

**Герасимов И.М., Кошечев В.Д.**

**ОСНОВЫ  
АВТОМАТИЗИРОВАННОГО  
ПРОЕКТИРОВАНИЯ**

*Рекомендовано в качестве учебного пособия Учебно-методической комиссией  
по специальности 150100 — Автомобиле- и тракторостроение  
(решение №38 от 19.05.2004 г.)*

Санкт-Петербург  
2006

ББК 30.2-5-05  
УДК 621.398.001.2  
Г37

**Герасимов И.М., Кощев В.Д.**

Основы автоматизированного проектирования. Учебное пособие.

С-Петербург, ООО «Творческий Центр «Победа», 2006 г. — 68 с., 24 илл.

Г37

В пособии изложены основные понятия и подходы к проектированию, освещены вопросы технического, математического, программного и информационного обеспечения САПР, основы математического моделирования, принципы постановки задачи оптимизации и некоторые способы ее решения.

Для студентов, аспирантов, инженеров, решающих задачи автоматизированного проектирования.

**ISBN 5-221-00014-8**

# 1. ОБЩИЕ ОСНОВЫ ПРОЕКТИРОВАНИЯ

## 1.1. ОСНОВНЫЕ ПОНЯТИЯ

Прогресс науки и техники неизбежно приводит к появлению все более сложных технических объектов, состоящих из большого количества взаимосвязанных элементов. Важной стадией при создании новых технических объектов является *проектирование* — процесс составления описания, необходимого для создания в заданных условиях еще не существующего объекта, на основе первичного описания этого объекта и (или) алгоритма его функционирования. Проектирование сложных объектов занимает длительное время и требует привлечения значительного числа специалистов. Такие сроки часто неприемлемы, так как оказываются соизмеримыми со временем морального старения объекта проектирования. Поэтому две противоречивые тенденции — усложнение технических объектов и сокращение сроков проектирования делают автоматизацию проектирования сложных систем необходимостью.

Различают восемь стадий создания нового объекта:

- предпроектные исследования;
- техническое задание;
- техническое предложение;
- эскизный проект;
- технический проект;
- рабочий проект;
- изготовление;
- отладка, испытания.

Проектирование начинается при наличии первичного описания — *технического задания (ТЗ)*, в котором в общем виде сформулированы назначение будущего объекта и требования к его свойствам. В результате выполнения процесса проектирования (определенных операций и процедур) составляется окончательное описание — *конструкторская (проектная) документация*, несущая в себе всю необходимую информацию для создания (изготовления) в определенных условиях производства технического объекта и его эксплуатации.

