCH) предназначена для решения многокритериальных задач выбора вариантов из заданного конечного множества

$X = (x_1, \dots , x_n)$.

Предполагается, что каждый из вариантов x_i оценивается по m частным критериям F_k. Частные критерии могут иметь как числовые так и порядковые шкалы. Примером числового критерия в задаче выбора автомобиля из заданного множества (например, при покупке подержанного автомобиля) может служить пробег автомобиля, выраженный в километрах – чем он меньше, тем лучше. Цвет автомобиля естественно оценивать в порядковой шкале, например, вида:

черный-белый-коричневый- ... –желтый.

Здесь предполагается, что степень предпочтительности убывает слева направо.

Таким образом в данной системе приняты следующие предположения об исходной задаче:

- множество исходных вариантов конечно, и ЛПР (пользователь – лицо принимающее решение) может перечислить элементы этого множества

ЛПР может определить цель по каждому критерию, а также задать некоторые параметры критериев, описанные ниже

ЛПР может ранжировать критерии по важности и указывать равноценные критерии.

Кроме того предполагается, что ЛПР может принимать или отвергать предлагаемые ему альтернативы, а также указывать не устраивающие его по некоторым критериям оценки отвергаемой альтернативы.

Система QCH поддерживает непрерывный диалог с пользователем, в процессе которого у ЛПР запрашивается вся необходимая информация о задаче и создается отчет о ней. Для запуска этого диалога в меню «Запуск» выбирается пункт «Диалог».

3.1.2 Необходимые исходные данные

Для работы с системой необходимо ввести следующие исходные данные об альтернативах и критериях:

количество критериев – число m

тип каждого критерия (порядковый или числовой)

цель по каждому критерию (максимум или минимум)

все принимаемые значения (для перечислимых критериев)

набор ординальной (порядковой) информации вида «критерий P важнее критерия Q»

набор ординальной информации вида «критерий P равноценен критерию Q»

количество альтернатив (вариантов), среди которых производится выбор – число n

весь набор альтернатив

значения всех частных критериев для каждой альтернативы

минимальные требования к выбираемой альтернативе по каждому из частных критериев.

3.1.3 Типы критериев

Для каждого критерия необходимо задать некоторые свойства. Эти свойства описываются далее. Свойство «цель» описывает намерения ЛПР по данному критерию: минимум или максимум. Если целью по данному

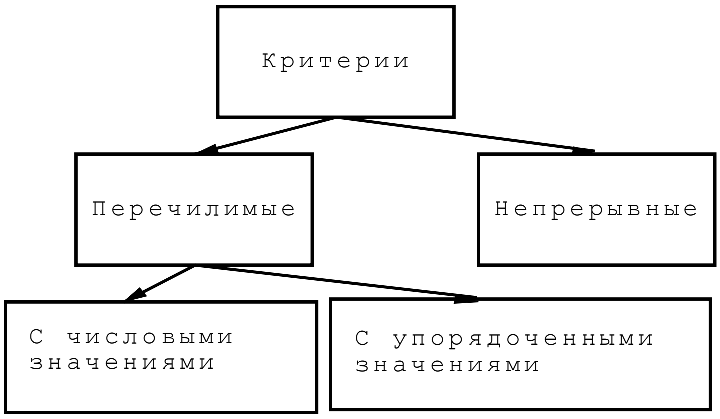


Рис. 3.1

критерию является максимум, то система предпочитает альтернативы с большим значением данного критерия, если же минимум, то наоборот.

Структура задания типа критерия изображена на рис. 3.1. Для непрерывного критерия ЛПР должен задать максимальное и минимальное значение. Непрерывный критерий принимает значения на заданном промежутке. Перечислимый критерий принимает значения из фиксированного набора. Значения перечислимых критериев могут задаваться численно, например:

Отлично	5
Хорошо	4
Удовлетворительно	3
Плохо	2

В этом случае пользователь задает текстовые значения и в системе они заменяются на численные.

Однако, система не требует от ЛПР задания численных значений, если задать тип критерия как «Перечислимый с упорядоченными значениями». В этом случае имеется в виду порядковая шкала и достаточно задать лишь набор значений данного критерия, а также упорядочить их в порядке предпочтения.

3.1.4 Функции, реализованные в системе.

Описываемая система позволяет выполнять следующие функции:

- задание информации о задаче в диалоге с пользователем
- построение множества оптимальных вариантов в соответствии с методом t - упорядочения
- реализация метода ограничений, позволяющего пользователю выбрать одну альтернативу из t - оптимального множества
- импорт из базы данных
- импорт данных из текстового файла
- проверка корректности ординальной информации
- создание отчета.

3.1.5 Инсталляция системы.

Требования к аппаратуре и окружению. Для работы системы QCH необходима следующая минимальная конфигурация: 486 процессор и 8Мб ОЗУ. После инсталляции система занимает на диске 8 – 10Мб, в зависимости от количества устанавливаемых компонентов. Для работы системы достаточно наличие на компьютере Windows 95/98 или Windows NT 4.0.

Установка системы. Для запуска процесса инсталляции необходимо запустить файл **setup.exe** и следовать указаниям системы. Система спросит имя папки, в которую будут скопированы файлы, необходимые для ее работы. Также будет создана папка в меню «Программы». Для запуска системы необходимо выбрать и запустить программу **QChoice.exe**. В состав системы также входят компоненты:

QCHOICE.HLP – Файл справки системы.

В подкаталоге \DEMOS располагаются примеры задач для данной системы.

Auto.qct – пример решения задачи выбора автомобиля из заданного множества вариантов.

Flat.txt – пример создания текстового файла, для последующего импорта в систему.

Flat.DB – пример таблицы БД Paradox для демонстрации импорта данных в систему из стандартной БД.

Flat.qct – пример решения задачи о выборе квартиры из множества вариантов.

В подкаталоге **\Document** будет располагаться данное «Руководство пользователю», а также краткая инструкция пользователю системы.

На рис. 3.2 показана общая структура данной системы. Исходная информация о задаче задается в диалоге с пользователем. После завершения диалога система имеет информацию об альтернативах, критериях, и ординальную информацию о критериях. В системе также предусмотрено получение данных об альтернативах из текстового файла или баз данных Dbase и Paradox. После этого производится нормализация исходных данных и запуск метода t-упорядочения.

Выбор результирующей альтернативы производится на основе описанного в этой книге и достаточно хорошо известного метода ограничений. Для более эффективного использования метода ограничений производится предварительное ранжирование альтернатив, по результатам которого определяется порядок предложения альтернатив на рассмотрение пользователя. В качестве метода ранжирования альтернатив применяется метод аддитивной свертки. При этом учитывается ординальнальная информация о критериях, полученная от пользователя.

При работе метода ограничений помимо текущего варианта, пользователю также предлагается «идеальная» альтернатива (с наилучшими возможными оценками по всем критериям) и установленные ограничения. Эти данные предоставляются пользователю, как в численном виде, так и в виде диаграммы. При таком представлении пользователь может наилучшим образом оценить достоинства и недостатки текущей оцениваемой альтернативы.

После получения решения система создает отчет о задаче.

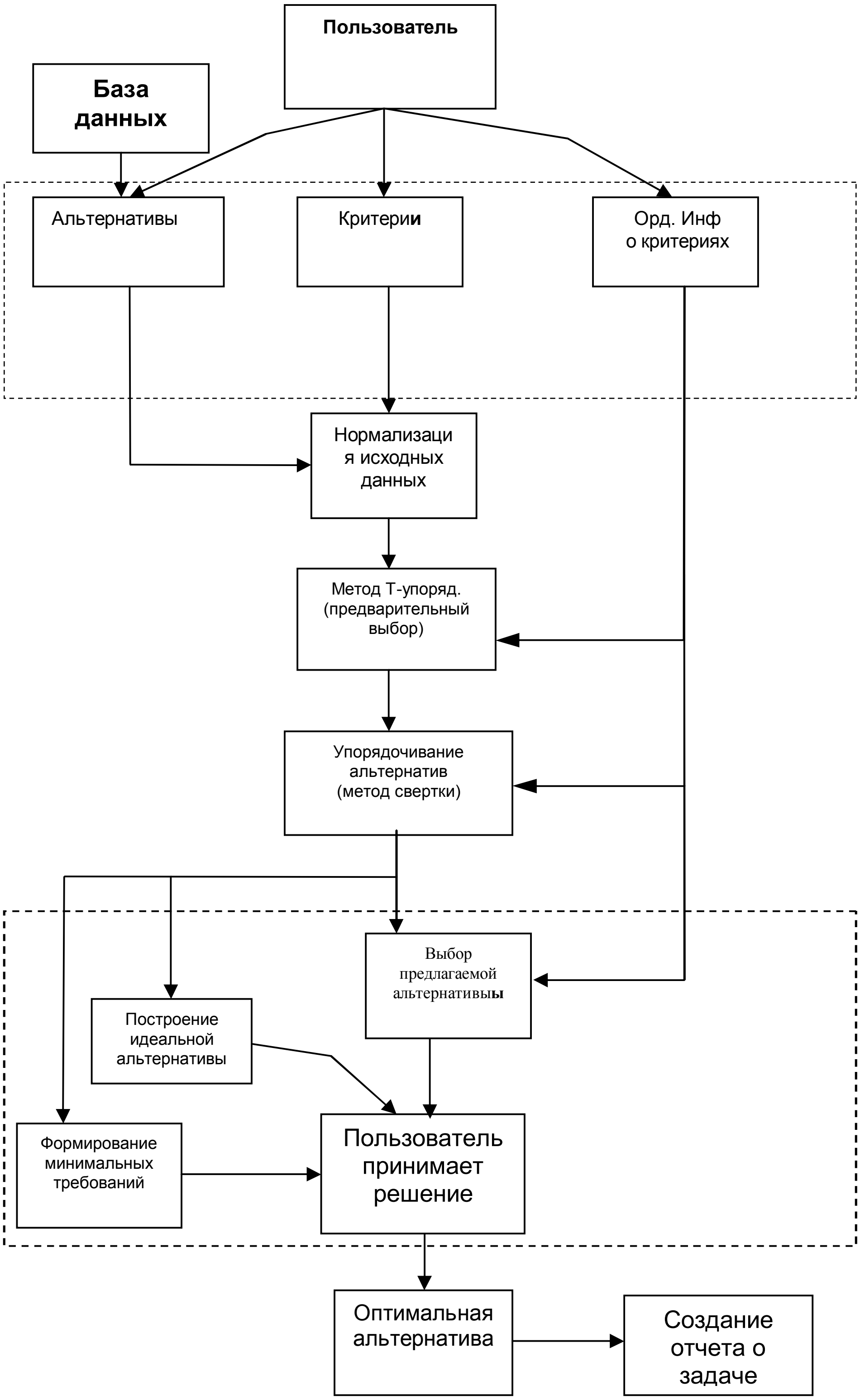


Рис.3.12 Общая структура систем

3.1.6 Первый запуск системы.

При запуске система предложит вам выбрать один из следующих вариантов работы:

- *создать новую задачу в диалоге с системой* – вызывается диалог, в результате которого система получит от пользователя необходимые данные об интересующей его задаче и поможет произвести правильный выбор
- *создать новую задачу и задать параметры вручную* – система выходит из диалогового режима и предлагает пользователю задавать данные о задаче и принимать решение на основе главного меню и диалоговых окон
- *загрузить ранее созданную задачу* – система предложит пользователю загрузить задачу из файла
- *загрузить ранее созданную и задать собственные предпочтения* – система предложит пользователю загрузить задачу из файла, а затем будет запущен диалог, в результате которого пользователь сможет изменить заданные параметры задачи и, задав свои предпочтения, принять решение о выборе какой-либо альтернативы
- *получить данные из базы данных или текстового файла* – этот пункт следует выбрать, если исходные данные располагаются в текстовом файле или базе данных. После выбора этого пункта будет реализован соответствующий диалог.

3.1.7 Получение данных из текстового файла или базы данных.

Система QCH позволяет получать исходные данные об альтернативах из стандартной базы данных Paradox или Dbase, а также из текстового файла. Для этого в системе реализованы соответствующие диалоги, которые описаны ниже.

Получение данных из текстового файла.

После запуска импорта данных из текстового файла, необходимо задать требуемый файл. Данные об альтернативах в текстовом файле должны располагаться в виде таблицы. Строка таблицы должна соответствовать одной альтернативе, а столбец содержит оценки альтернатив по соответствующим критериям.

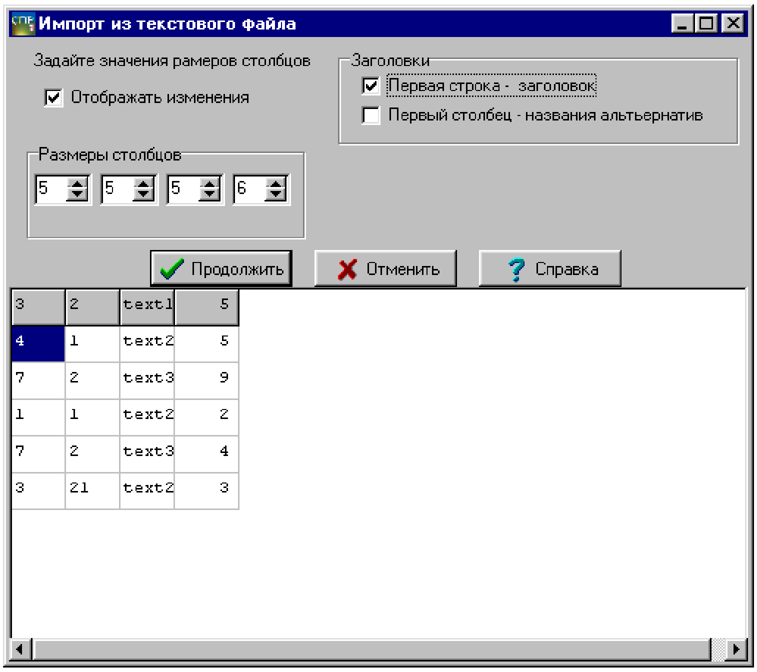


Рис.3.3 Импорт из текстового файла

Описание каждой альтернативы должно начинаться с новой строки, а значения различных критериев должны разделяться пробелами. Пример такого файла приведен ниже:

```
3  2  text1  5
4  1  text2  5
7  2  text3  9
1  1  text2  2
7  2  text3  4
```

В первой строке такого файла могут быть расположены названия критериев, а в первом столбце названия альтернатив. Однако это не обязательно, при отсутствии названий, система назовет альтернативы и критерии стандартным образом.

После выбора исходного файла, система попросит задать количество столбцов в исходном файле и после этого будет запущен диалог, изображенный на рис. 3.3.

Исходный файл будет представлен на экране в виде таблицы. От пользователя требуется задать размеры столбцов таким образом, чтобы все значения, относящиеся к различным критериям, были бы расположены в различных столбцах изображенной таблицы, и соответственно относящиеся к одному критерию, располагались бы в одном столбце. Изменение размера столбцов будет непосредственно отображаться на экране.

Если первый столбец файла содержит названия альтернатив, то необходимо отметить флаг "первый столбец - названия альтернатив". Если первая строка файла содержит названия критериев, то необходимо отметить флаг «первая строка – заголовок» При этом система автоматически определит : какие критерии считать перечислимыми, а какие непрерывными. После нажатия кнопки «Продолжить», система спросит пользователя об его предпочтениях, попросит упорядочить полученные значения перечислимых критериев и запустит диалог принятия решения.

Получение данных из базы данных

После запуска импорта из базы данных, необходимо задать файл необходимой таблицы базы данных. После этого данная таблица будет показана на экране (см. рис. 3.3). Затем необходимо нажать кнопку «Дальше». Система предложит пользователю список всех полей данной базы (рис. 3.4). Из этого списка требуется выбрать поле, в соответствии с которым будут именоваться альтернативы. После этого необходимо выбрать поля, которые будут использоваться в качестве

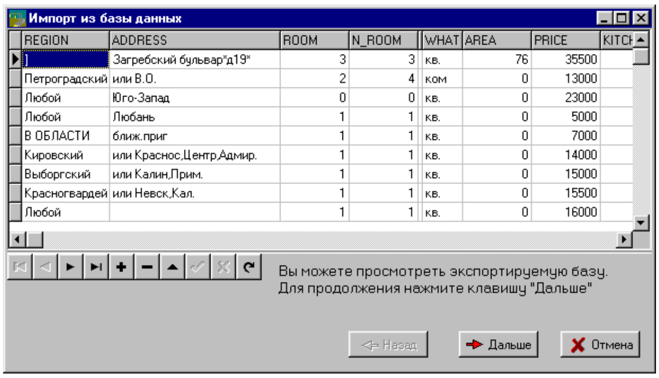


Рис.3.14 Импорт из БД

критериев в данной системе. Для этого необходимо перенести нужные поля из правого списка в левый. При этом система автоматически определит, какие критерии считать перечислимыми, а какие непрерывными. После нажатия кнопки «Продолжить», система спросит пользователя об его предпочтениях и запустит диалог принятия решения.

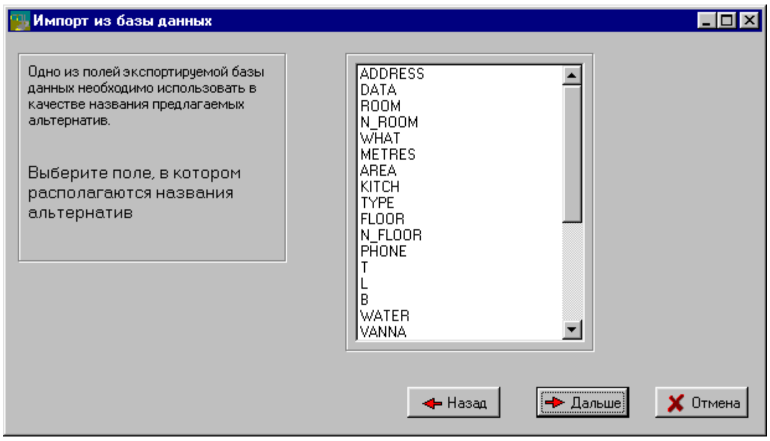


Рис.3.15 Выбор поля

После этого будет запущен диалог, аналогичный «Созданию новой задачи в диалоге с пользователем».

3.1.8 Принятие решений в диалоге с пользователем.

В данной системе реализован диалог с пользователем, позволяющий пользователю описать интересующую его задачу и получить решение, отвечая на вопросы, задаваемые системой. Такой подход может расширить круг пользователей системы, потому что от них не требуется специальных знаний в области принятия решений. Процедура принятия решения состоит

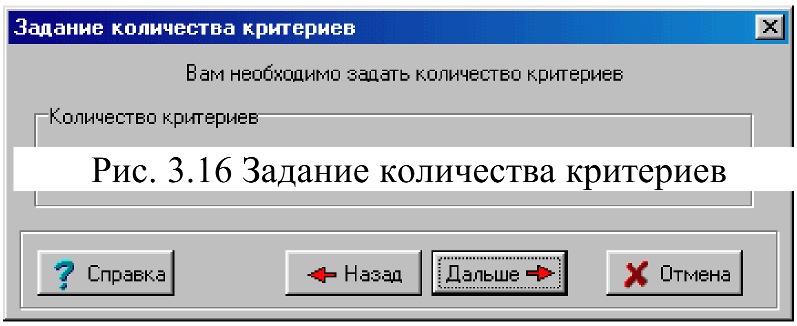


Рис. 3.16 Задание количества критериев

из пяти этапов.

задание критериев

задание альтернатив

задание дополнительной информации о критериях

предварительный выбор альтернатив (метод t-упорядочения)

метод ограничений.

На первых двух этапах пользователь непосредственно задает данные о задаче, описывая множества критериев и альтернатив. Далее в диалоге с пользователем определяются предпочтения ЛПР. Исходя из заданных предпочтений, система отбирает наиболее предпочтительные варианты. На пятом этапе система предлагает пользователю выбрать одну альтернативу в соответствии с методом ограничений.

Диалог состоит из последовательности диалоговых окон, в каждом из которых пользователь задает ту или иную информацию о задаче. После задания этой информации следует нажать кнопку «Дальше». При этом будет вызвана очередная экранная форма. Система реализована таким образом, что пользователь всегда может вернуться к предыдущим этапам, чтобы изменить ранее введенную информацию о задаче. Эта последовательность завершается запуском метода ограничений.

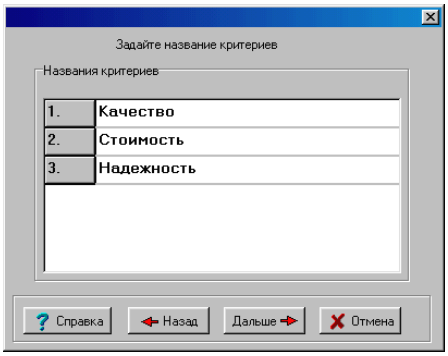


Рис. 3.17 Задание названий критериев

Задание критериев в диалоге с пользователем.

В системе QCH критерии можно задавать двумя способами: либо задавать критерии в диалоге, либо при помощи команд главного меню. Рассмотрим задание критериев в диалоге.

Вначале система запрашивает название критериев и их количество, после этого для каждого критерия система задает пользователю следующие вопросы:

«Какова ваша цель по данному критерию?»,

«Можете ли вы перечислить все значения данного критерия или хотите задать максимальное и минимальные значения?»,

«Можете ли вы численно оценить каждое принимаемое критерием значение?»

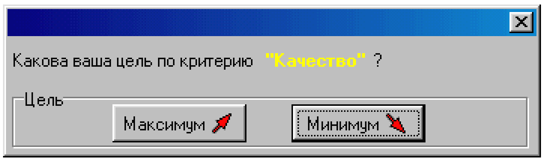


Рис. 3.18

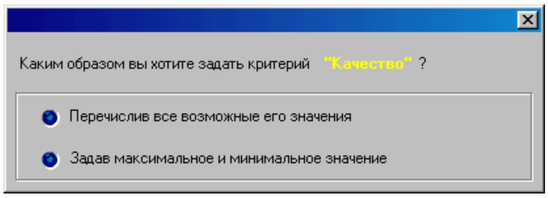


Рис. 3.19

Примеры вопросов о критериях

В процессе ответов на эти вопросы система формирует параметры критериев. В заключении выводится окно «свойства критериев», где пользователь должен либо задать максимальное и минимально значение, либо перечислить значения, принимаемые данным критерием.

Задание альтернатив в диалоге с системой.

После задания критериев требуется задать альтернативы. Альтернативы в системе задаются перечислением. Для каждой альтернативы необходимо задать все значения по каждому из заданных критериев. Для непрерывных критериев, задается соответствующее число, а для перечислимых критериев система предложит выбрать значение из множества возможных.

Сначала система запрашивает названия альтернатив и их количество, затем для каждой альтернативы предлагается задать ее значения в диалоге, показанном на рис. 3.10.

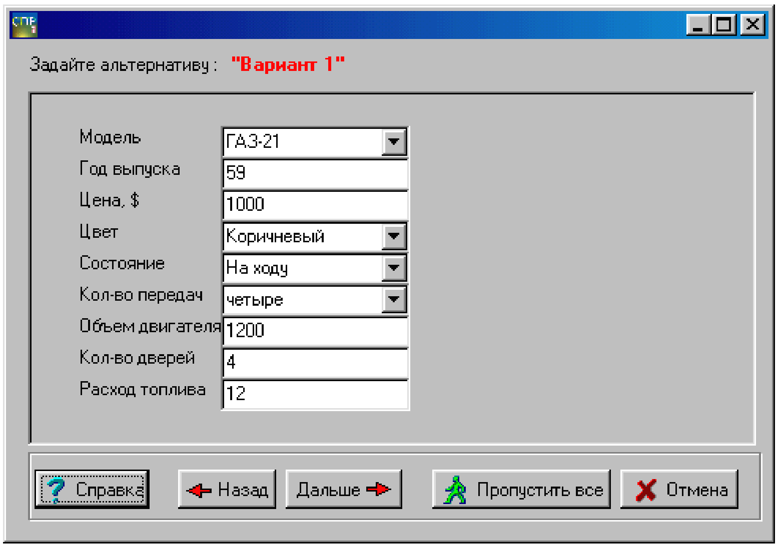


Рис 3.110 Задание альтернативы в диалоге

При помощи кнопки «Пропустить все» можно отказаться от последовательного задания альтернатив и задать их, например, непосредственно в окне альтернативы после окончания диалога.

Задание дополнительной информации о критериях в диалоге с пользователем.

Дополнительная информация поможет значительно сократить исходное множество вариантов. Особенно эффективно задание дополнительной информации для непрерывных критериев.

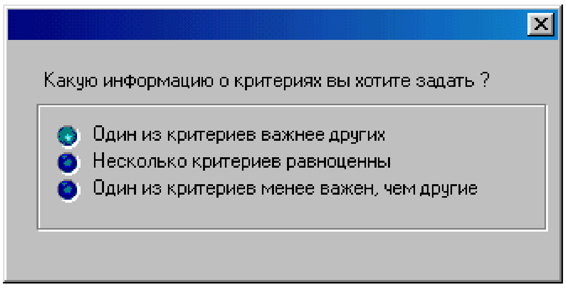


Рис. 3.111 Задание дополнительной информации

Система предложит пользователю выбрать один из типов доп. информации.

Один критерий важнее других.

Несколько критериев равноценны.

Один из критериев менее важен.

Для первого и третьего варианта будет предложено выбрать критерий, а для второго отметить равноценные критерии.

В случае важности одного из критериев, необходимо будет ответить на следующий вопрос «По сравнению с какими критериями данный критерий будет более важен?». Возможны три варианта:

Важнее всех критериев

Важнее какого-нибудь одного

Важнее нескольких критериев

Эту информацию система спросит у пользователя.

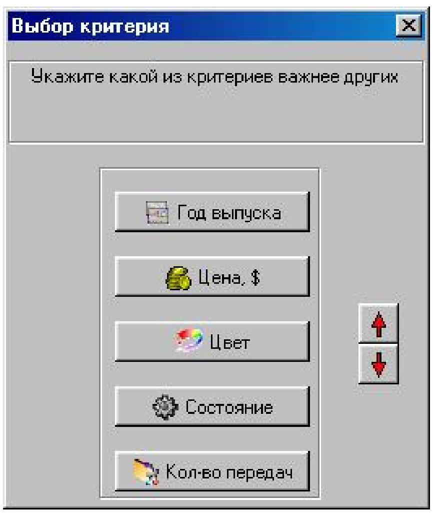


Рис. 3.12 Выбор критерия

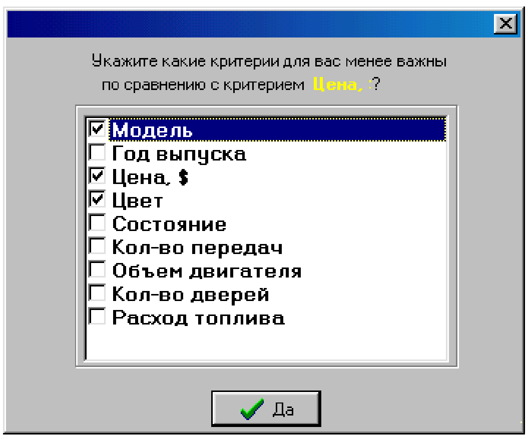


Рис. 3.13 Задание группы равноценных критериев

3.1.9 Метод ограничений.

Для выбора одной альтернативы из оптимального множества в данной системе реализован метод ограничений. Суть этого метода заключается в том, что ЛПР предоставляется один из вариантов, который он может принять или отвергнуть. В случае если ЛПР не устраивает данный вариант, ему предоставляется возможность задать минимальные требования по какому-нибудь критерию, с целью преложить ему более приемлемую альтернативу. Если же в исходном множестве не имеется альтернатив, удовлетворяющих требованиям ЛПР, то система попросит ослабить требования по какому-нибудь критерию. Внешний вид системы при запуске метода ограничений изображен на рис.3.14.

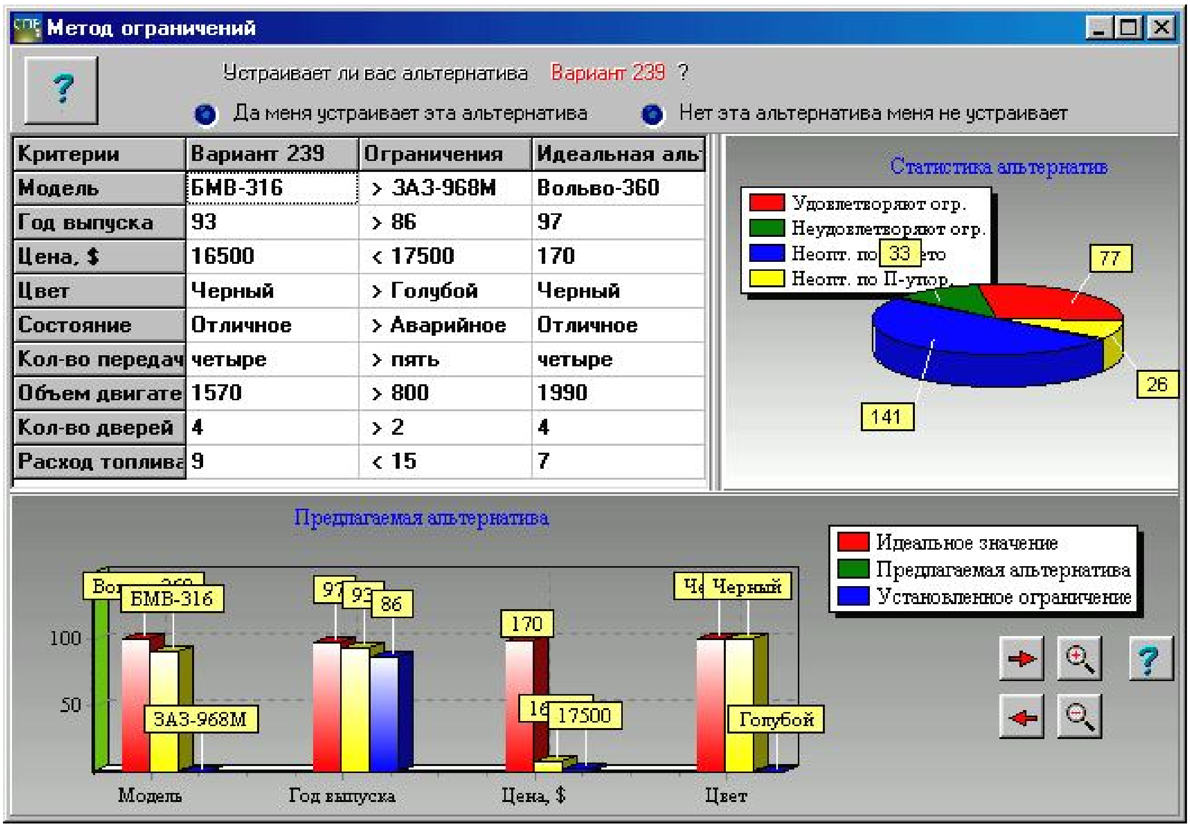


Рис. 3.114 Метод ограничений

В таблице, помимо значений предлагаемой альтернативы, также располагаются минимальные требования по каждому из критериев и идеальная альтернатива. Этой альтернативы не существует среди заданного набора. Идеальная альтернатива содержит наилучшие значения по каждому из критериев в отдельности. С ее помощью нее пользователь может видеть: какое наилучшее значение он, в принципе, может получить по каждому критерию. При этом значения по другим критериям, скорее всего, будут хуже.

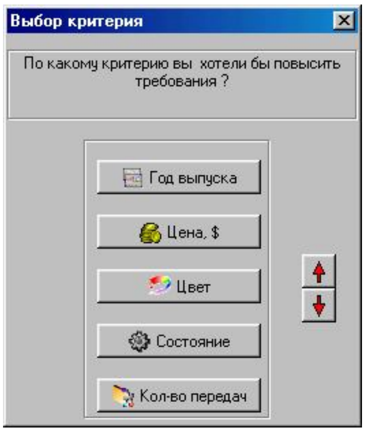


Рис. 3.115 Выбор критерия

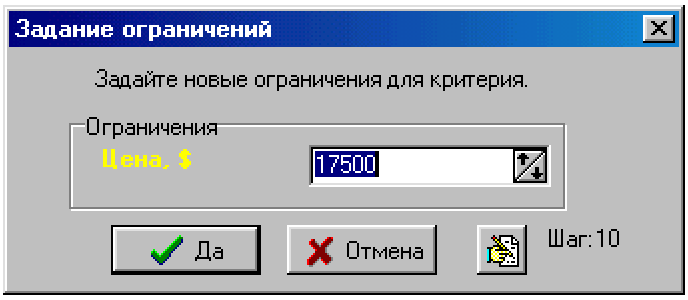


Рис. 3.116 Задание ограничений

Эта процедура будет продолжаться до тех пор, пока ЛПР не согласится с предлагаемой альтернативой. После этого будет запущен процесс создания отчета о задаче.

Диаграмма «Статистика альтернатив»

На данной диаграмме можно увидеть распределение исходного множества альтернатив по следующим группам:

Удовлетворяют огр. – это альтернативы, удовлетворяющие установленным минимальным требованиям и прошедшие предварительный отбор. Пользователь в методе ограничений осуществляет свой выбор именно из этого множества альтернатив. При значительном количестве таких альтернатив, не следует соглашаться с предлагаемой альтернативой, так как имеется возможность увеличить свои требования по критериям.

Не удовлетворяют огр. – это альтернативы, не удовлетворяющие ограничениям, но прошедшие предварительный отбор.

Не оптимальные по Парето – эти альтернативы не входят во множество Парето данной задачи. Поэтому они не могут быть выбраны ни при каких условиях, так как всегда в исходном множестве альтернатив найдется заведомо лучшая.

Не опт. по t-упорядочиванию – это альтернативы, которые были отсеяны с использованием дополнительной информации о критериях. При отсутствии данной информации это множество всегда будет пустым.

Диаграмма «Предлагаемая альтернатива»

Для большей наглядности и оценки положительных и отрицательных качеств предлагаемой альтернативы в методе ограничений выводится данная диаграмма. По каждому из критериев приводятся значения предлагаемой альтернативы, установленные ограничения и идеальное (максимально допустимое) значение.

По умолчанию на одной странице выводится информация о 4-х критериях. Для просмотра информации по остальным критериям предназначены кнопки : 

Для изменения числа критериев, одновременно выводимых на экран, предназначены кнопки : 

Задание параметров метода ограничений

Метод ограничений является основным методом в данной системе, но его эффективность зависит от вспомогательных методов, осуществляющих предварительный отбор альтернатив и выбор предлагаемого варианта в методе ограничений. Параметры этих методов можно задать, выбрав пункт главного меню «Опции» – «Настройка метода ограничений». При этом пользователю будет предложен диалог, изображенный на рис. 3.17.

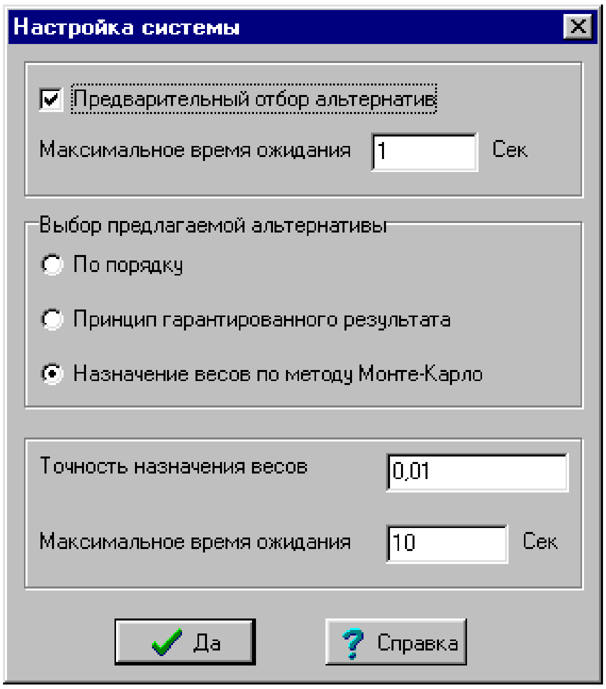


Рис. 3.117 Настройка метода ограничений

Настройка предварительного выбора

Предварительный отбор альтернатив – определяет: следует ли системе осуществлять предварительный выбор. Если данная опция будет отключена, то система будет отбирать альтернативы только по принципу Парето не используя ординальную информацию о критериях. Данную опцию разумно выключать при большом (>10) количестве критериев, когда предварительный отбор занимает достаточно много времени.

Максимальное время ожидания – Задает максимально допустимое время прохождения предварительного отбора (по умолчанию 10 секунд), по прохождении этого времени пользователю будет предложено либо отказаться от предварительного отбора, либо изменить эти параметры, либо изменить ординальную информацию о критериях (подробнее см пункт «Предварительный выбор альтернатив»).

При большом количестве критериев и большом количестве информации о них, предварительный выбор альтернатив может занимать достаточно много времени. Поэтому имеет смысл ограничить максимальное время прохождения предварительного отбора альтернатив.

Настройка выбора предлагаемой альтернативы

Как уже было отмечено выше, эффективность метода ограничений зависит от разумного выбора предлагаемой альтернативы. Однако в методе ограничений окончательный выбор все равно остается за пользователем, поэтому важность метода выбора предлагаемой альтернативы отходит на второй план. В данной системе для выбора предлагаемой альтернативы используется метод аддитивной свертки. Для каждой альтернативы вычисляется обобщенное значение (ОЗ).

$$\text{ОЗ} = \sum_{i=1}^{N} W_i * Alt_i$$

Пользователю предлагается альтернатива с максимальным значением ОЗ, из числа альтернатив, удовлетворяющим ограничениям.

Веса критериев (Wi) вычисляются с использованием одного из следующих методов:

Назначение весов по методу Монте-Карло.

Назначение весов по принципу гарантированного результата.

Рассмотрим задаваемые параметры выбора предлагаемой альтернативы.

По порядку – отсутствует дополнительный выбор предлагаемой альтернативы. Альтернативы предлагаются пользователю в произвольном порядке.

Принцип гарантированного результата – для определения весов критериев используется принцип гарантированного результата. Веса назначаются наихудшим образом. Этот метод можно использовать при «осторожности» пользователя или при большом количестве критериев, когда метод Монте-Карло не эффективен. Для данного метода необходимо задать параметр «минимальный вес», определяющий минимальное значение весового коэффициента.

Назначение весов по методу Монте-Карло – Назначение весов по методу Монте-Карло. Для данного метода задается точность вычисления весов и максимальное время ожидания.

При большом количестве критериев назначение весов может производиться неоправданно долго. Параметр «минимальное время ожидания» ограничивает этот процесс. Ограничение максимального времени работы данного метода связано с тем, что окончательный выбор все равно осуществляется в методе ограничений, а данный метод является вспомогательным.

3.1.10 Главное окно.

После запуска системы, на экране появляется основное окно программы.

Данная система является многооконной и состоит из набора окон, каждое из которых предназначено для ввода или вывода какой-либо информации. Названия окон перечислены ниже:

Задание альтернатив – в этом окне располагаются альтернативы и их характеристики по заданным критериям.

Критерии – выводятся критерии и их свойства.

Ординальная информация о критериях – выводится ординальная информация о критериях.

Нормализованные исходные данные – Выводятся нормализованные исходные данные.

Предварительные результаты – в этом окне выводятся множество оптимальных альтернатив после предварительного отбора.

Информация – в этом окне располагается статистика, а также поле для ввода пользователем краткого комментария к решаемой задаче

Каждое из этих окон можно свернуть или развернуть на все окно системы. Также имеется возможность изменения шрифтов каждого из окон системы. Расположение окон и установленные шрифты располагаются в файлах с расширением *.DS .

Следует отметить, что большинство функций главного меню могут быть реализованы вызовом Popup-menu. Эти меню нажатием правой кнопки мыши на соответствующем окне.

3.1.11 Главное меню программы.

Рассмотрим функции, которые можно реализовать при помощи главного меню.

Файл – Работа с файлами задач.

Новый – Создание новой задачи.

При выборе этого пункта создается новая задача. Параметры, принимаемые при создании новой задачи, можно задать при помощи меню «Опции» -«Новая задача».

Открыть – Открытие существующей задачи.

При выборе этого пункта меню программа вызывает стандартный диалог, в котором запрашивается имя загружаемой задачи.

Сохранить – сохранение задачи.

Данная команда позволяет сохранить задачу с установленным именем, показанным в верхней части главного окна. В случае если данный файл существует, программа спросит подтверждение на перезапись данного файла.

Сохранить как – Сохранение задачи с новым именем.

Данная команда позволяет сохранить задачу, задав при этом новое имя файла.

Импорт данных – Запуск импорта из базы данных или текстового файла.

Выход – при помощи этого пункта меню осуществляется выход из программы, при выходе программа предлагает сохранить имеющуюся задачу.

Запуск –

Выбрать – Запуск метода предварительного выбора.

При этом осуществляется запуск алгоритма принятия решения и осуществляется построение множества оптимальных вариантов. После правильного завершения работы алгоритма, в окно «Информация» выводится статистика работы алгоритма, а окно «результаты» становится активным. В окне результатов располагаются выбранные варианты и их значения по каждому из критериев и ОП.

Диалог – запуск диалога.

Метод ограничений – запуск метода ограничений.

Критерии – позволяет задавать информацию о критериях, располагаемую в окне, «Критерии».

Добавить – Добавление нового критерия.

Удалить – Удаление критерия.

Свойства – Задание свойств критерия.

Количество – Задание количества критериев

Альтернативы – Данные команды позволяют модифицировать набор альтернатив.

Удалить – Удаление альтернативы.

Добавить – Добавление критерия.

Эта команда меню позволяет добавлять новые альтернативы. После появления новой альтернативы необходимо задать ее имя и значения по каждому из критериев.

Редактировать – редактирование названия альтернативы.

Эта команда позволяет редактировать имя альтернативы, которое располагается в таблице альтернатив и таблице результатов.

Автоформат – Включение/Выключение автоматического задания ширины столбцов в окне «альтернативы».

Доп. Информация – При помощи этих команд осуществляется задание ординальной

информации о критериях.

Добавить – Добавление новой строки ординальной информации о критериях.

Удаление – Удаление строки ординальной информации.

Очистить – Удаление всей дополнительной информации.

Опции – задание параметров системы.

Задание параметров метода ограничений – Определение параметров метода предварительного отбора альтернатив и метода выбора предлагаемой альтернативы.

Новая задача – Задание параметров новой задачи.

Сохранить – Сохранение расположения окон и установленных шрифтов в файле **default.ds,** эти параметры будут загружаться автоматически при запуске.

Сохранить в файл – Сохранение расположения окон и установленных шрифтов в файле, имя которого задается при выборе этой команды.

Загрузить – Загрузка расположения окон и установленных шрифтов из файла **default.ds.**

Загрузить из файла – Загрузка расположения окон и установленных шрифтов из произвольного файла.

Запускать диалог при старте – Если этот пункт включен, то при старте система будет автоматически загружать диалог с пользователем.

Изменить панель управления – Вызов диалога, в котором задаются элементы панели управления.

Дополнительно – Некоторые дополнительные функции.

Случайно заполнить таблицу – Этот пункт позволяет заполнить альтернативы случайными значениями критериев. Эта функция используется для тестирования метода, а также может применяться для его более подробного его изучения.

Окно – задание расположения окон.

Рядом – Располагает все не свернутые окна, таким образом, что окна не накладываются друг на друга.

Каскадом – Располагает все не свернутые окна таким образом, что окна накладываются друг на друга.

Упорядочить все - Упорядочивает все свернутые окна.

Свернуть все – Сворачивает все окна.

Альтернативы – Делает окно «альтернативы» активным.

Критерии – Делает окно «критерии» активным.

Нормализованные данные - Делает окно «нормализованные данные» активным.

Информация – Делает окно «Информация» активным.

Результаты – Делает окно «результаты» активным.

Доп. информация – Делает окно «Дополнительная информация » активным.

Отчет –

Создание отчета – При помощи этой команды можно создать отчет о текущей задаче.

Справка – Информация о системе и вызов справки.

3.1.12 Рабочие окна программы.

Рассмотрим более подробно каждое из диалоговых окон, назначение которых описано в пункте. 2.3

Задание альтернатив

В этом окне располагаются альтернативы и их значения по заданным критериям. По строкам располагаются введенные альтернативы, по столбцам заданные критерии. Например, на рис. 3.1.17, альтернатива «Вариант 1» характеризуется следующими значениями по заданным критериям.

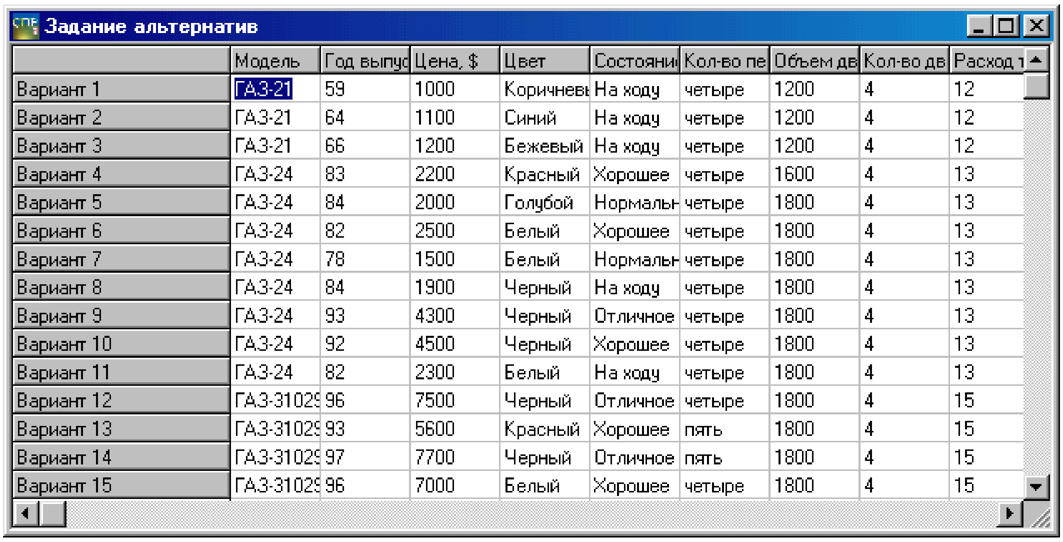
```
Модель             -    ГАЗ 21
Год Выпуска        -    59
Цена $             -    1000
Цвет               -     Коричневый
Состояние          -     На ходу
К-во передач       -    Четыре
Объем двигателя    -    1200
К-во дверей        -    4
Расход топлива     -    10
```

Аналогично задаются, все альтернативы.

При нажатии правой кнопки мыши в окне альтернатив, на экране возникает следующее подменю.

Удалить – удаление альтернативы.

Добавить – добавление альтернативы



	Модель	Год выпус	Цена, $	Цвет	Состояни	Кол-во пе	Объем дв	Кол-во дв	Расход т
Вариант 1	ГАЗ-21	59	1000	Коричневь	На ходу	четыре	1200	4	12
Вариант 2	ГАЗ-21	64	1100	Синий	На ходу	четыре	1200	4	12
Вариант 3	ГАЗ-21	66	1200	Бежевый	На ходу	четыре	1200	4	12
Вариант 4	ГАЗ-24	83	2200	Красный	Хорошее	четыре	1600	4	13
Вариант 5	ГАЗ-24	84	2000	Голубой	Нормальн	четыре	1800	4	13
Вариант 6	ГАЗ-24	82	2500	Белый	Хорошее	четыре	1800	4	13
Вариант 7	ГАЗ-24	78	1500	Белый	Нормальн	четыре	1800	4	13
Вариант 8	ГАЗ-24	84	1900	Черный	На ходу	четыре	1800	4	13
Вариант 9	ГАЗ-24	93	4300	Черный	Отличное	четыре	1800	4	13
Вариант 10	ГАЗ-24	92	4500	Черный	Хорошее	четыре	1800	4	13
Вариант 11	ГАЗ-24	82	2300	Белый	На ходу	четыре	1800	4	13
Вариант 12	ГАЗ-3102	96	7500	Черный	Отличное	четыре	1800	4	15
Вариант 13	ГАЗ-3102	93	5600	Красный	Хорошее	пять	1800	4	15
Вариант 14	ГАЗ-3102	97	7700	Черный	Отличное	пять	1800	4	15
Вариант 15	ГАЗ-3102	96	7000	Белый	Хорошее	четыре	1800	4	15

Рис. 3.118 Окно альтернативы

Редактировать – редактирование названия альтернативы.

Эта команда позволяет редактировать имя альтернативы, которое располагается в таблице альтернатив и таблице результатов.

Шрифт – задание шрифта для данного окна.

Автоформат – Вкл/выкл автоматическую установку ширины столбцов в данном окне.

Если курсор находится на значении перечислимого критерия, то система предлагает выбрать одно из его заданных значений (см. рис. 3.19).

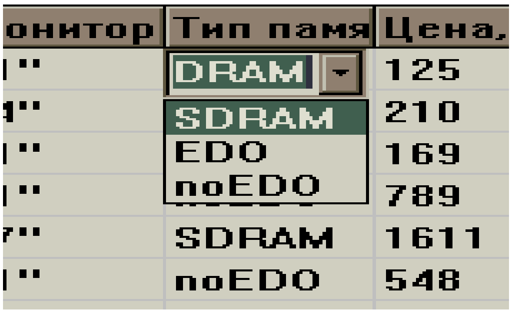


Рис. 3.119 Задание значения перечислимого критерия

Удаление альтернативы

Удалить альтернативу из текущего набора, можно при помощи подменю окна альтернативы или при помощи команды главного меню «Альтернатива» – «Удалить».

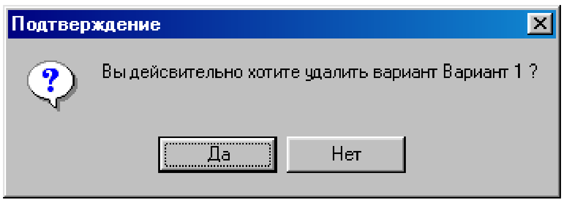


Рис. 3.120 Подтверждение удаления критерия

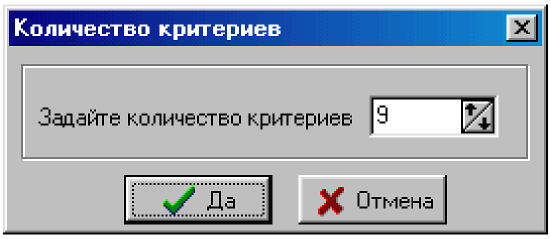


Рис. 3.121 Задание количества альтернатив

Перед вызовом одной из этих команд необходимо указать удаляемую альтернативу при помощи мыши в окне альтернатив. После этого программа спросит подтверждение, пример которого показан на рис. 3.20. И если пользователь согласится, то альтернатива будет удалена.

Добавление альтернатив

Добавить новые альтернативы к исходному набору можно несколькими способами. Если пользователю необходимо добавить одну альтернативу, то можно воспользоваться командой «Добавить» подменю окна альтернативы или командой главного меню «Альтернативы» – «Добавить». При необходимости добавить несколько альтернатив в систему, можно воспользоваться командой главного меню «Альтернативы» – «Количество». При этом будет вызван диалог, изображенный на рис. 3.21. Следует отметить, что при помощи этой команды можно только добавлять альтернативы.

Изменение названия альтернативы

Изменение названия альтернативы можно произвести при помощи команды «Редактировать имя варианта», которая расположена в подменю окна альтернативы. Это подменю можно вызвать, нажав правую кнопку мыши на окне альтернативы. После этого будет выдан диалог, позволяющий задать новое имя варианта. Имя варианта, используемого по умолчанию, можно задать при помощи команды главного меню «Опции» – «Новая задача».

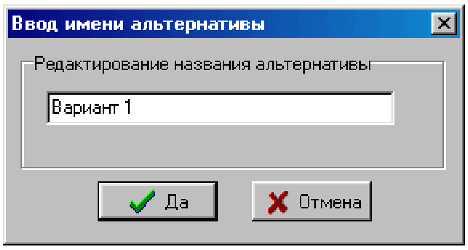


Рис. 3.22 Изменение имени альтернативы

Окно «Критерии»

Это окно, расположенное на рис. 3.23, предназначено для задания критериев и их свойств. В данном окне критерии располагаются в виде таблицы, в которой

Имя критерия	Мини...	Макс...	Тип критерия	Цель	Зада...	Нормализ...
Модель	1.000	32.000	Переч	Макс	Отнош.	Линейная
Год выпуска	0.000	100.0...	Непр.	Макс	Числ.	Линейная
Цена, $	0.000	1800...	Непр.	Мин	Числ.	Линейная
Цвет	1.000	11.000	Переч	Макс	Отнош.	Линейная
Состояние	1.000	7.000	Переч	Макс	Отнош.	Линейная
Кол-во передач	1.000	3.000	Переч	Макс	Отнош.	Линейная
Объем двигателя	750.0...	2500....	Непр.	Макс	Числ.	Линейная
Кол-во дверей	2.000	6.000	Непр.	Макс	Числ.	Линейная
Расход топлива	5.000	15.000	Непр.	Мин	Числ.	Линейная

Рис. 3.1 Окно критерии

показывается название критерия и его свойства: тип, цель, минимальное и максимальное значение. При нажатии правой кнопки мыши вызывается подменю, содержащее следующие команды:

Добавить – Добавление нового критерия.

Удалить – Удаление критерия.

Свойства – Задание свойств критерия.

Шрифт – Задание шрифта для данного окна.

Добавление нового критерия

Добавить новые критерии к исходному набору можно несколькими способами. Если пользователю необходимо добавить один критерий, то можно воспользоваться командой «Добавить» подменю окна критерии или командой главного меню «Критерии» – «Добавить». При этом пользователю будет предложено задать свойства созданного критерия.

При необходимости добавить сразу несколько критериев, следует воспользоваться командой главного меню «критерии» – «количество». При помощи этой команды вызывается диалог, позволяющий задать количество критериев, показанный на рис. 3.24. При этом нельзя задать количество критериев меньше, чем их было в системе, при такой попытке будет выдано сообщение изображенное на рис. 3.25.

При использовании данной команды свойства добавляемых критериев будут заданы по умолчанию. Эти свойства можно задать при помощи команды «Опции» – «Новая задача».

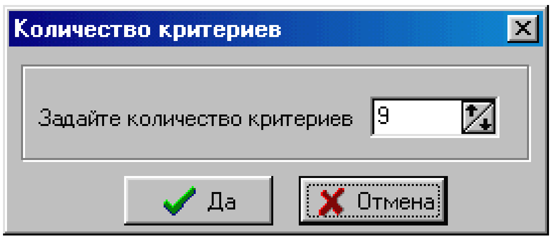


Рис. 3.24 Задание количества критериев

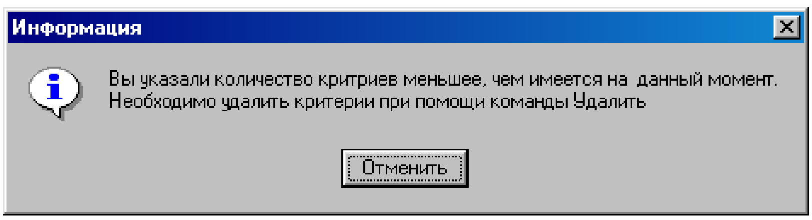


Рис. 3.125 При помощи команды "Количество" нельзя уменьшать количество критериев.

Удаление критерия

Для удаления критерия следует воспользоваться командой главного меню «критерий» – «удалить» или командой подменю окна «Критерии» «удалить». Перед этим необходимо выбрать удаляемый критерий в окне «Критерии». Система не может работать с количеством критериев меньше двух и при попытке удалить одного из двух критериев будет выдано сообщение об ошибке.

Изменение свойств критерия

Для изменения свойств критерия следует воспользоваться командой главного меню «критерий» – «свойства» или командой подменю окна «Критерии» «свойства». Перед этим необходимо выбрать удаляемый критерий в окне «Критерии». Если критерий не будет выбран, то будет выдан диалог, показанный на рис. 3.26, в котором пользователь должен выбрать изменяемый критерий.

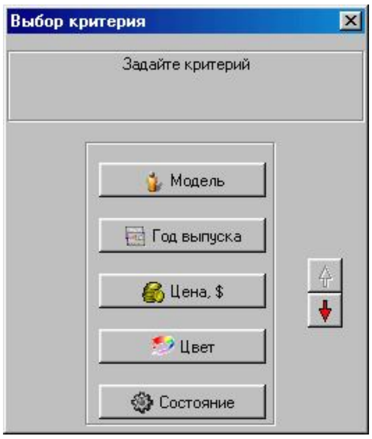


Рис. 3.126 Выбор критерия

Диалог изменения свойств критерия различен для перечислимых и непрерывных критериев. Эти диалоги показаны на рис. 3.27 и 3.28. Пользователь должен изменить необходимые свойства критерия и нажать клавишу «Да». Различные типы критериев описаны выше, в разделе «исходные данные». Для перечислимого критерия необходимо задать значения, которые он может принимать. Это производится при помощи кнопок «добавить» и «удалить».

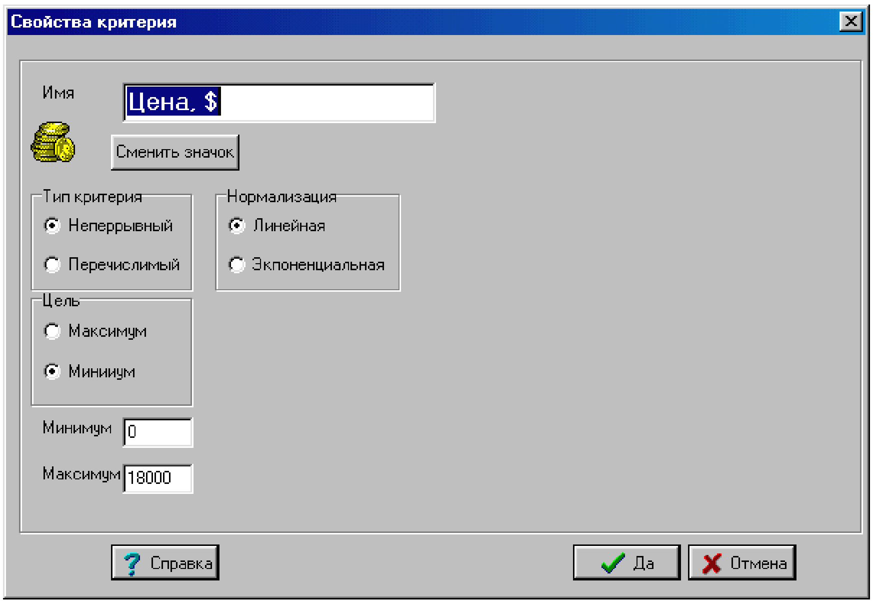


Рис. 3.127 Изменение свойств непрерывного критерия

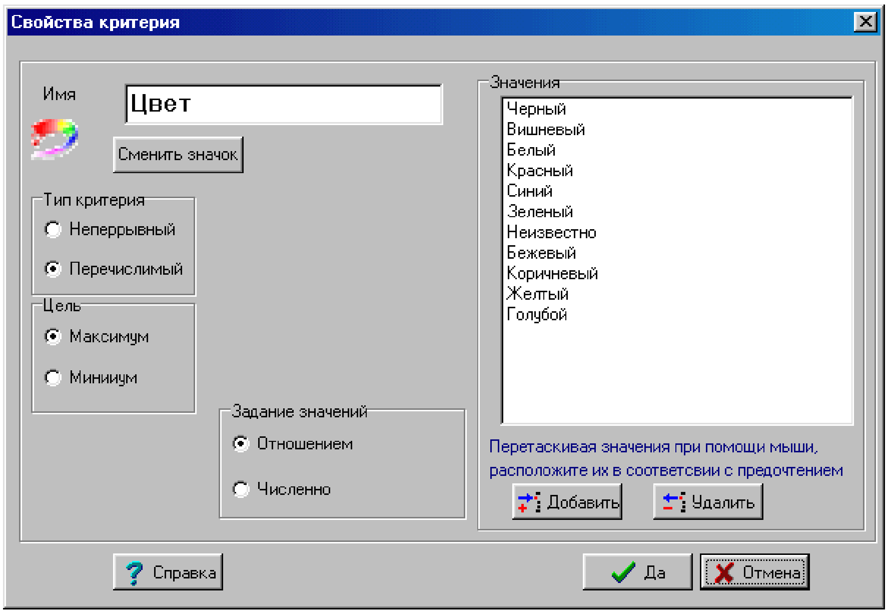


Рис. 3.128 Изменение свойств перечислимого критерия

Если значения перечислимого критерия задаются численно, то необходимо ввести эти численные оценки. Если же значения задаются перечислением, то значения необходимо упорядочить их по предпочтению.

Это делается при помощи мыши. Необходимо указать мышью на критерий и, не отпуская кнопку мыши, переместить его на нужное место. Изменить введенные значения можно, нажав правую кнопку мыши на списке и выбрав команду “Свойства”.

Для изменения значка редактируемого критерия необходимо нажать кнопку «Сменить значок» и выбрать новый значок. Диалог задания значка показан на рис. 3.29.

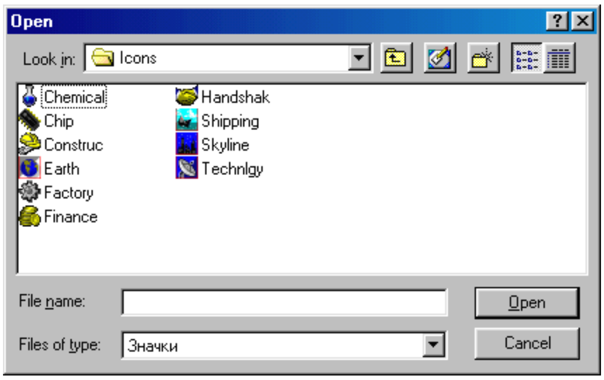


Рис. 3.129 Задание значка

Ординальная информация о критериях

В этом окне выводится ординальная информация о критериях. Внешний вид данного окна показан на рис. 3.30. Там располагаются строки ординальной информации типа: «Критерий 1 важнее критерия 2» и «Критерий 1 равноценен Критерию 2». Соотношения важности и равноценности соответственно обозначаются символами «>» и **«=».** ЛПР, предоставляя информацию о том, что один критерий важнее другого, предполагает, что после нормализации исходных данных, любое увеличение более важного критерия, при уменьшении менее важного критерия на такую же величину, улучшит оценку. ЛПР, предоставляя

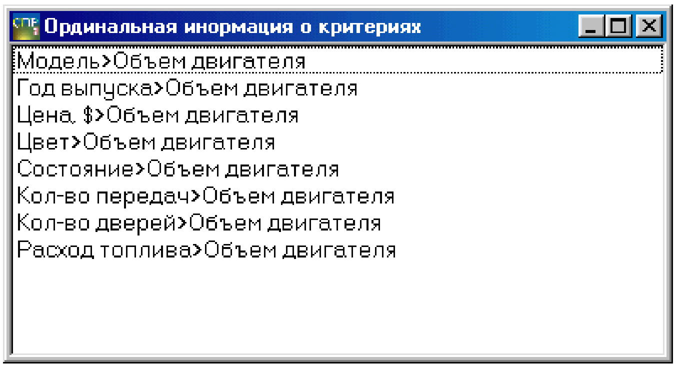


Рис. 3.130 Ординальная информация о критериях

информацию о равноценности критериев, предполагает, что после нормализации исходных данных для него безразлично любое уменьшение одного из равноценных критериев при увеличении второго на такую же величину.

При нажатии правой кнопки мыши в окне ординальной информации появляется меню, содержащее следующие команды:

Добавить – Добавление новой строки ординальной информации о критериях.

Удаление – Удаление строки ординальной информации.

Шрифт – Задание шрифта для данного окна.

Добавление ординальной информации о критериях

Для добавления ординальной информации о критериях можно воспользоваться командой главного меню «Доп. Информация» – «Добавить» или командой подменю окна «Доп. Информация» «Добавить». При этом на экране появится диалог, изображенный на рис. 3.31.

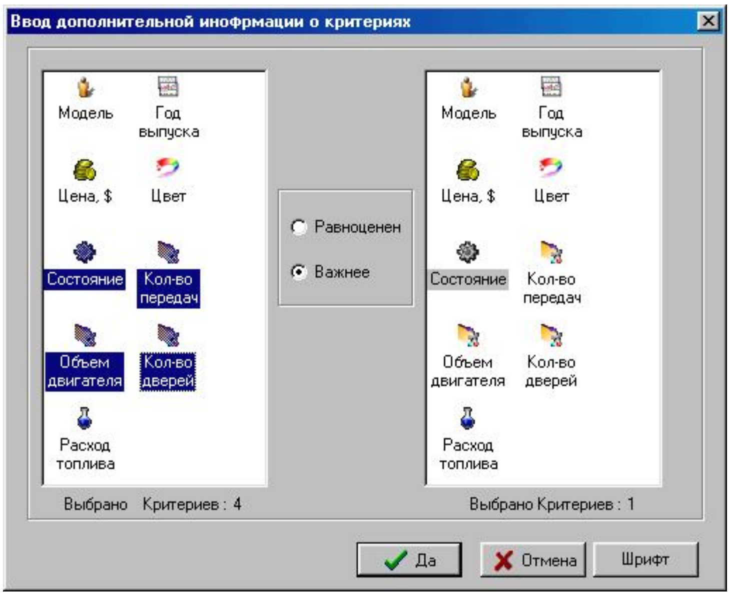


Рис. 3.31 Добавление ординальной информации о критериях

Данная система позволяет задавать сразу несколько строк ординальной информации. Для этого надо выбрать первую группу критериев, выбрать вторую группу и задать отношение. Для выделения нескольких критериев следует указывать на выбираемый критерий, удерживая при этом клавишу **Ctrl**. После этого необходимо нажать кнопку «Да». Добавление строк в ординальную информацию будет проводиться по следующему принципу: для каждого из выбранных критериев левой части будет добавляться о важности или равноценности с каждым из критериев, выбранных в правой части. Например, если выбрать в левой части критерии 1,2,3 ,а в правой критерии 4,5, то будут добавлены следующие строки ординальной информации.

Критерий 1 > Критерий 4
Критерий 2 > Критерий 4
Критерий 3 > Критерий 4
Критерий 1 > Критерий 5
Критерий 2 > Критерий 5
Критерий 3 > Критерий 5

Таким образом, можно быстро ввести большое количество ординальной информации. Если среди добавляемых строк ординальной информации будут противоречивая, то система выдаст сообщение изображенное на рис. 3.32.

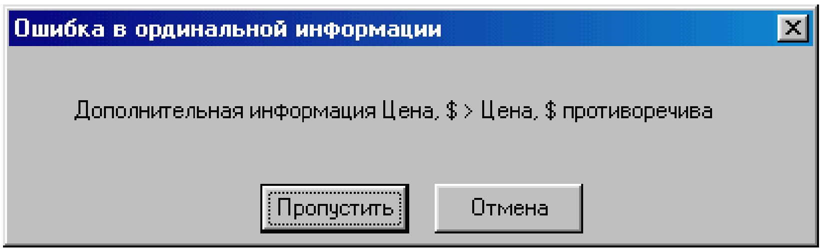


Рис. 3.132 Некорректность ординальной информации

Для игнорирования этой строки необходимо нажать кнопку «Пропустить», в этом случае программа продолжит дальнейшее создание строк ординальной информации в соответствии с выбранными критериями. Для отмены выбора следует нажать кнопку «отмена».

Противоречивой считается информация следующего вида:

Критерий 1 важнее (равноценен) Критерия 1.

Критерий 1 важнее (равноценен) Критерия 2 ,если в ординальной информации имеются строки, задающие отношение между критериями 1 и 2.

Если после дополнения информации до транзитивности возникают строки типа «Критерий 1 важнее Критерия 1» или одновременно Критерий 1 важнее Критерия 2 и Критерий 2 важнее Критерия 1.

Однако в последнем случае сообщение о противоречивости будет выдано только при запуске выбора.

Удаление ординальной информации о критериях

Для удаления строчки ординальной информации необходимо ,выбрав соответствующую строку, нажать клавишу **Del** или выбрать соответствующую команду в меню. При этом программа запросит подтверждение на удаление этой строки.

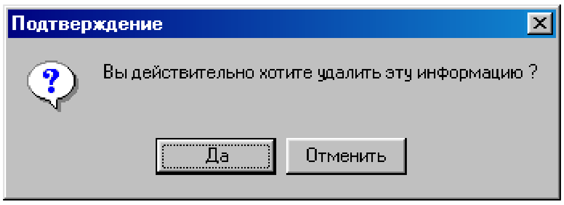


Рис. 3.133 Удаление строки ординальной информации

3.1.13 Нормализованные исходные данные.

В этом окне выводятся нормализованные исходные данные. Так как в методе Т-упорядочивания подразумевается однородность критериев, то предварительно необходимо их нормализовать. В данной системе используется следующий способ нормализации исходных данных.

Если критерий Р максимизируется, то нормализованное значение вычисляется по формуле:

$$\text{Альт}(i,j) = \frac{\text{Альт}^*(i,j) - Kp(j)\min}{Kp(j)\max - Kp(j)\min},$$

где Альт* – ненормализованные исходные данные,

Альт – нормализованные исходные данные

Kp(j)max, Kp(j)min – максимальные и минимальные значения критериев, заданные пользователем.

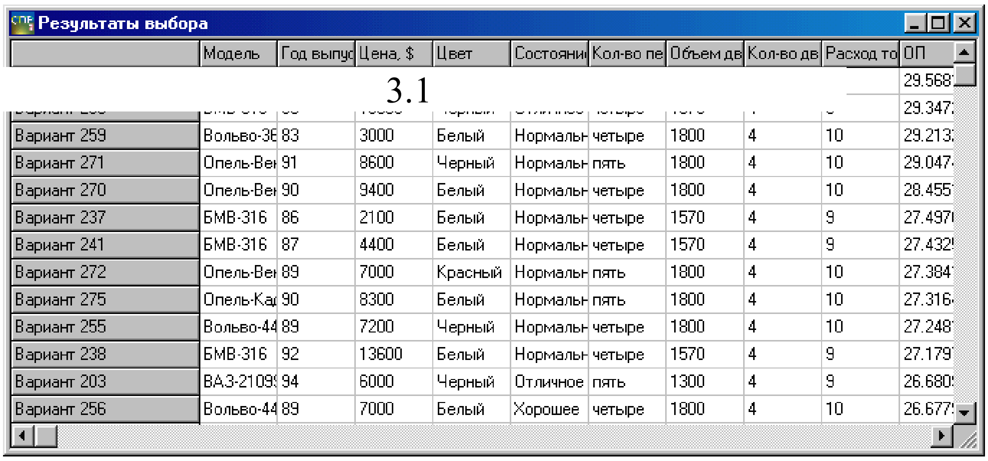
Если критерий Р минимизируется , то нормализованное значение вычисляется по формуле:

$$\text{Альт}(i,j) = \frac{Kp(j)\max - \text{Альт}^*(i,j)}{Kp(j)\max - Kp(j)\min}$$

где Альт* – ненормализованные исходные данные, Альт – нормализованные исходные данные

Кр(j)max, Кр(j)min – максимальные и минимальные значения критериев, заданные пользователем.

Внешний вид данного окна «нормализованные данные» расположен на рис. 3.34.

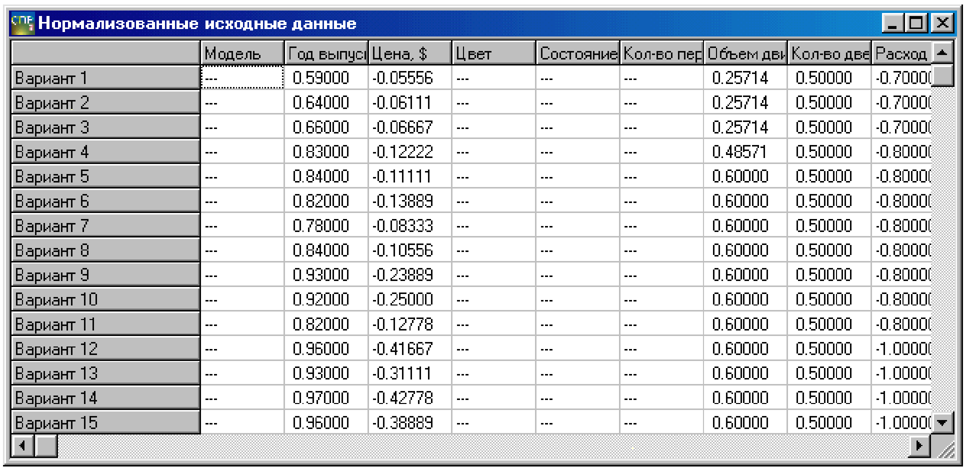


	Модель	Год выпус	Цена, $	Цвет	Состояни	Кол-во пе	Объем дв	Кол-во дв	Расход то	ОП
										29.568
										29.347
Вариант 259	Вольво-3	83	3000	Белый	Нормальн	четыре	1800	4	10	29.213
Вариант 271	Опель-Be	91	8600	Черный	Нормальн	пять	1800	4	10	29.047
Вариант 270	Опель-Be	90	9400	Белый	Нормальн	четыре	1800	4	10	28.455
Вариант 237	БМВ-316	86	2100	Белый	Нормальн	четыре	1570	4	9	27.497
Вариант 241	БМВ-316	87	4400	Белый	Нормальн	четыре	1570	4	9	27.432
Вариант 272	Опель-Be	89	7000	Красный	Нормальн	пять	1800	4	10	27.384
Вариант 275	Опель-Ка	90	8300	Белый	Нормальн	пять	1800	4	10	27.316
Вариант 255	Вольво-44	89	7200	Черный	Нормальн	четыре	1800	4	10	27.248
Вариант 238	БМВ-316	92	13600	Белый	Нормальн	четыре	1570	4	9	27.179
Вариант 203	ВАЗ-2109	94	6000	Черный	Отличное	пять	1300	4	9	26.680
Вариант 256	Вольво-44	89	7000	Белый	Хорошее	четыре	1800	4	10	26.677

Рис. 3.134 Нормализованные исходные данные

Окно «Предварительные Результаты»

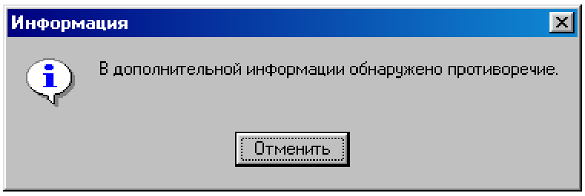
В окне «Результаты», расположенном на рис. 3.35, располагаются альтернативы, выбранные после предварительного выбора.



	Модель	Год выпус	Цена, $	Цвет	Состояние	Кол-во пер	Объем дви	Кол-во две	Расход
Вариант 1	---	0.59000	-0.05556	---	---	---	0.25714	0.50000	-0.7000
Вариант 2	---	0.64000	-0.06111	---	---	---	0.25714	0.50000	-0.7000
Вариант 3	---	0.66000	-0.06667	---	---	---	0.25714	0.50000	-0.7000
Вариант 4	---	0.83000	-0.12222	---	---	---	0.48571	0.50000	-0.8000
Вариант 5	---	0.84000	-0.11111	---	---	---	0.60000	0.50000	-0.8000
Вариант 6	---	0.82000	-0.13889	---	---	---	0.60000	0.50000	-0.8000
Вариант 7	---	0.78000	-0.08333	---	---	---	0.60000	0.50000	-0.8000
Вариант 8	---	0.84000	-0.10556	---	---	---	0.60000	0.50000	-0.8000
Вариант 9	---	0.93000	-0.23889	---	---	---	0.60000	0.50000	-0.8000
Вариант 10	---	0.92000	-0.25000	---	---	---	0.60000	0.50000	-0.8000
Вариант 11	---	0.82000	-0.12778	---	---	---	0.60000	0.50000	-0.8000
Вариант 12	---	0.96000	-0.41667	---	---	---	0.60000	0.50000	-1.0000
Вариант 13	---	0.93000	-0.31111	---	---	---	0.60000	0.50000	-1.0000
Вариант 14	---	0.97000	-0.42778	---	---	---	0.60000	0.50000	-1.0000
Вариант 15	---	0.96000	-0.38889	---	---	---	0.60000	0.50000	-1.0000

Рис. 3.35 Результаты выбора

Для запуска выбора необходимо воспользоваться главным меню или нажать клавишу **F9**. При этом осуществляется запуск алгоритма принятия решения и построения множества оптимальных вариантов. В случае если исходная информация некорректна будет выдано соответствующее сообщение об ошибке. Такая ошибка может быть вызвана некорректностью ординальной информации (рис. 3.36).



3.136 Ошибка некорректности ординальной информации

После правильного завершения работы алгоритма в окно «Информация» выводится статистика работы алгоритма, а также окно «результаты» становится активным. Альтернативы в окне упорядочены по обобщенному значению и в соответствии с данным порядком будут предлагаться в методе ограничений. ОЗ располагается в правом столбце таблицы выбранных альтернатив. Построение ОЗ кратко описано в пункте «Задание параметров метода ограничений». При большом (>10) количестве критериев, предварительный отбор может занимать достаточно много времени.

Также имеется возможность отключить предварительный отбор, а также задать максимальное время его прохождения. Если выбор будет продолжаться дольше установленного времени, система выдаст сообщение, изображенное на рис. 3.37.

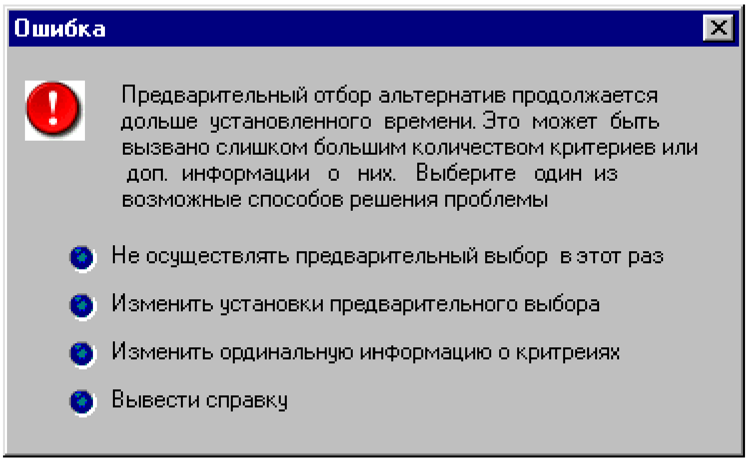


Рис. 3.37 Время предварительного отбора превысило допустимое

При этом пользователю предлагается выбрать один из вариантов:

Не осуществлять предварительный выбор в этот раз – предварительный выбор не будет закончен, однако, при повторном запуске он будет проводиться вновь.

Изменить установки предварительного выбора – будет запущен диалог задания параметров метода ограничений, в котором пользователь сможет либо совсем отключить предварительный выбор, либо изменить максимальное время его прохождения.

Изменить ординальнальную информацию о критериях – Будет запущен диалог, позволяющий пользователю отредактировать ординальную информацию о критериях. Это может позволить уменьшить время предварительного отбора.

Окно «информация»

Внешний вид этого окна показан на рис. 3.38. В нем располагаются область для ввода комментария о задаче, а также выводится статистика.

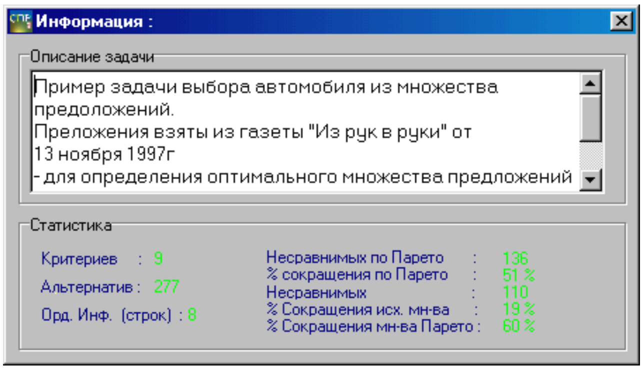


Рис. 3.38 Окно информация

3.1.14 Создание, загрузка и сохранение задачи

Создание новой задачи.

Для создания новой задачи необходимо выбрать пункт главного меню «Файл» – «Новый». Параметры, принимаемые при создании новой задачи, можно задать при помощи меню «Опции» – «Новая задача». При выборе этого пункта появляется диалог, показанный на рис. 3.39. Этот диалог позволяет задавать параметры, которые принимаются при создании новой задачи.

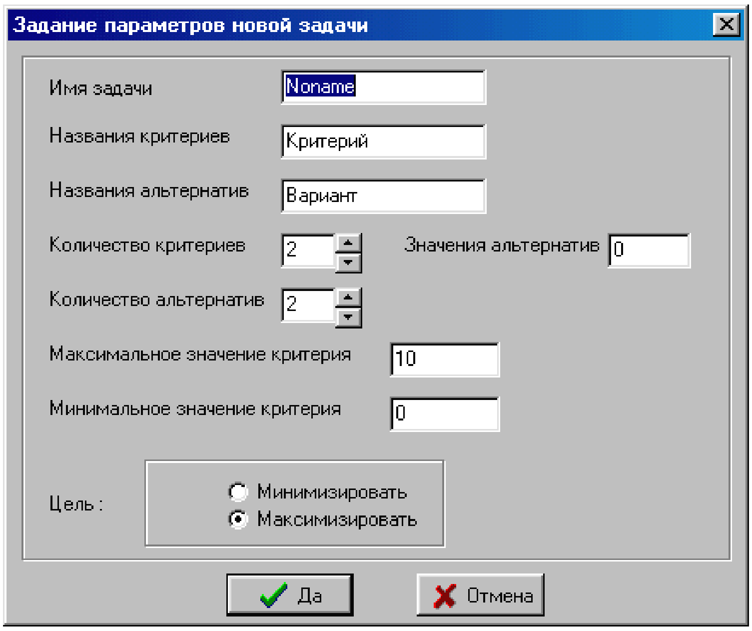


Рис. 3.39 Задание параметров новой задачи

Открытие существующей задачи.

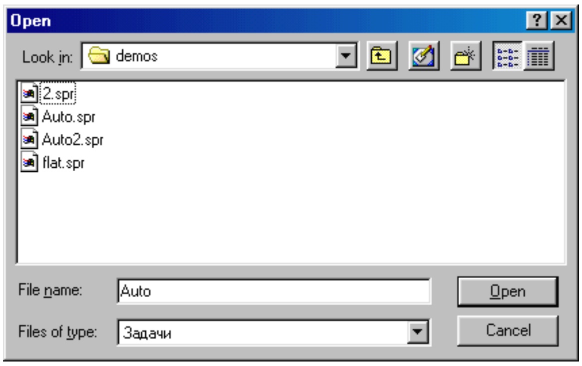


Рис. 3.40 Открытие существующей задачи.

Открытие существующей задачи производится при помощи меню «файл» – «открыть».При выборе этого пункта меню программа вызывает стандартный диалог, в котором запрашивается имя загружаемой задачи**.** Этот диалог показан на рис. 3.40.

Сохранение задачи

Сохранение задачи с текущим именем производится по команде главного меню «файл» – «сохранить». Данная команда позволяет сохранить задачу с установленным именем, показанным в верхней части главного окна. Если данный файл существует, программа спросит подтверждение на перезапись данного файла.

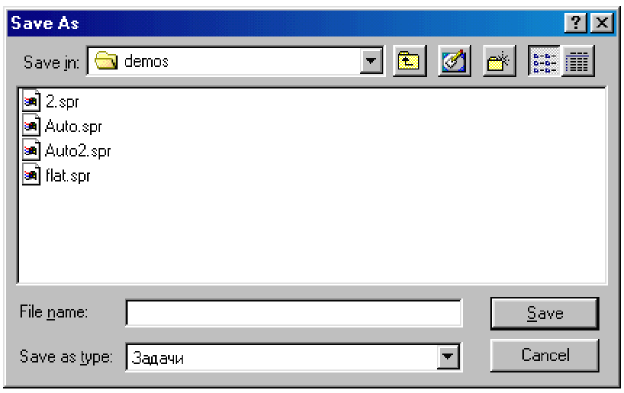


Рис. 3.41 Сохранение задачи

Для сохранения файла с новым именем следует воспользоваться командой «файл» – «сохранить как». Эта команда позволяет сохранить задачу, задав при этом новое имя файла. При этом вызывается стандартный диалог, в котором запрашивается имя файла. Этот диалог располагается на рис. 3.41.

3.1.15 Создание отчета

Данная система позволяет сохранить в файле описание критериев, альтернатив, ординальную информацию, а также результаты выбора. Для создания файла отчета необходимо выбрать в главном меню команду «Отчет» – «Создать отчет». Для создания отчета необходимо произвести выбор альтернативы при помощи метода ограничений. При создании отчета на экране появится диалог, изображенный на рис. 3.42. Данная система предполагает два типа отчетов:

Отчет в текстовом файле

Отчет в формате Microsoft Word

Для создания отчета в формате Microsoft Word необходимо, чтобы последний был уже установлен на машине. Следует заметить, что создание отчета в формате Word, при большом количестве альтернатив и критериев, может занять довольно продолжительное время.

В этом диалоге можно задать имя файла, в который будет помещен отчет. Также на этой панели можно указать программе компоненты, которые следует добавлять в отчет. Для добавления в отчет результатов выбора необходимо сначала провести вычисление.

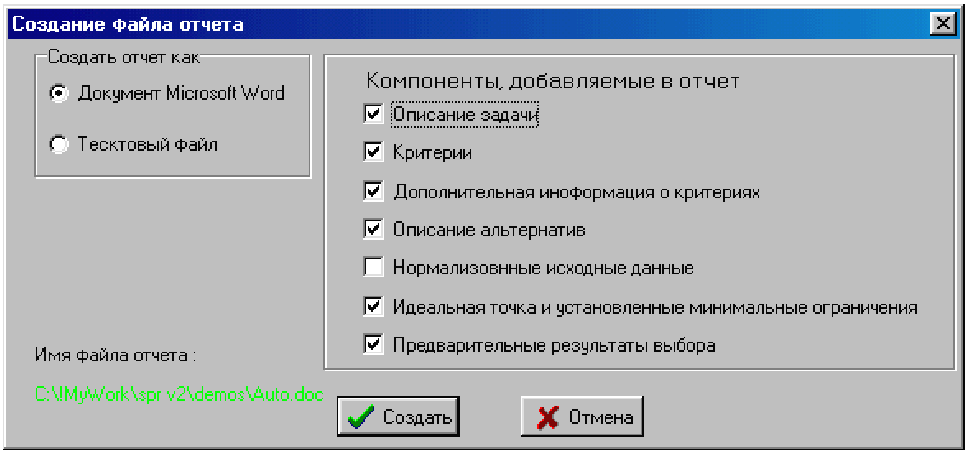


Рис. 3.42 **Создание файла отчета**

После задания параметров отчета следует нажать кнопку «Создать».После этого отчет будет создан в файле с заданным именем.

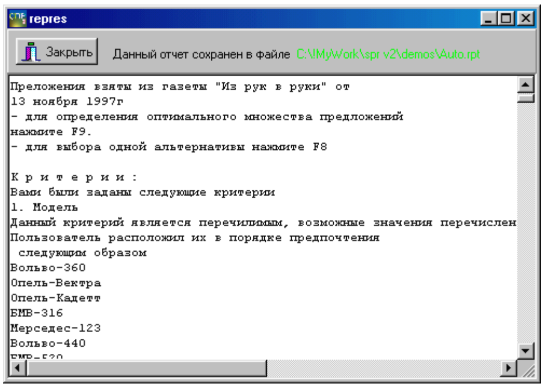


Рис. 3.43 Отчет в формате текстового файла

3.1.16 Пример решения задачи

В качестве примера рассмотрим следующую задачу. Имеется набор различных предложений по продаже автомобилей, пользователю необходимо выбрать наиболее устраивающий его вариант. Альтернативы описываются следующими критериями:

Модель
Год выпуска
Цена, $
Цвет
Состояние
Кол-во передач
Объем двигателя
Кол-во дверей
Расход топлива

Для каждого из вышеперечисленных критериев необходимо определить тип и задать соответствующие параметры. Для перечислимых критериев необходимо перечислить все принимаемые значения.

После запуска системы выбираем «Создание новой задачи в диалоге с системой».

Задаем количество критериев – 9 и вводим названия вышеперечисленных критериев.

Теперь система будет задавать вопросы по каждому из критериев. Например, для критерия «Модель».

Таблица 3.1

Вопрос системы	Ответ пользователя
«Какова ваша цель по критерию «Модель»?»	максимум.
«Каким образом вы хотите задать критерий «Модель»?»	«Перечислив все его значения».
Можете ли вы каждому перечисленному значению критерия	Нет

«Модель» сопоставить числовое значение?	

Затем необходимо задать все значения критерия «Модель» и расположить их в порядке предпочтения. После этого аналогичный диалог будет проведен для каждого из критериев. Возможные типы критериев и из свойства показаны в табл. 3.2, а принимаемые значения критериев в табл. 3.3.

Таблица 3.2

№	Критерий	Тип	Задание	Цель	Мин.	Макс.
1	Модель	Перечислимый	Отношением	Максимум	–	–
2	Год выпуска	Непрерывный	Численно	Максимум	0	100
3	Цена, $	Непрерывный	Численно	Минимум	0	18000
4	Цвет	Перечислимый	Отношением	Максимум	–	–
5	Состояние	Перечислимый	Отношением	Максимум	–	–
6	Кол-во передач	Перечислимый	Отношением	Максимум	–	–
7	Объем двигателя	Непрерывный	Численно	Максимум	750	2500
8	Кол-во дверей	Непрерывный	Численно	Максимум	2	6
9	Расход топлива	Непрерывный	Численно	Минимум	5	15

Таблица 3.3

Критерий	Принимаемые значения
Модель	БМВ – 520 БМВ – 320 Вольво –360 Опель-Вектра БМВ – 316 . . . . . М – 412 ИЖ – 412 ЗАЗ – 968М
Цвет	Черный Вишневый Белый Красный Синий Зеленый Неизвестно Бежевый Коричневый Желтый Голубой

На этом задание критериев закончено и необходимо приступить к заданию альтернатив. Система спросит количество альтернатив и затем попросит ввести значения альтернатив по каждому из критериев. В табл. 3.4 показан пример задания альтернатив.

Таблица 3.4

№	Модель	Год выпуска	Цена, $	Цвет	Состояние	Кол-во передач	Объем двигателя	Кол-во дверей	Расход топлива
Вариант 1	ГАЗ-21	59	1000	Корич.	На ходу	четыре	1200	4	12
Вариант 2	ГАЗ-21	64	1100	Синий	На ходу	четыре	1200	4	12
Вариант 3	ГАЗ-21	66	1200	Бежевый	На ходу	четыре	1200	4	12
Вариант 4	ГАЗ-24	83	2200	Красный	Хорошее	четыре	1600	4	13
Вариант 5	ГАЗ-24	84	2000	Голубой	Норм.	четыре	1800	4	13
Вариант 6	ГАЗ-24	82	2500	Белый	Хорошее	четыре	1800	4	13
Вариант 7	ГАЗ-24	78	1500	Белый	Норм.	четыре	1800	4	13
……………. ……………. ……………. ……………. ……………. ……………. ……………. …………….									
Вариант 275	Опель-Кадетт	90	8300	Белый	Нормальное	пять	1800	4	10
Вариант 276	Опель-Кадетт	88	8500	Белый	Нормальное	пять	1800	4	10
Вариант 277	Опель-Кадетт	94	4200	Желтый	Нормальное	пять	1800	4	10

После задания альтернатив требуется задать ординальную информацию о критериях. Пусть для ЛПР наиболее важными являются критерии «Цена» и «Год выпуска». Система предложит выбрать один из видов ординальной информации, которые описаны выше. Зададим информацию о важности одного критерия по сравнению с другими. Выберем критерий «цена». На вопрос: «По отношению к каким критериям он более важен?», – выберем «всех критериев»». Аналогичным образом зададим информацию о критерии «год выпуска». После задания дополнительной информации будет проведен предварительный выбор.

Затем будет запущен метод ограничений. Пользователю будет предложена одна из альтернатив. Например «Вариант 7». Пусть эта альтернатива не устраивает пользователя. Выбираем «Нет, эта альтернатива меня не устраивает». После этого система спросит критерий, по которому данная альтернатива не удовлетворяет пользователя (например, «Модель»), и спросит наихудшее значение данного критерия, которое удовлетворит пользователя. Далее будет предложена следующая альтернатива «Вариант 75». Эта альтернатива не удовлетворяет пользователя по цене. После задания верхнего предела цены осталось 2 альтернативы удовлетворяющие ЛПР. После согласия ЛПР с предлагаемой альтернативой система спросит те части задачи, которые пользователь хочет поместить в отчет и создаст необходимый отчет о задаче.

3.2 EXPERT – KN. ИНСТРУМЕНТАЛЬНОЕ СРЕДСТВО ПОСТРОЕНИЯ НЕЙЛОРОВСКИХ ДИАГНОСТИРУЮЩИХ СИСТЕМ.

3.2.1 Назначение и структура системы.

При разработке экспертных систем часто приходиться сталкиваться с ситуацией наличия неопределенности в исходных данных. Иными словами, нельзя с абсолютной уверенностью сказать, что некоторое утверждение истинно или ложно. Наличие этой неопределенности диктует необходимость сопоставлять фактам и правилам некоторые значения, описывающие их вероятностные характеристики – коэффициенты уверенности. Существуют различные способы работы в условиях неопределенности. Основные из них – это использование аппарата *нечеткой логики* и методика, основанная на *теории Байеса* (данная теория изложена в материале раздела X). Второй подход весьма неплохо зарекомендовал себя на практике.

Данное инструментальное средство предназначено для создания байесовских экспертных систем в произвольной предметной области при условии наличия ненадежных (неопределенных) данных.

Возможности системы

1) Создание и редактирование БЗ (базы знаний):

   - Редактирование общей информации об ЭС;
   - Добавление/удаление/модификация свидетельств;
   - Добавление/удаление/модификация гипотез;

2) Сохранение/загрузка БЗ в файле на диске. Поддерживаются следующие форматы файлов:

   *.nkb – текстовый формат. Описание БЗ на разработанном языке представления знаний;

   *.mdb – файл базы данных СУБД Microsoft Access;

3) Проверка корректности вводимых пользователем данных;
4) Создание комплекта инсталляционных дискет, необходимых для установки разработанной ЭС на ПК пользователя;
5) В процессе работы ЭС – отображение информации обо всех проделанных экспертной системой действиях, вероятностях гипотез и ценах свидетельств. Сохранение отчета о работе ЭС в файле;
6) Сохранение сеанса работы с ЭС в файле с возможностью последующей загрузки.

Общая структура программного средства

Разрабатываемое средство представляет собой совокупность двух подсистем:

- Редактор базы знаний;
- Оболочка экспертной системы.

Редактор базы знаний (главное окно представлено на рис.17.1.2.1) предназначен для создания, редактирования и последующего сохранения на диске в одном из поддерживаемых форматов файлов базы знаний. Созданный файл базы знаний разрабатываемой экспертной системы может быть при необходимости загружен редактором и модифицирован. Следует отметить, что в соответствии с терминологией принятой в теории ЭС, *пользователь редактора БЗ* выступает в качестве *эксперта* или *инженера по знаниям*.

Пользователь ЭС, как конечного результата работы описываемого инструментального средства, не имеет необходимости работать с редактором БЗ. Для него предназначена оболочка экспертной системы.

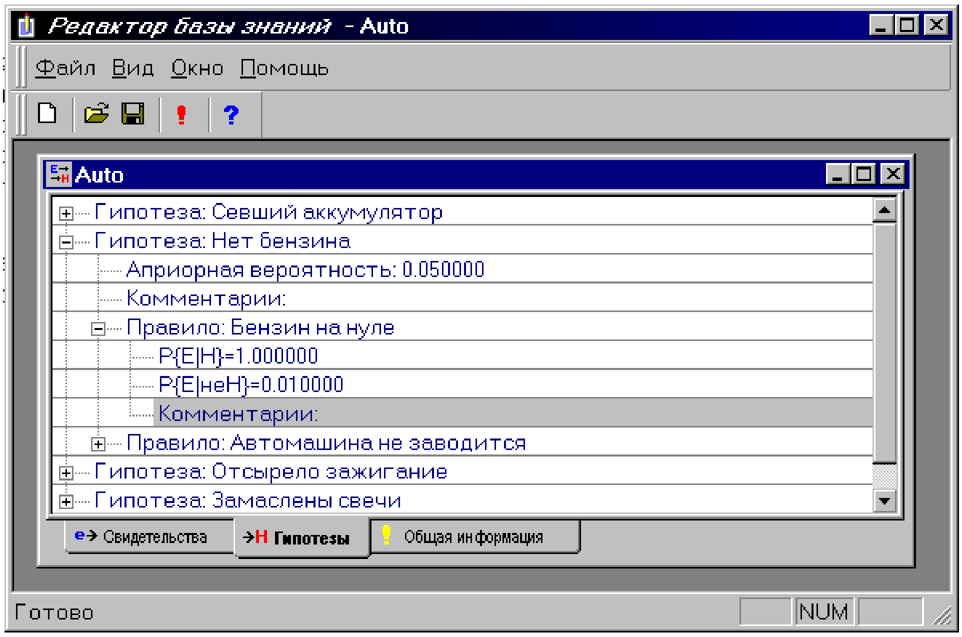


Рис. 3.24 Главное окно редактора БЗ

Оболочка экспертной системы (главное окно представлено на рис.3.45) реализует машину логического вывода и содержит интерфейс пользователя, необходимый для его общения с экспертной системой. Данная подсистема представляет собой «пустую экспертную систему», которая после «наполнения» знаниями (посредством загрузки файла базы знаний) «становится» экспертной системой в конкретной предметной области. В процессе работы экспертная система, в соответствии с алгоритмом вывода, выбирает вопросы к пользователю, получает на них ответы, вычисляет вероятности гипотез.

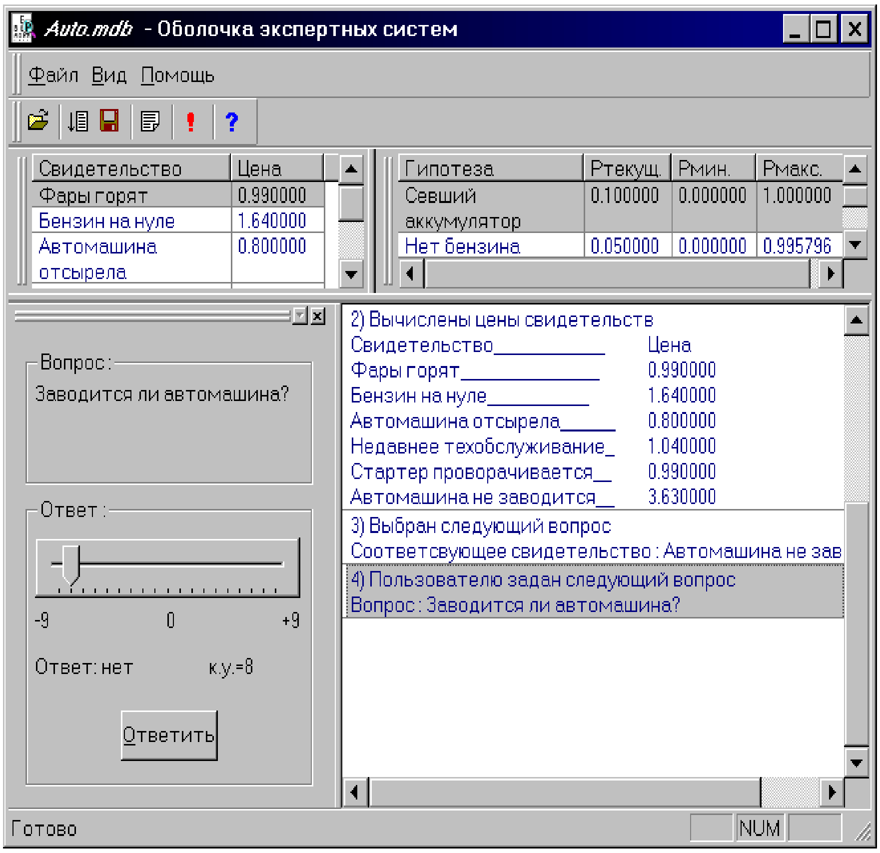
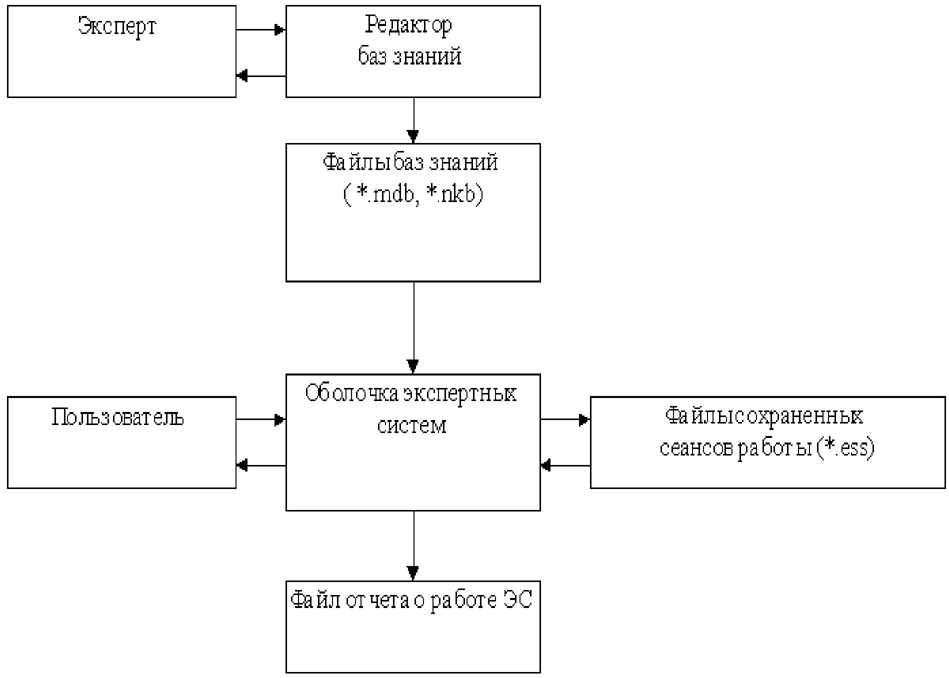


Рис. 3.43 Главное окно оболочки ЭС

При нахождении решения система сообщает об этом пользователю. В случае если система приходит к выводу, что решение не может быть найдено, она также информирует об этом пользователя. Во время работы система отображает информацию о производимых действиях, вероятностях гипотез и ценах свидетельств в удобной для пользователя форме. Также имеется возможность сохранять в файле текущее состояние процесса поиска решения и, впоследствии, восстановить это состояние, загрузив его из соответствующего файла. Кроме того, пользователь имеет возможность сохранять в файле отчет о работе для его последующего изучения. Ниже приведена схема взаимодействия программного средства и пользователя.



Следует также отметить, что имеется возможность запускать оболочку ЭС из редактора БЗ. При этом оболочка автоматически загружает редактируемую базу знаний. Далее пользователь, на основе анализа работы экспертной системы, может, если необходимо, вносить соответствующие изменения в БЗ.

Редактор БЗ и оболочка ЭС разработаны для платформы Win32 (операционные системы Windows 95/98/2000, Windows NT v.4.0).

Конечным результатом работы данного инструментального средства является конкретная экспертная система в некоторой предметной области. На базе инструментального средства могут быть созданы экспертные системы в таких предметных областях, как медицина, вычислительная техника, геология, математика, сельское хозяйство, управление, электроника, юриспруденция и др.

3.2.2 Общая характеристика системы.

Необходимые ресурсы, язык и система программирования

Для работы системы необходима следующая минимальная конфигурация: 486 процессор и 8Мб ОЗУ. После инсталляции система занимает на диске 10 –12Мб, в зависимости от количества устанавливаемых компонент. Для работы системы необходимо наличие на компьютере операционной системы Windows 95/98/2000 или Windows NT 4.0.

Данная система создания байесовских экспертных систем была реализована на языке Си ++, в системе программирования Visual C++ 5.0.

Представление знаний в экспертных системах

База знаний состоит из трех блоков:

Общая информация;

Список гипотез;

Список свидетельств.

В блоке (1) может содержаться информация о назначении экспертной системы, ее разработчиках и тому подобное. Этот блок необязателен.

В блоке (2) содержится последовательность описаний гипотез.

В описании каждой гипотезы должна содержаться следующая информация:

Название гипотезы;

Априорная вероятность гипотезы;

Комментарии, описание гипотезы (например, в случае медицинской базы знаний это может быть описание соответствующей болезни);

Список описаний правил вида «если E, то H».

В описании каждого правила должна содержаться следующая информация:

Номер соответствующего свидетельства E в списке свидетельств;

P{E | H};

P{E | не H};

Комментарии.

В блоке (3) содержится последовательность описания свидетельств.

В описании каждого свидетельства должна содержаться следующая информация:

Название свидетельства;

Текст вопроса, задаваемого пользователю при необходимости получить данное свидетельство;

Комментарии, описание свидетельства.

На основе указанных выше положений были разработаны два формата представления знаний, предназначенных для хранения БЗ на диске:

Описание БЗ на языке представления знаний;

Описание БЗ в файле базы данных СУБД Microsoft Access.

Каждый из этих форматов может быть использован для хранения БЗ экспертной системы.

Характеристики решаемых задач и квалификация пользователя

Данное программное средство может быть применено для создания экспертных систем в любых областях человеческой деятельности, где возникает ситуация наличия ненадежных данных. Для работы с ненадежными данными был реализован подход, основанный на теории

Байеса. Система не требует от пользователя специальных знаний в области искусственного интеллекта, экспертных систем или в теории вероятностей. Все необходимые сведения для работы с программным продуктом содержатся в пользовательской документации. Система имеет удобный пользовательский интерфейс, систему установки (инсталляции) на ПК пользователя, поддерживает различные форматы файлов БЗ, позволяет создавать и просматривать отчет о работе ЭС.

В случае описания БЗ на языке представления знаний – БЗ хранится в текстовом файле. При этом ее различные элементы описываются *конструкциями языка представления знаний.* Этот формат может быть использован для хранения БЗ небольших размеров (не более 100 – 200 правил). При этом размер соответствующего дискового файла будет также невелик (порядка 10 KB) и будет обеспечено приемлемое время загрузки (несколько секунд).

Описание БЗ в файле базы данных СУБД Microsoft Access лучше применять при больших размерах БЗ (несколько сотен правил и более). При этом будет обеспечено приемлемое время загрузки (несколько секунд).

3.2.3 Инсталляция системы.

Для установки инструментального средства создания байесовских ЭС необходимо запустить исполняемый файл «setup.exe» с диска 1 из комплекта инсталляционных дискет. После запуска программа выведет на экран краткую информацию об устанавливаемом программном средстве, затем активизирует диалоговое окно «Choose destination location».

Шаг 1. Choose destination location. Данное окно предназначено для выбора директория, в который необходимо установить систему. В нижней части этого окна будет указан директорий по умолчанию («X:\Program Files\Expert System Shell\», где X – имя системного диска). Если Вас не устраивает данный директорий, нажмите кнопку «Browse» («Обзор») и в открывшемся диалоговом окне выберете нужный директорий. После того, как директория будет выбрана, нажмите кнопку «Next».

Шаг 2. Select program folder. Здесь вы можете указать название подпункта в меню «Start/Programs» («Пуск/Программы»), который будет добавлен программой установки. Значение по умолчанию «Инструментальное средство создания ЭС». Затем снова нажмите кнопку «Next» и программа установки начнет копирование файлов.

Шаг 3. Setup complete. После окончания копирования файлов будет активизировано диалоговое окно, содержащее информацию об окончании установки. Нажмите кнопку «Finish» – установка завершена.

Программой установки в меню «Пуск/Программы» будет добавлен пункт «Инструментальное средство создания ЭС». Он содержит три подпункта «Руководство пользователя», «Редактор баз знаний» и «Оболочка экспертных систем». Путем выбора этих пунктов Вы можете запускать соответствующие подсистемы инструментального средства.

В указанном Вами директории будут расположены следующие файлы:

X:\USER_PATH\BIN

kbedit.exe – редактор баз знаний;

esshell.exe – оболочка ЭС;

X:\USER_PATH\SAMPLES

auto.nkb – база знаний в текстовом формате;

auto.mdb – база знаний в формате БД Microsoft Access;

auto.ess – сохраненный сеанс работы с ЭС;

auto.rep – файл отчета о работе с ЭС;

X:\USER_PATH\DOC

user_man.doc – руководство пользователя в формате MS Word v7.0;

X:\USER_PATH\INSTALL

файлы, необходимые для создания инсталляционных дискет. Файлы для дискеты 1 расположены в подкаталоге **\disk1**, для дискеты 2 – в подкаталоге **\disk2,** для дискеты 3 – в подкаталоге **\disk3**.

Следует помнить, что <u>данный программный продукт должен быть переписан на Ваш компьютер только с помощью программы установки</u>. Не допускается копирование файлов инструментального средства на другой компьютер, программа не будет правильно функционировать. При запуске исполняемых файлов инструментального средства будет произведена проверка того, было ли оно установлено на данном компьютере, и при отрицательном результате будет выдано соответствующее сообщение пользователю, после чего программа завершит свою работу.

3.2.4 Система в конкретной предметной области.

Для настройки инструментального средства на конкретную предметную область, в которой и предполагается использование новой экспертной системы, используется редактор баз знаний. Как было сказано выше, редактор баз знаний необходим для создания, редактирования и сохранения в файле базы знаний разрабатываемой ЭС (накопленные знания в данной предметной области).

Редактор БЗ является приложением операционных систем Windows 95, Windows 98, Windows 2000, Windows NT v.3.5 и Windows NT v.4.0. Исполняемый файл редактора БЗ называется KBEdit.exe (Knowledge Base Editor). Запустить редактор БЗ можно, выбрав пункт меню «Пуск/Программы/Инструментальное средство создания ЭС/Редактор баз знаний».

Главное окно

При запуске «Редактора баз знаний» на экране появится окно заставки с информацией о программе и разработчике. После того, как пользователь нажмет кнопку манипулятора «мышь» или любую клавишу, окно заставки исчезнет и появится главное окно программы (рис. 3.44).

Главное окно редактора содержит следующие элементы:

Заголовок;

Меню;

Панель инструментов;

Панель состояния;

Дочерние окна редактирования БЗ.

Система меню

Редактор БЗ является многодокументным приложением. Это означает, что в нем могут одновременно редактироваться несколько баз знаний.

Для приложений такого типа характерно наличие двух различных меню. Первое из них – основное меню – является активным (видимым) в том случае, когда редактор БЗ не содержит ни одного дочернего окна редактирования БЗ. В противном случае – главное меню заменяется меню документа.

Основное меню.

Основное меню имеет следующую структуру:

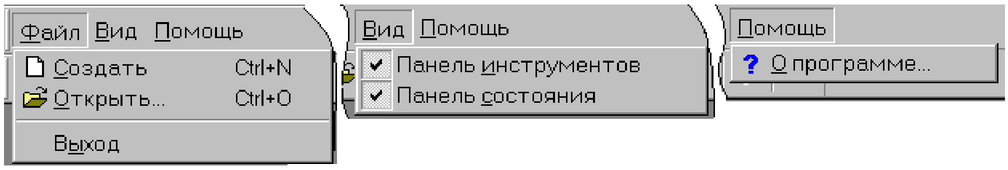


Рис. 3.44 Структура основного меню

Ниже приводится краткое описание этих пунктов:

«Файл/Создать» – создать новую БЗ;

«Файл/Открыть» – загрузить БЗ из файла;

«Файл/Выход» – закончить работу;

«Вид/Панель инструментов» – этот пункт меню является переключателем. Если панель инструментов является видимой, то в левой части этого пункта меню отображается небольшая «галочка»;

«Вид/Панель состояния» – действует аналогично пункту «Панель инструментов», но для панели состояния;

«Помощь/О программе» – при выборе этого пункта открывается диалоговое окно с информацией о названии программы, разработчике, номере версии. Вид окна приведен ниже:

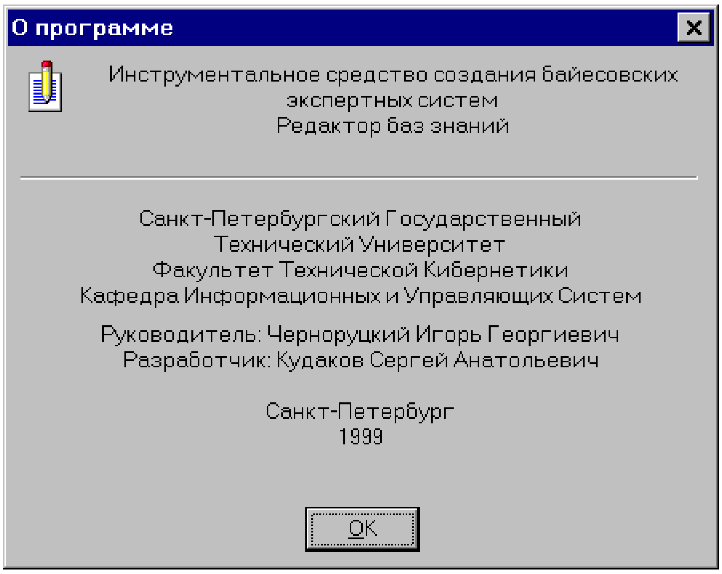


Рис. 3.45 Диалоговое окно “О программе”

Меню документа

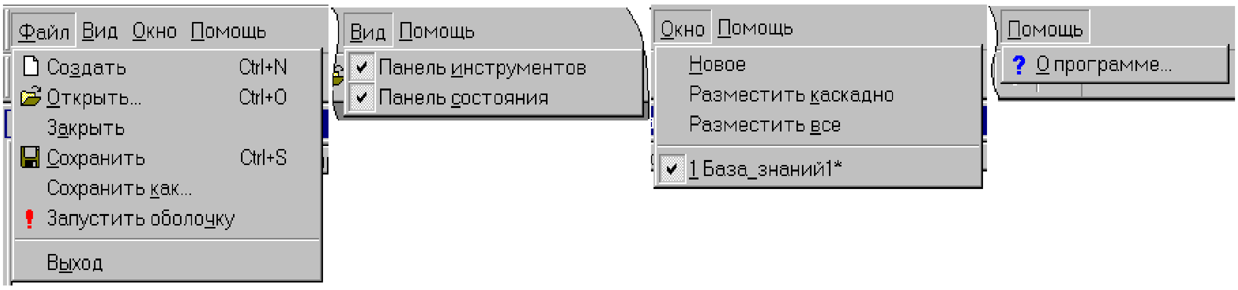


Рис. 3.46 Структура меню документа

Ниже приводится краткое описание этих пунктов:

«Файл/Создать» – создать новую БЗ;

«Файл/Открыть» – загрузить БЗ из файла;

«Файл/Закрыть» – при выборе этого пункта активное дочернее окно будет закрыто. Если редактируемая в этом окне БЗ не сохранена, приложение предложит сохранить ее в файле;

«Файл/Сохранить» – сохранить редактируемую БЗ в файле на диске. Если БЗ еще не была сохранена после создания, то приложение запросит имя файла, в котором будет сохранена БЗ;

«Файл/Сохранить как» – аналогично «Файл/Сохранить», но имя файла запрашивается всегда;

«Файл/Запустить оболочку ЭС» – при выборе этого пункта будет запущена оболочка экспертной системы, при запуске она автоматически загрузит редактируемую БЗ. Если база знаний не была сохранена после модификации, то приложение сначала предложит пользователю сохранить БЗ в файле;

«Файл/Выход» – завершить работу;

«Вид/Панель инструментов» – аналогично соответствующему пункту основного меню;

«Вид/Панель состояния» – аналогично соответствующему пункту основного меню;

«Окно/Новое» – открывает копию активного дочернего окна редактирования БЗ;

«Окно/Разместить каскадом» – при выборе этого пункта дочерние окна будут размещены одно над другим с перекрытием;

«Окно/Разместить все» – при выборе этого пункта дочерние окна будут размещены так, чтобы они не перекрывались;

«<u>1</u> <Окно 1>, <u>2</u> <Окно 2>...» – при выборе одного из этих пунктов (их количество соответствует количеству открытых дочерних окон), соответствующее дочернее окно станет активным;

«Помощь/О программе» – аналогично соответствующему пункту основного меню.

Панель инструментов

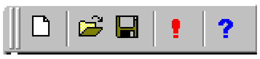


Рис. 3.47 Панель инструментов

Панель инструментов содержит ряд небольших кнопок с различными рисунками. Нажатие этих кнопок имеет те же последствия, что и выбор соответствующих пунктов меню. При наведении указателя манипулятора «мышь» на одну из кнопок панели инструментов, появляется короткая подсказка о назначении этой кнопки. Далее приведено соответствие пунктов меню и кнопок панели инструментов:

 – «Файл/Создать»

 – «Файл/Открыть»

 – «Файл/Сохранить»

 – «Файл/Запустить оболочку ЭС»

 – «Помощь/О программе...»

Панель состояния

Панель состояния представлена на рис. 3.50. Она расположена в нижней части главного окна. Панель состояния отображает краткую информацию о состоянии приложения или описание действия соответствующего пункту меню, на котором находится указатель манипулятора «мышь».

Рис. 3.50 Панель состояния

В правой части панели состояния расположены три небольшие окна, содержащие информацию о состоянии клавиш «Num Lock», «Caps Lock» и «Scroll Lock».

Дочерние окна редактирования БЗ

Дочерние окна отображают загруженную БЗ и предоставляют возможность ее редактировать (рис. 3.51).

Верхняя часть дочернего окна редактирования БЗ содержит заголовок. Заголовок содержит пиктограмму и имя файла, из которого была загружена БЗ (или в котором она была сохранена). Если после создания БЗ еще не была сохранена, то вместо имени файла заголовок содержит строчку «База_знанийN», где N – номер созданной БЗ в течение текущего сеанса работы программы. Кроме того, если база знаний была модифицирована и не сохранена после этого, к имени файла прибавляется символ «*». После сохранения он исчезает.

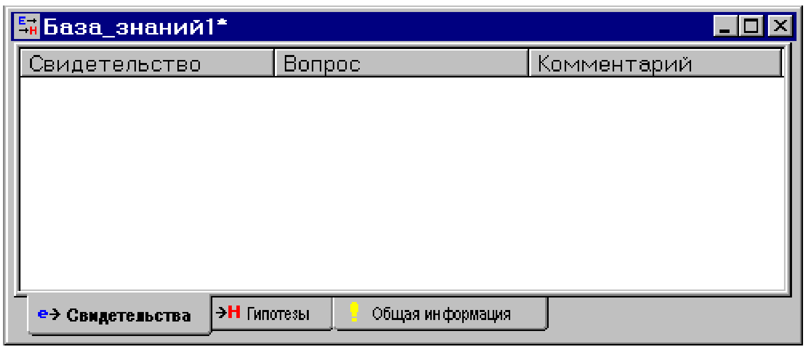


Рис. 3.51 Дочернее окно редактирования БЗ

База знаний в разрабатываемом инструментальном средстве состоит из трех частей:

Общая информация;

Описание свидетельств;

Описание гипотез.

Для каждой из этих частей имеется закладка в окне редактирования БЗ.

Панель с закладками расположена в нижней части окна. Для переключения закладки необходимо навести указатель манипулятора «мышь» на нужную закладку и щелкнуть левой кнопкой «мыши». После этого содержимое окна меняется и начинает отображать соответствующую часть БЗ.

Загрузка БЗ

Для того, чтобы загрузить БЗ, необходимо выбрать пункт «Файл/Открыть» в меню приложения или нажать соответствующую кнопку панели инструментов. Откроется диалоговое окно выбора файла для загрузки (рис. 3.52).

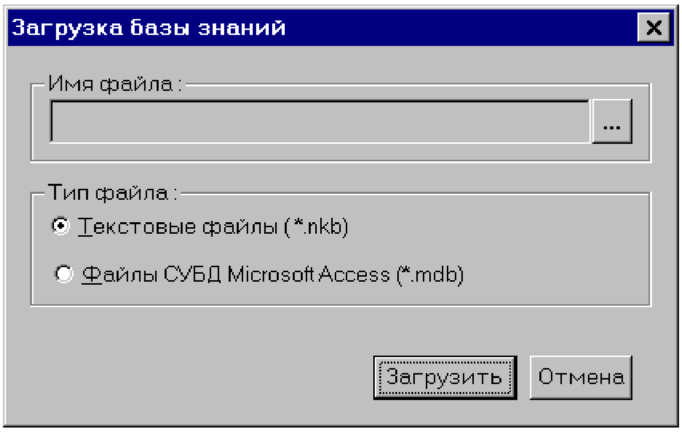


Рис. 3.52 Диалоговое окно выбора файла для загрузки

Для загрузки БЗ необходимо с помощью переключателя выбрать тип файла (*.nkb или *.mdb), затем нажать на кнопку справа от поля редактирования «Имя файла», после чего откроется стандартный диалог открытия файла. Когда файл будет выбран, его имя отобразится в поле редактирования «Имя файла».

После нажатия кнопки «Загрузить», редактор попытается загрузить базу знаний. Если во время загрузки в файле будут обнаружены ошибки – приложение откроет диалоговое окно с информацией об ошибке (имя файла, номер строки с ошибкой, строка с ошибкой и описание ошибки). Пример окна с сообщением об ошибке изображен на рис. 3.53.

Следует заметить, что в некоторых случаях информация об ошибочной строке не имеет смысла (например, в случае системных ошибок работы с файлом). В таком случае это диалоговое окно с сообщением об ошибке будет содержать только имя файла и описание ошибки.

Коды возможных ошибок при загрузке и сохранении файлов, используемых данным инструментальным средством, представлены ниже.

В случае если загрузка прошла успешно – откроется новое дочернее окно редактирования БЗ, заполненное информацией о загруженной базе знаний.

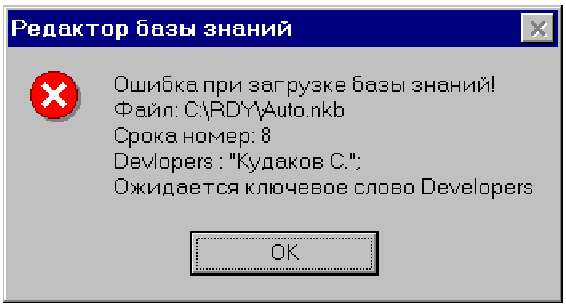


Рис. 3.53 Диалоговое окно с сообщением об ошибке

Сохранение БЗ

Для того, чтобы сохранить редактируемую БЗ, необходимо выбрать пункт меню «Файл/Сохранить» или «Файл/Сохранить как». В первом случае имя файла, в котором будет сохранена база знаний, будет запрошено только, если она еще не была сохранена после создания. Во втором – имя файла будет запрошено в любом случае.

Для сохранения БЗ необходимо с помощью переключателя выбрать тип файла (*.nkb или *.mdb), затем нажать на кнопку справа от поля редактирования «Имя файла», после чего откроется стандартный диалог сохранения файла. Когда файл будет выбран, его имя отобразится в поле редактирования «Имя файла».

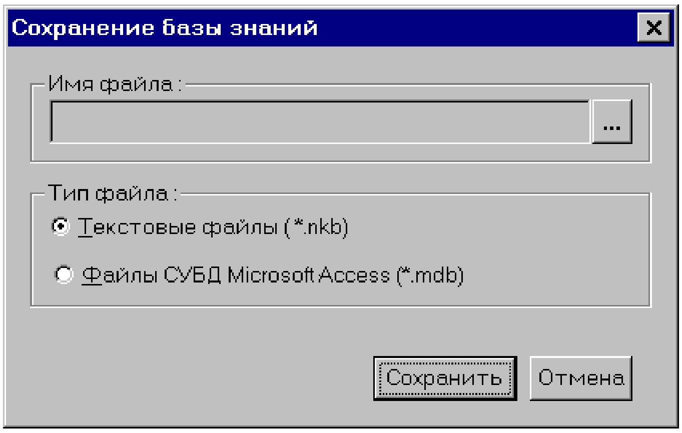


Рис. 3.54 Диалоговое окно выбора файла для сохранения

Если в процессе сохранения БЗ возникнут ошибки, приложение выведет диалоговое окно с сообщением об этом.

Коды ошибок при загрузке и сохранении файлов, используемых инструментальным средством

Лишние символы после логического конца файла;

Недопустимое значение Kmin;

Недопустимое значение Kmax;

Ожидается } или Hypothes;

Ожидается } или Evidence;

Ожидается } или Rule;

Недопустимое имя гипотезы;

Недопустимое имя свидетельства;

Недопустимое значение априорной вероятности;

Недопустимое имя свидетельства в описании гипотезы;

Ссылка на неизвестное свидетельство в описании гипотезы;

Недопустимое значение P{E|H};

Недопустимое значение P{E|неH};

Ожидается ключевое слово;

Неожидаемый конец файла;

Ожидается ключевое слово «"»;

Недопустимое значение количества градаций в ответах пользователя;

Таблица GeneralInfo должна содержать ровно одну запись;

Недопустимый текст вопроса в описании свидетельства;

Ссылка на неизвестную гипотезу в описании правила;

Неправильный формат файла сохранения сессии.

Редактирование БЗ

Переходя непосредственно к созданию базы знаний для экспертной системы в конкретной предметной области, Вы выступаете в роли *эксперта* или *инженера по знаниям.* На данном этапе для работы с базой знаний ЭС предназначено дочернее окно редактирования БЗ (рис. 3.51). Оно содержит три закладки – для редактирования свидетельств, гипотез и ввода общей информации.

Например, если речь идет о медицинской экспертной системе, то вариантам решения будут соответствовать различные заболевания (*гипотезы*). Также имеется некоторое количество *свидетельств*, где свидетельство является доказательством, которое может или не может подтвердить правильность указанной гипотезы. Информация о гипотезах и свидетельствах задаётся пользователем редактора БЗ на соответствующих закладках дочернего окна редактирования. База знаний также должна содержать информацию, определяющую связь между свидетельствами и гипотезами. Эта связь задается продукциями (правилами) вида «Если E, то H», где E – некое свидетельство, а H – гипотеза. В примере с медицинской ЭС в качестве свидетельств будут выступать симптомы заболеваний (например, температура тела, наличие или отсутствие кашля и тому подобное).

Рассмотрим методы редактирования свидетельств, гипотез и общей информации об ЭС в редакторе баз знаний инструментального средства создания байесовских систем.

Редактирование свидетельств

Дочернее окно редактирования БЗ с активной закладкой свидетельства и редактируемым полем «Вопрос» изображено на рис. 3.55. В этом состоянии окно содержит таблицу со списком свидетельств. Таблица содержит следующие столбцы:

Свидетельство – в этом поле содержится название свидетельств;

Вопрос – текст вопроса (без вопросительного знака), который будет задан пользователю оболочкой ЭС, при необходимости получить информацию об этом свидетельстве;

Комментарии – некоторая сопроводительная информация, разъяснения (если они необходимы).

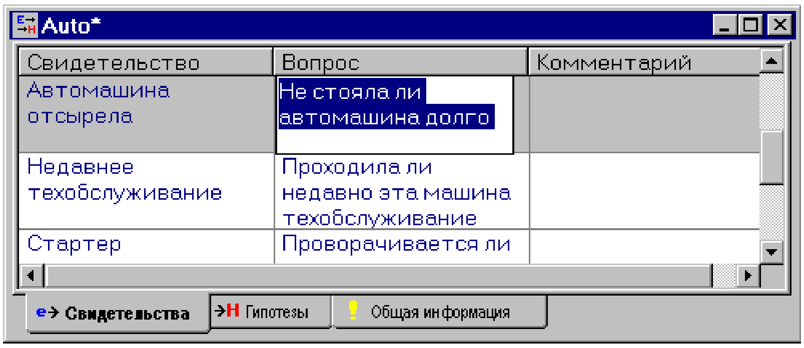


Рис. 3.55 Дочернее окно редактирования БЗ. Активна закладка "Свидетельства"

Для редактирования свидетельств БЗ существует два способа.

Редактирование с помощью встроенного редактора (built-in editing). Для того чтобы изменить одно из полей свидетельства необходимо:

- Выделить это свидетельство (с помощью клавиш со стрелками «Вверх», «Вниз» или манипулятора «мышь»);
- Произвести одинарный щелчок левой кнопкой «мыши» над редактируемой записью поля и на некоторой время задержать «мышь» в этой позиции;
- После этого на месте записи появится окно редактирования, в котором будет находиться текст из поля (рис. 3.55);
- Для окончания редактирования необходимо произвести одинарный щелчок «мышью» в любом месте кроме этого поля;
- Если введенное значение допустимо – оно заменит предыдущее, в противном случае приложение выведет диалоговое окно с информацией об ошибке, и ввод будет отменен;
- Допустимые значения для полей «Свидетельство» и «Вопрос» – алфавитно-цифровые символы, знак подчеркивания и пробел. Эти поля не могут быть пустыми. Кроме того, не может быть двух (и

более) свидетельств с одинаковым названием. Поле «Комментарий» может содержать любые символы или не содержать их вообще.

Второй способ редактирования – *использование контекстного меню* (рис. 3.56).

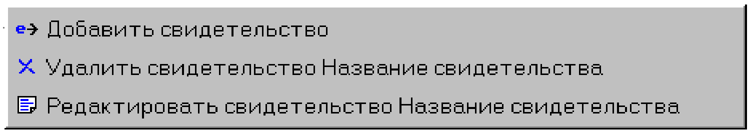


Рис. 3.56 Контекстное меню свидетельств

Для появления контекстного меню на экране, необходимо нажать на правую кнопку «мыши» в пределах окна. При этом если указатель мыши будет расположен над одной из строчек таблицы – все пункты этого меню будут доступны, в противном случае будет доступен только пункт «Добавить свидетельство».

Рассмотрим подробнее каждый пункт этого меню.

«Добавить свидетельство»

При выборе этого пункта откроется диалог добавления свидетельств, изображенный на рис. 3.57. После введения значений в соответствующие поля необходимо нажать кнопку «ОК».

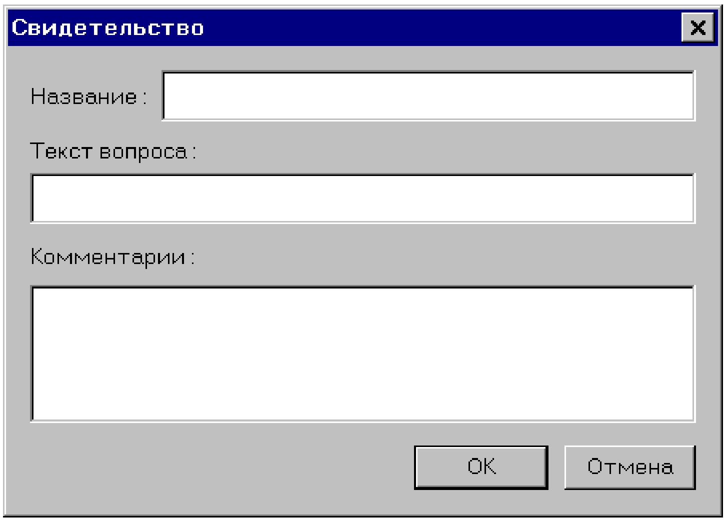


Рис. 3.57 Диалог добавления/модификации свидетельств

Будет проверена допустимость введенных значений (по правилам, указанным в описании первого способа редактирования свидетельств) и, если они удовлетворяют указанным правилам, свидетельство будет добавлено в базу знаний. В противном случае будет открыто диалоговое окно с сообщением об ошибке, свидетельство не будет добавлено.

«Удалить свидетельство <Название свидетельства>»

Приложение запросит подтверждение и при утвердительном ответе удалит соответствующее свидетельство.

«Редактировать свидетельство <Название свидетельства>»

Откроется диалоговое окно добавления/модификации свидетельств (рис. 3.257). Его поля будут заполнены прежними значениями. После редактирования и нажатия кнопки «ОК», будет проверена допустимость введенных значений и, если они удовлетворяют указанным ранее правилам, свидетельство будет изменено. В противном случае будет открыто диалоговое окно с сообщением об ошибке, свидетельство не будет изменено.

Редактирование гипотез и правил

Дочернее окно редактирования БЗ с активной закладкой «Гипотезы» приведено на рис. 3.58.

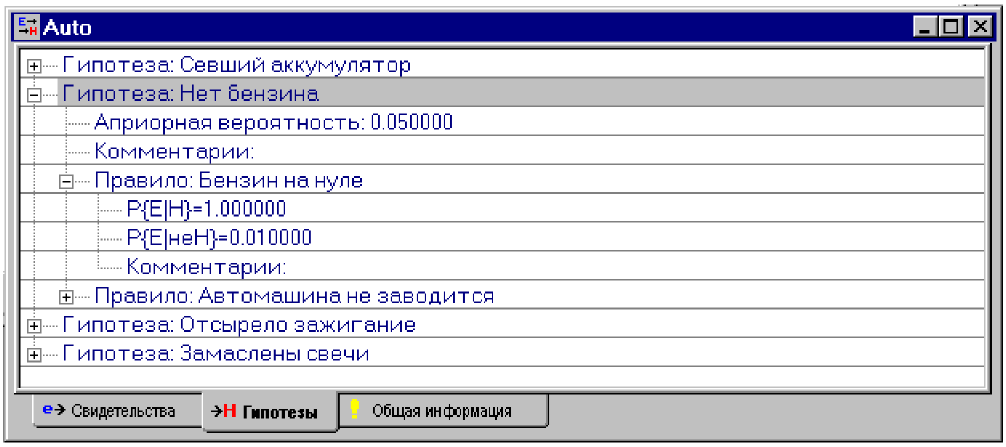


Рис. 3.58 Дочернее окно редактирования БЗ. Активна закладка "Гипотезы"

Как видно из рис. 3.58, гипотезы базы знаний в дочернем окне стандартного редактора БЗ представлены в виде списка иерархических объектов. Объекты данной иерархии расположены следующим образом:

Гипотеза 1 (включая название, априорную вероятность и комментарии);

Правило 1 (название соответствующего этому правилу свидетельства; P{E|H}; P{E|неH} и комментарии);

Правило 2 (название соответствующего этому правилу свидетельства; P{E|H}; P{E|неH} и комментарии);

….

Гипотеза 2 (включая название, априорную вероятность и комментарии);

….

и так далее…

где

– P{E|H} – условная вероятность истинности свидетельства, соответствующего этому правилу, при условии, что гипотеза, которой принадлежит это правило, истинна;

– P{E|неH} – условная вероятность истинности свидетельства, соответствующего этому правилу, при условии, что гипотеза, которой принадлежит это правило, ложна.

Редактирование гипотез и правил, принадлежащих этим гипотезам, осуществляется с помощью контекстного меню:

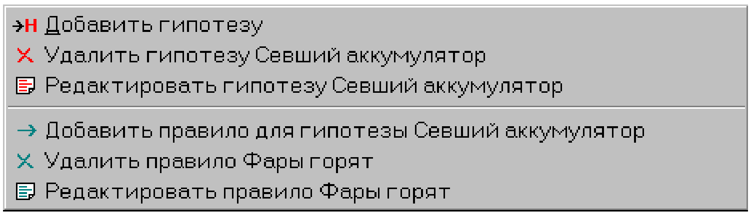


Рис. 3.59 Контекстное меню гипотез и правил

Рассмотрим подробнее каждый пункт этого меню.

«Добавить гипотезу»

Данный пункт меню доступен всегда. При его выборе открывается диалоговое окно редактирования гипотез (рис. 3.60). Способ работы с этим окном аналогичен способу работы с диалогом редактирования свидетельств, описанном в предыдущем разделе.

«Удалить гипотезу...», «Редактировать гипотезу...» и «Добавить правило для гипотезы...»

Эти пункты доступны в случае, когда контекстное меню было активизировано над гипотезой, а также над любым ее элементом: априорная вероятность, комментарий, правило и любой элемент правила (P{E|H}, P{E|неH}, комментарий).

«Удалить правило…» и *«Редактировать правило…»*

Данные пункты доступны в случае, когда меню активизировано над правилом, а также над любым его элементом (P{E|H}, P{EнеH}, комментарий).

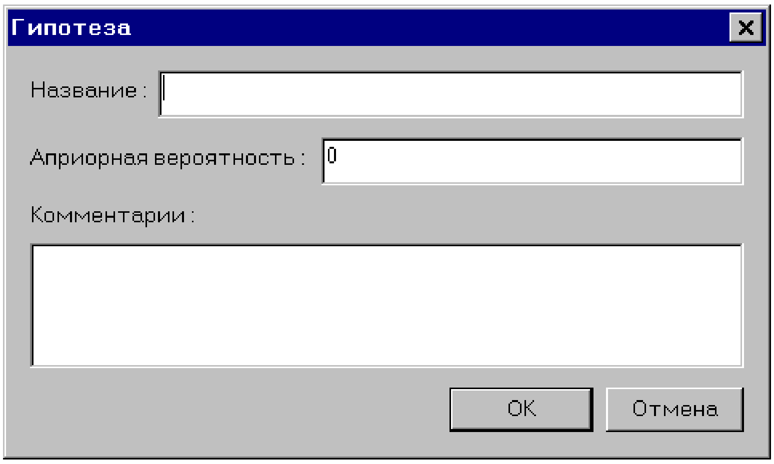


Рис. 3.60 Диалоговое окно редактирования гипотез

Правила допустимости вводимых значений свойств гипотез таковы:

Название гипотезы должно состоять из алфавитно-цифровых символов, символов пробела и подчеркивания;

Не должно быть двух и более гипотез с одинаковым названием;

Априорная вероятность должна быть вещественным числом, принимающим значения в диапазоне между 0 и 1.

При модификации и добавлении правил используется диалоговое окно, изображенное на рис. 3.61.

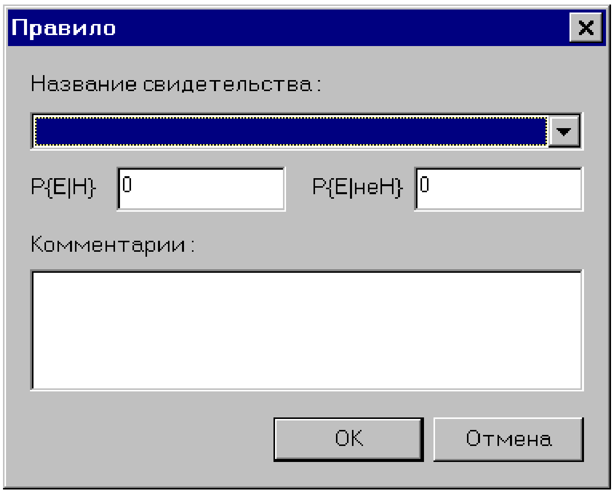


Рис. 3.61 Диалог редактирования правил

Способ работы с данным окном аналогичен способу работы с диалогами редактирования свидетельств и гипотез. Единственным отличием является наличие поля редактирования со встроенным списком (combo-box), которое используется для ввода названия свидетельства, соответствующего этому правилу (рис. 3.62). Для ввода свидетельства необходимо выбрать соответствующее название из списка, после чего оно отобразится в поле редактирования.

Диалоговое окно редактирования правил с активным списком свидетельств изображено ниже:

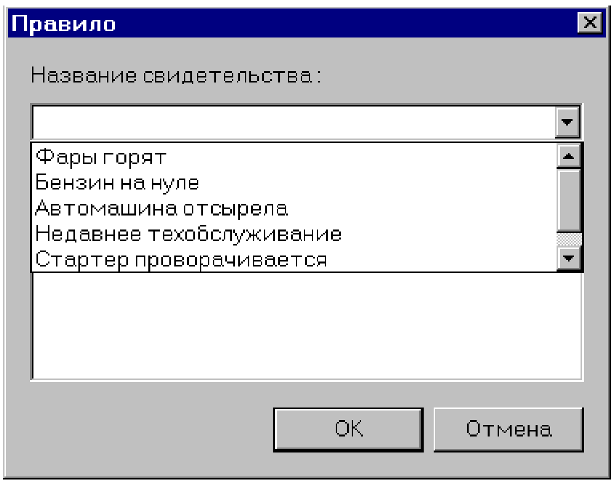


Рис. 3.62 Выбор названия свидетельства

Правило допустимости вводимых значений свойств правил:

P{E|H} и P{E|неH} должны быть вещественными числами, принимающими значения в диапазоне между 0 и 1.

Редактирование общей информации

Общая информация БЗ ЭС представлена в окне редактирования БЗ в виде таблицы (рис. 3.63), имеющей два столбца «Элемент» (название элемента общей информации БЗ) и «Значение» (значение этого элемента).

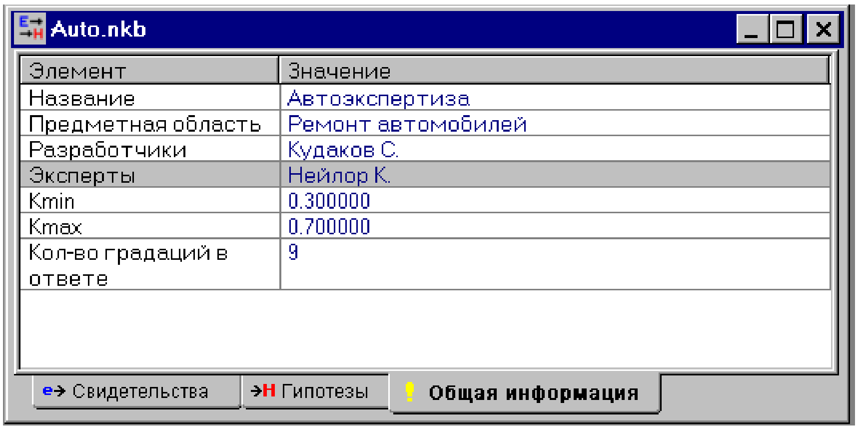


Рис. 3.63 Дочернее окно редактирования БЗ. Активна закладка "Общая информация"

Редактирование общей информации базы знаний осуществляется с использованием встроенного (built-in) редактирования. Элементы общей информации БЗ с указанием допустимых значений перечислены в таблице 3.5.

Таблица 3.5

Элемент общей информации	Описание	Допустимые значения
Название	Название ЭС	Любые символы
Предметная область	Краткое описание предметной области ЭС	То же
Разработчики	Перечисление разработчиков	То же
Эксперты	Перечисление экспертов или	То же

	библиографических источников, из которых была заимствована информация для БЗ	
Kmin	Kmin – коэффициент критерия нижнего порогового значения для принятия конкретной гипотезы (величина пропорциональная Pmin, вычисленной до начала работы ЭС, т.е. с учетом всех свидетельств). Kmin – коэффициент пропорциональности	Вещественное число в диапазоне между 0 и 1
Kmax	То же, но для Pmax	То же
Количество градаций в ответе	При ответе на вопросы ЭС пользователь должен вводить «коэффициент уверенности» в этом ответе. Описываемый элемент общей информации задает максимальное значение этого коэффициента	Целое положительное число

3.2.5 Работа готовой экспертной системы.

Работа пользователя с экспертной системой в конкретной предметной области, база знаний которой создана с помощью редактора баз знаний, осуществляется при помощи оболочки экспертной системы (рис. 3.45).

Оболочка экспертной системы позволяет загрузить базу знаний из файла и запустить процесс логического вывода. В процессе работы экспертная система, в соответствии с алгоритмом вывода, выбирает вопросы к пользователю, получает на них ответы и вычисляет вероятности гипотез. При нахождении решения система сообщает об этом пользователю. В случае если система приходит к выводу, что решение не может быть найдено, она также информирует об этом пользователя. Во время работы система отображает информацию о производимых действиях, вероятностях гипотез и ценах свидетельств в удобной для пользователя форме. Оболочка ЭС позволяет сохранить в файле текущее состояние системы и, впоследствии, восстановить это состояние, загрузив файл с диска. Также имеется возможность сохранять в файле отчет о работе ЭС для его последующего изучения.

Оболочка ЭС является приложением операционных систем Windows 95, Windows 98, Windows 2000, Windows NT v.3.5 и Windows NT v.4.0. Исполняемый файл оболочки ЭС называется ESShell.exe (Expert System Shell).

Главное окно

Главное окно приложения представлено на рис. 3.45. Оно содержит следующие элементы:

- Заголовок;
- Меню;
- Панель инструментов;
- Панель состояния;
- Панель гипотез;
- Панель свидетельств;
- Панель ответов;
- Окно отчета о работе ЭС.

Заголовок

Заголовок – это горизонтальная полоса в верхней части окна. Заголовок содержит следующие элементы:

Небольшая пиктограмма (иконка). При нажатии левой или правой кнопки манипулятора «мышь» над этой пиктограммой открывается системное меню (рис. 3.64).

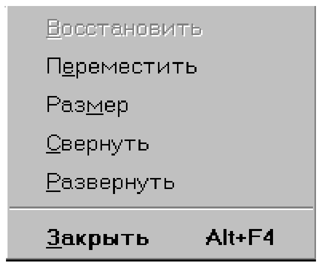


Рис. 3.64 **Системное меню**

Название программы («Оболочка экспертных систем»);

Имя файла, из которого была загружена БЗ (если БЗ не загружена этот элемент заголовка отсутствует);

Кнопки минимизации, максимизации и закрытия окна.

Система меню

Система меню представлена на рис. 3.65.

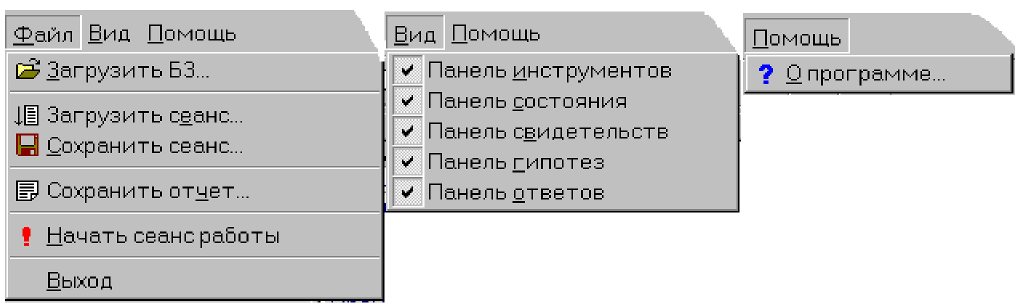


Рис. 3.65 Система меню

Рассмотрим назначение различных пунктов меню.

«Файл/Загрузить БЗ» – загрузить базу знаний из файла.

Более подробно процесс загрузки описан в пункте Загрузка БЗ;

«Файл/Загрузить сеанс» – выбор этого пункта позволяет загрузить сохраненный ранее сеанс работы пользователя с оболочкой ЭС.

В этом файле сохраняется имя файла БЗ, его тип, некоторая служебная информация, а также номера вопросов заданных системой пользователю и ответы на них. При выборе этого пункта система пытается загрузить БЗ, при этом имя файла БЗ не запрашивается, так как оно содержится в файле сохранения сеанса. После успешной загрузки система проделывает те же действия, которые пользователь уже выполнил во время данного сеанса. Все проделанные действия отображаются в окне отчета о работе ЭС. Имя файла запрашивается у пользователя, при этом используется стандартный диалог открытия файла.

«Файл/Сохранить сеанс» – сохраняет сеанс работы с ЭС.

Информация, заносимая в файл сохранения сеанса работы, приведена в предыдущем пункте. Впоследствии, можно будет продолжить работу с ЭС, загрузив данный сеанс. Имя файла запрашивается у пользователя, при этом используется стандартный диалог сохранения файла. Этот пункт доступен только в случае, когда база знаний загружена.

«Файл/Сохранить отчет» – сохраняет содержимое окна отчета о работе ЭС в текстовом файле.

Имя файла запрашивается у пользователя. При этом используется стандартный диалог сохранения файла. Этот пункт доступен только в случае, когда база знаний загружена.

«Файл/Начать сеанс работы» – после выбора этого пункта начинается работа пользователя с ЭС.

Система проводит необходимые вычисления, выбирает следующий вопрос, задает его пользователю и переходит в режим ожидания ответа. Все проделанные действия отображаются в окне отчета о работе ЭС. Этот пункт доступен только в случае, если база знаний загружена.

«Файл/Выход» – закончить работу с программой.

«Вид/Панель инструментов» – этот пункт меню работает как двоичный переключатель. Для того, чтобы панель инструментов стала видимой (невидимой) необходимо в левой части этого пункта меню установить (снять) небольшую «галочка». Это производится путём нажатия левой кнопки «мыши».

«Вид/Панель состояния» – аналогично пункту «Панель инструментов».

«Вид/Панель свидетельств» – аналогично пункту «Панель инструментов».

«Вид/Панель гипотез» – аналогично пункту «Панель инструментов».

«Вид/Панель ответов» – аналогично пункту «Панель инструментов».

«Помощь/О программе». При выборе этого пункта открывается диалоговое окно с информацией о названии программы, разработчике и номере версии (рис. 3.66).

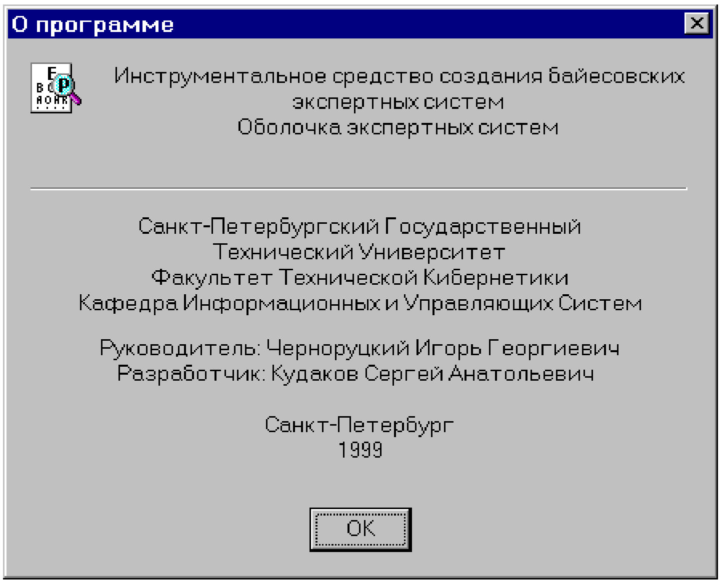


Рис. 3.66 Диалог "О программе"

Панель инструментов

Панель инструментов содержит ряд небольших кнопок с различными пиктограммами. Нажатие этих кнопок приводит к выполнению тех же действий, что и выбор соответствующих пунктов в меню.

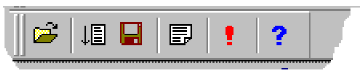


Рис. 3.67 Панель инструментов

Далее приведено соответствие пунктов меню и кнопок панели инструментов:

 – «Файл/Загрузить БЗ»;

 – «Файл/Загрузить сеанс»;

 – «Файл/Сохранить сеанс»;

 – «Файл/Сохранить отчет»;

 – «Файл/Начать сеанс работы»;

 – «Помощь/О программе».

Панель состояния

Панель состояния представлена на рис. 3.68. Она расположена в нижней части главного окна. Панель состояния отображает краткую информацию о состоянии приложения или описание действия соответствующего пункту меню, на котором находится указатель манипулятора «мышь».

Рис. 3.68 Панель состояния

В правой части панели состояния расположены три небольшие окна, содержащие информацию о состоянии клавиш «Num Lock», «Caps Lock» и «Scroll Lock».

Панель свидетельств

Данная панель содержит список свидетельств загруженной базы знаний и значения цен этих свидетельств (рис. 3.69). Список заполняется после загрузки БЗ, в процессе работы значения цен свидетельств обновляются (изначально они равны 0).

Рис. 3.69 Панель свидетельств

Панель гипотез

Эта панель содержит список гипотез, значения текущей, а также минимально и максимально возможной вероятностей для каждой гипотезы. Список заполняется после загрузки БЗ, в процессе работы значения вероятностей обновляются (изначально они равны 0).

Гипотеза	Ртекущ.	Рмин.	Рмакс.
Севший аккумулятор	0.000000	0.000000	0.000000
Нет бензина	0.000000	0.000000	0.000000
Отсырело зажигание	0.000000	0.000000	0.000000
Замаслены свечи	0.000000	0.000000	0.000000

Рис. 3.70 Панель гипотез

Панель ответов

Оболочка ЭС использует данную панель для того, чтобы задавать пользователю вопросы и получать на них ответы (рис. 3.71).

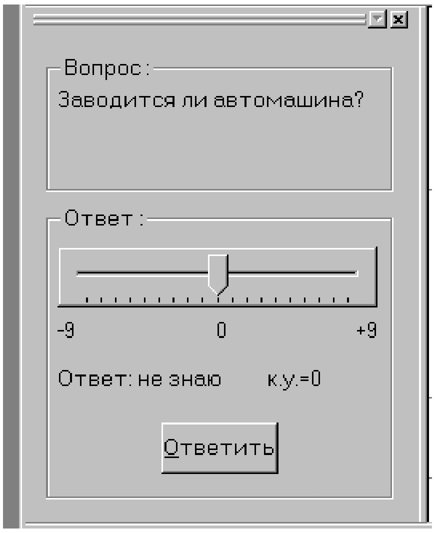


Рис. 3.71 Панель ответов

После выбора следующего вопроса, система выводит его текст в поле «Вопрос» (при этом кнопка «Ответить» становится доступной). Ответ вводится с помощью ползунка (slider), расположенного в поле «Ответ». На шкале ползунка указаны значения 0, максимальное утвердительное и максимальное отрицательное значение (они определяются значением элемента «Количество градаций в ответе» в общей информации загруженной БЗ). Ответ выбирается следующим образом: пользователь помещает ползунок в нужное положение и нажимает кнопку «Ответить». При передвижении ползунка изменяется содержимое полей «Ответ:» и «к.у.» (коэффициент уверенности).

Положение ползунка левее 0 означает ответ «Нет» с соответствующим коэффициентом уверенности, правее – «Да», 0 – пользователь не знает ответа на вопрос (на рис. 3.71 отражена именно эта ситуация).

Окно отчета о работе системы

В данном окне содержится список действий, произведенных системой, а также информация о ценах свидетельств и вероятностях гипотез (рис. 3.72).

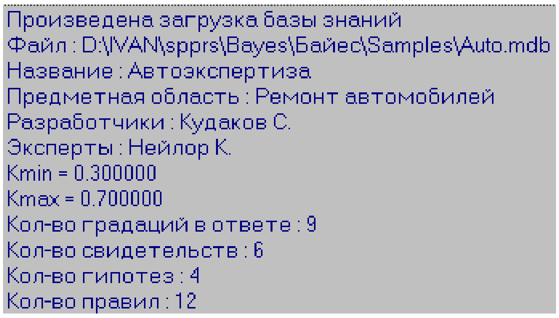


Рис. 3.72 Окно отчёта о работе системы

При сохранении отчета, содержимое этого окна копируется в текстовый файл. В случае, если системой было найдено решение или было обнаружено, что его невозможно найти, информация об этом (причина завершения и название гипотезы) в окне отчета будет также продублирована в информационном окне (рис. 3.73).

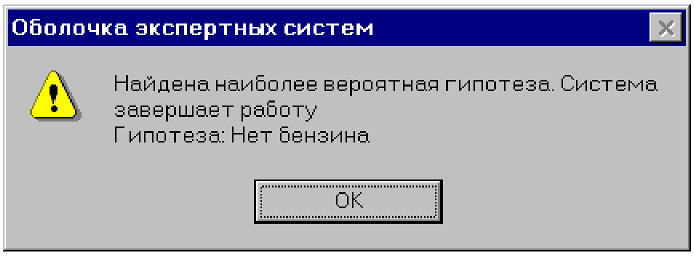


Рис. 3.73 Информационное окно с сообщением о завершении работы

Установка разработанной байесовской экспертной системы на ПК пользователя

Для установки ЭС необходимо создать комплект инсталляционных дискет. Для этого необходимо скопировать содержимое директориев *X:\USER_PATH\INSTALL\DISK1*, *X:\USER_PATH\INSTALL\DISK2* и *X:\USER_PATH\INSTALL\DISK3* на дискеты, а также подготовить дискету с БЗ разработанной экспертной системы.

Для этого скопируйте на эту дискету файл базы знаний ЭС, создайте на ней файл под названием kb.inf, и отредактируйте его с помощью любого текстового редактора следующим образом :

первая строчка – имя файла БЗ (например, auto.mdb);

вторая строчка – тип файла:

«mdb» – если БЗ сохранена в формате *.mdb (файлы баз данных СУБД MS Access);

«nkb» – если БЗ сохранена в формате *.nkb (текстовые файлы с описанием БЗ на языке представления знаний);

третья строчка – название экспертной системы (например, «Экспертная система для диагностики неисправности автомобиля»).

Ниже приведен пример такого файла (kb.inf):

auto.mdb *mdb* *Экспертная система для диагностики неисправности автомобиля*

Для инсталляции экспертной системы пользователь должен запустить файл setup.exe с дискеты 1 и далее следовать инструкциям программы установки. В результате инсталляции будут установлены файлы экспертной системы, а также в меню «Пуск\Программы» будет добавлен пункт «Байесовские экспертные системы\<Название экспертной системы>». При выборе этого пункта будет запущена *Оболочка экспертной системы* и загружена соответствующая база знаний.

3.2.6 Пример решения модельной задачи.

Рассмотрим пример решения конкретной задачи – создание байесовской экспертной системы в области ремонта автомобилей с помощью разработанного инструментального средства. Необходимо отметить, что данная экспертная система будет очень простой с небольшой по размерам БЗ, однако, этот пример позволит изучить различные стадии процесса создания экспертных систем с помощью разработанного средства.

1 этап. Выделим возможные причины неисправности автомобиля (гипотезы). В данном случае это будут:

Севший аккумулятор;

Нет бензина;

Отсырело зажигание;

Замаслены свечи.

2 этап. Зададим, значения априорных вероятностей гипотез.

3 этап. Далее определим критерии, которые помогут нам определить причину неисправности (свидетельства):

Фары горят;

Бензин на нуле;

Автомашина отсырела;

Недавнее техобслуживание;

Стартер проворачивается;

Автомашина не заводится.

4 этап. Определим связи (зависимости) между гипотезами и свидетельствами, укажем значения условных вероятностей:

P{E|H} – вероятность того, что свидетельство E имеет место при условии, что гипотеза H верна;

P{E|неH} – вероятность того, что свидетельство E имеет место при условии, что гипотеза H ложна.

Например, ясно, что если в баке автомобиля нет бензина, то прибор, отображающий уровень бензина будет показывать на нулевое значение, с максимальным значением вероятности. Иначе говоря, если гипотеза «Нет бензина» верна, то свидетельство «Бензин на нуле» имеет место с вероятностью 1.0. При этом если гипотеза «Нет бензина» ложна, то свидетельство «Бензин на нуле» может иметь место (например, если прибор уровня бензина неисправен), но с крайне небольшой вероятностью, скажем 0.01. Итак, мы задали одно из правил БЗ. Аналогично можно задать остальные.

Приведем БЗ целиком. Она будет записана в следующем виде:

Описания свидетельств – нумерованный список пар <*Название свидетельства*> <*Текст вопроса пользователю*>;

Описание гипотез – <*Название гипотезы*>, значение априорной вероятности гипотезы, список правил, относящихся к данной гипотезе. В описании каждого правила указывается: название свидетельства, с которым связана данная гипотеза этим правилом, значения условных вероятностей P{E|H} и P{E|неH}. Следует отметить, что кроме названия свидетельства, указывается также его номер.

<u>Свидетельства</u>

1) Фары горят

Вопрос: Горят ли фары?

2) Бензин на нуле

Вопрос: Не показывает ли прибор уровня бензина слишком малое значение?

3) Автомашина отсырела

Вопрос: Не стояла ли автомашина долго под дождем?

4) Недавнее техобслуживание

Вопрос: Проходила ли недавно эта машина техобслуживание?

5) Стартер проворачивается

Вопрос: Проворачивается ли стартер?

6) Автомашина не заводится

Вопрос: Заводится ли автомашина?

<u>Гипотезы</u>

1) Севший аккумулятор

Априорная вероятность=0.1

Правила:

Фары горят(1) P{E|H}=0 P{E|неH}=0.99

Бензин на нуле(2) P{E|H}=0.7 P{E|неH}=0.05

Недавнее техобслуживание(4) P{E|H}=0.2 P{E|неH}=0.5

Стартер проворачивается(5) P{E|H}=0 P{E|неH}=0.99

Автомашина не заводится(6) P{E|H}=1 P{E|неH}=0.01

2) Нет бензина

Априорная вероятность=0.05

Правила:

Бензин на нуле(2) P{E|H}=1 P{E|неH}=0.01

Автомашина не заводится(6) P{E|H}=0.9 P{E|неH}=0.02

3) Отсырело зажигание

Априорная вероятность=0.01

Правила:

Автомашина отсырела(3) P{E|H}=0.9 P{E|неH}=0.1

Недавнее техобслуживание(4) P{E|H}=0.25 P{E|неH}=0.5

Автомашина не заводится(6) P{E|H}=0.9 P{E|неH}=0.02

4) Замаслены свечи

Априорная вероятность=0.01

Правила:

Недавнее техобслуживание(4) P{E|H}=0.01 P{E|неH}=0.5

Автомашина не заводится(6) P{E|H}=0.9 P{E|неH}=0.02

Далее необходимо создать БЗ с помощью редактора, имеющегося в разработанном средстве, отредактировать ее и сохранить в одном из поддерживаемых форматов. Для этого необходимо установить «Инструментальное средство создания байесовских ЭС» на Ваш ПК (см. раздел Инсталляция системы). Затем необходимо выполнить следующие действия:

Запустить редактор БЗ;

Создать новую БЗ;

Ввести свидетельства, гипотезы, правила и общую информацию;

Сохранить БЗ в одном из поддерживаемых форматов, например «*.nkb».

Подробное описание процесса создания баз знаний приводиться в разделе Система в конкретной предметной области. Ниже приведен фрагмент файла БЗ формата «*.nkb» для разрабатываемой ЭС:

```
// File C:\DevStud\MyProjects\KBEdit\Auto.nkb - knowledge base of expert
system
// Created with Knowledge Base Editor v 2.0

// Общая информация
GeneralInfo
{
	Title : "Автоэкспертиза";
	Sphere : "Ремонт автомобилей";
	Developers : "Кудаков С.";
	Experts : "Нейлор К.";
	Kmin : 0.300000;
	Kmax : 0.700000;
	ResponsePrecision : 11;
}

// Описание свидетельств
Evidences
{
	Evidence
	{
		Name : "Фары горят";
		Comment : "";
		Question : "Горят ли фары";
	}
	Evidence
	{
		Name : "Бензин на нуле";
		Comment : "";
		Question : "Не показывает ли прибор уровня бензина слишком
малое значение";
	}
	Evidence
	{
		Name : "Автомашина отсырела";
		Comment : "";
		Question : "Не стояла ли автомашина долго под дождем";
	}
```

```
        Evidence
        {
                Name : "Недавнее техобслуживание";
                Comment : "";
                Question : "Проходила ли недавно эта машина
техобслуживание";
        }
        Evidence
        {
                Name : "Стартер проворачивается";
                Comment : "";
                Question : "Проворачивается ли стартер";
        }
        Evidence
        {
                Name : "Автомашина не заводится";
                Comment : "";
                Question : "Заводится ли автомашина";
        }
}

// Описание гипотез
Hypotheses
{
        Hypothes
        {
                Name : "Севший аккумулятор";
                Comment : "";
                PriorProb : 0.100000;
                Rules
                {
                        Rule
                        {
                                EvidenceName : "Фары горят";
                                Comment : "";
                                P+ : 0.000000;
                                P- : 0.990000;
                        }
                        Rule
                        {
                                EvidenceName : "Бензин на нуле";
```

```
                                Comment : "";
                                P+ : 0.700000;
                                P- : 0.050000;
                        }
                        Rule
                        {
                                EvidenceName : "Недавнее техобслуживание";
                                Comment : "";
                                P+ : 0.200000;
                                P- : 0.500000;
                        }
                        Rule
                        {
                                EvidenceName : "Стартер проворачивается";
                                Comment : "";
                                P+ : 0.000000;
                                P- : 0.990000;
                        }
                        Rule
                        {
                                EvidenceName : "Автомашина не заводится";
                                Comment : "";
                                P+ : 1.000000;
                                P- : 0.010000;
                        }
                }
        }
  . . . . .
  . . . . .
  . . . . .
}
```

5 этап. Проверка корректности работы ЭС.

Необходимо запустить оболочку ЭС и загрузить созданный с помощью редактора БЗ файл базы знаний. Затем необходимо начать сеанс работы.

Экспертная система выполнит некоторые инициализирующие действия и задаст пользователю первый вопрос – «Заводится ли автомашина?». Далее пользователь с помощью элементов управления панели ответов вводит свой ответ. И так далее.

Ниже приводится содержимое файла отчета о работе ЭС. Система сделала вывод, что наиболее вероятной является гипотеза «Нет бензина», получив от пользователя информацию о том, что «Автомашина не заводится» и «Бензин на нуле» (т. е. датчик уровня бензина показывает нулевое значение). Такой вывод кажется вполне разумным и логичным.

```
//отчет о работе с ЭС. Создан оболочкой нейлоровских экспертных систем
версии 2.0

Произведена загрузка базы знаний
Файл : C:\DevStud\MyProjects\KBEdit\Auto.nkb
Название : Автоэкспертиза
Предметная область : Ремонт автомобилей
Разработчики : Кудаков С.
Эксперты : Нейлор К.
Kmin = 0.300000
Kmax = 0.700000
Кол-во градаций в ответе : 11
Кол-во свидетельств : 6
Кол-во гипотез : 4
Кол-во правил : 12

1) Вычислены нижние и верхние пороговые значения вероятностей гипотез
Гипотеза___________      Pmin_______ Pmax
Севший аккумулятор_      0.000000___ 0.700000
Нет бензина________      0.000000___ 0.697057
Отсырело зажигание_      0.000017___ 0.601911
Замаслены свечи____      0.000006___ 0.331579

2) Вычислены цены свидетельств
Свидетельство____________     Цена
Фары горят_______________     0.917431
Бензин на нуле___________     1.415133
Автомашина отсырела______     0.082212
Недавнее техобслуживание_     0.137696
```

```
Стартер проворачивается__     0.917431
Автомашина не заводится__     2.238155

3) Выбран следующий вопрос
Соответствующее свидетельство : Автомашина не заводится

4) Пользователю задан следующий вопрос
Вопрос : Заводится ли автомашина?

5) Получен ответ на вопрос системы
Ответ : Нет Вероятность : 1.000000

6) Вычислены вероятности гипотез
Гипотеза___________     Ртекущ.____ P(min)_____ P(max)
Севший аккумулятор_     0.000000-__ 0.000000=__ 0.000000-
Нет бензина________     0.005342-__ 0.000000=__ 0.349406-
Отсырело зажигание_     0.001030-__ 0.000057+__ 0.013724-
Замаслены свечи____     0.001030-__ 0.000021+__ 0.002037-

7) Произведен анализ состояния системы
Решение пока не найдено. Система продолжает работу.

8) Вычислены цены свидетельств
Свидетельство____________     Цена
Фары горят_______________     0.000000
Бензин на нуле___________     0.349406
Автомашина отсырела______     0.009077
Недавнее техобслуживание_     0.003045
Стартер проворачивается__     0.000000
Автомашина не заводится__     0.000000

9) Выбран следующий вопрос
Соответствующее свидетельство: Бензин на нуле

10) Пользователю задан следующий вопрос
Вопрос : Не показывает ли прибор уровня бензина слишком малое значение?

11) Получен ответ на вопрос системы
Ответ : Да Вероятность : 1.000000

12) Вычислены вероятности гипотез
```

```
Гипотеза___________       Ртекущ.____ P(min)_____ P(max)
Севший аккумулятор_       0.000000=__ 0.000000=__ 0.000000=
Нет бензина________       0.349406+__ 0.349406+__ 0.349406=
Отсырело зажигание_       0.001030=__ 0.000057=__ 0.013724=
Замаслены свечи____       0.001030=__ 0.000021=__ 0.002037=

13) Произведен анализ состояния системы
Найдена наиболее вероятная гипотеза. Система завершает работу
Гипотеза : Нет бензина
```

6 этап. В заключении необходимо подготовить комплект инсталляционных дискет, необходимых для установки созданной экспертной системы на ПК конечного пользователя. Этот процесс подробно описан в разделе

Установка разработанной байесовской экспертной системы на ПК пользователя

3.2.7 Заключение.

Данный программный продукт является инструментальным средством создания байесовских экспертных систем. Он может быть применен для создания экспертных систем в любых областях человеческой деятельности, где возникает ситуация наличия ненадежных данных. Для работы с ненадежными данными был реализован подход, основанный на теории Байеса. Система не требует от пользователя специальных знаний в области искусственного интеллекта и теории экспертных систем. Все необходимые ему сведения содержаться в пользовательской документации, поставляемой вместе с программным средством. Система имеет удобный пользовательский интерфейс, систему установки (инсталляции) на ПК пользователя, поддерживает различные форматы файлов баз знаний, позволяет создавать и просматривать отчет о работе экспертной системы.

4 БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Кини Р.Л., Райфа Х. Принятие решений при многих критериях: предпочтения и замещения/Под ред. И.Ф. Шахнова. – М.: Радио и связь, 1981.
2. Вентцель Е.С. Исследование операций. – М.: Советское радио, 1972.
3. Борисов А.Н., Вилюмс Э.Р., Сукур Л.Я. Диалоговые системы принятия решений на базе МИНИ-ЭВМ: Информационное, математическое и программное обеспечение. – Рига: Зинатне, 1986.
4. Вязгин В.А., Федоров В.В. Математические методы автоматизированного проектирования: Учеб. пособие для втузов. – М.: Высш. шк., 1989.
5. Моисеев Н.Н. Математические задачи системного анализа. – М.: Наука, 1981.
6. Растригин Л.А. Современные принципы управления сложными объектами. – М.: Сов. радио, 1980.
7. Ракитский Ю.В., Устинов С.М., Черноруцкий И.Г. Численные методы решения жестких систем. – М.: Наука, 1979.
8. Перегудов Ф.И., Тарасенко Ф.П. Введение в системный анализ. – М.: Высш. шк., 1989.
9. Подиновский В.В., Ногин В.Д. Парето–оптимальные решения многокритериальных задач. – М.: Наука, 1982.

10. Экспертные системы. Принципы работы и примеры/ А. Брукинг, П.Джонс, Ф. Кокс и др.; Под. ред. Р. Форсайта. – М.: Радио и связь, 1987.
11. Розен В.В. Цель – оптимальность – решение. – М.: Радио и связь, 1982.
12. Черноруцкий И.Г. Методы принятия решений: Учеб. пособие. – Л.: Изд-во Ленингр. Политехн. Ин-та, 1990.
13. Федоренко Р.П. Приближенное решение задач оптимального управления. – М.: Наука, 1978.
14. Табак Д., Куо Б. Оптимальное управление и математическое программирование. – М.: Наука, 1975.
15. Таха Х. Введение в исследование операций: В 2-х книгах. Кн. 2. – М.: Мир, 1985.
16. Черноруцкий И.Г. Оптимальный параметрический синтез: электротехнические устройства и системы. – Л.: Энергоатомиздат, 1987.
17. Льюс Р.Д., Райфа Х. Игры и решения. М.: Изд-во иностр. лит-ры, 1961.
18. Ногин В.Д., Чистяков С.В. Применение линейной алгебры в принятии решений: Учеб. пособие. - СПб.: Изд-во СПбГТУ, 1998.
19. Поспелов Г.С. Искусственный интеллект – основа новой информационной технологии. – М.: Наука, 1988.
20. Черноруцкий И.Г. Методы оптимизации: Учеб. пособие. – СПб., Изд-во СПбГТУ, 1998.
21. Миркин Б.Г. Проблема группового выбора. М.: Наука, 1974.
22. Нейлор К. Как построить свою экспертную систему. М.: Энергоатомиздат, 1991.

23. Дэннис Дж., Шнабель Р. Численные методы безусловной оптимизации и решения нелинейных уравнений. – М.: Мир, 1988.
24. Дубов Ю.А., Травкин С.И., Якимец В.Н. Многокритериальные модели формирования и выбора вариантов систем. – М.: Наука, 1986.
25. Попов Э.В. Экспертные системы: Решение неформализованных задач в диалоге с ЭВМ. – М.: Наука, 1987.
26. Построение экспертных систем: Пер. с англ./ Под ред. Ф. Хейеса-Рота, Д. Уотермана, Д. Лената. – М.: Мир, 1987.
27. Хованов Н.В. Анализ и синтез показателей при информационном дефиците. – СПб.: Изд-во СпбГУ, 1996.
28. Глухов В.В., Медников М.Д., Коробко С.Б. Математические методы и модели для менеджмента. СПб.: Изд-во «Лань», 2000.
29. Змитрович А.И. Интеллектуальные информационные системы. Мн: НТООО «ТетраСистемс», 1997.
30. Малыхин В.И. Финансовая математика: Учеб. пособие для вузов. – М.: ЮНИТИ-ДАНА, 2000.
31. Уотшем Т.Дж., Паррамоу К. Количественные методы в финансах: Учеб. пособие для вузов. – М.: Финансы, ЮНИТИ, 1999.
32. Интеллектуальные системы принятия проектных решений/А.В.Алексеев, А.Н.Борисов, Э.Р.Вилюмс, Н.Н.Слядзь, С.А.Фомин. – Рига: Зинатне, 1997.
33. Шрейдер Ю.А. Равенство, сходство, порядок. – М.: Наука, 1971.
34. Химмельблау Д. Прикладное нелинейное программирование. – М.: Мир, 1975.
35. Подиновский В.В. Многокритериальные задачи с упорядоченными по важности однородными

критериями//Автоматика и Телемеханика, 1976. - № 11. – С. 118 – 127.