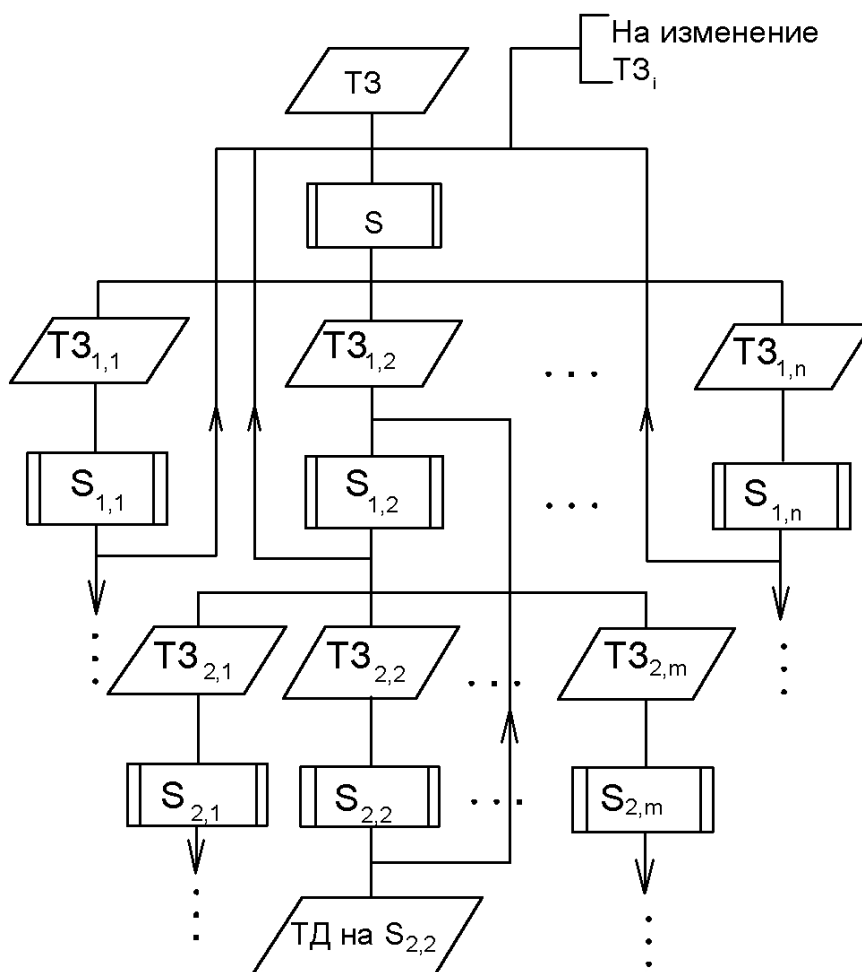


Рис. 1. Иерархические уровни проектирования

При проектировании сложных технических объектов используется блочно-иерархический подход, когда представления о техническом объекте расчленяются на иерархические уровни. На высшем уровне используется

наименее детализированное представление, отражающее только самые общие черты и особенности проектируемой системы. На следующих уровнях степень подробности рассмотрения возрастает, при этом рассматривается уже не система в целом, а отдельные блоки. Таким образом общая сложная задача проектирования разбивается на ряд более простых задач, последние также разбиваются и т.д. Такой подход позволяет на каждом уровне формулировать и решать задачи приемлемой сложности, поддающиеся уяснению человеком и решению с помощью имеющихся средств проектирования.

При переходе с некоторого  $i$ -го уровня (рис.1), на котором рассматривается система  $S_i$  на соседний, более низкий уровень  $i+1$ , происходит разбиение системы  $S_i$  на блоки,  $S_{i+1,1}$ ,  $S_{i+1,2}$ , ... ,  $S_{i+1,m}$  с фиксацией параметров связей между ними. Это позволяет обеспечить параллельность выполнения работ по проектированию с большой глубиной проработки отдельных блоков. Такое проектирование называют **разделённым**. Недостатками разделённого проектирования являются зависимость качества проектного решения в целом от разработки технического задания для последующего уровня и большого количества связей между уровнями и блоками, что увеличивает затраты времени на согласование вопросов при внесении изменений. Кроме того, при фиксации параметров связей между блоками часть вариантов проектных решений на основе неформализованных соображений исключается из рассмотрения. При этом может быть исключен из рассмотрения и наилучший вариант.

Этих недостатков практически лишено **совместное** проектирование, выполняемое относительно немногочисленным коллективом высококвалифицированных специалистов, как правило, широко использующих в своей работе системы автоматизированного проектирования. Однако при разработке сложных объектов с проработкой конструкций отдельных узлов и агрегатов совместное проектирование в настоящее время не практикуется.

Кроме того, проектирование разделяют на нисходящее и восходящее. **Нисходящее проектирование** (проектирование «сверху вниз») характеризуется последовательной детализацией описания технического объекта от общего к частному. **Восходящее проектирование** (проектирование «снизу вверх») — проектирование, при котором вначале разрабатываются или выби-

раются элементы, а затем формируется система из этих элементов. При практическом проектировании сложных систем обычно имеют место черты как нисходящего, так и восходящего проектирования.

Рассмотрим основные задачи, решаемые на различных иерархических уровнях. При этом будем предполагать, что основная последовательность решения задач нисходящая.

В проектировании выделяют два класса задач: синтеза и анализа.

**Задачи синтеза** связаны с получением проектных вариантов, а **задачи анализа** — с их оценкой.

Задачи синтеза в свою очередь делятся на синтез структуры и синтез параметров. *Цель синтеза структуры* — получение структуры (схемы) объекта, то есть состава элементов и способа их связи между собой. Например, применительно к проектированию коробок передач транспортных машин — это задача составления возможных кинематических схем, обеспечивающих заданные передаточные числа. *Цель синтеза параметров* — определение параметров элементов заданной структуры. Если ставится задача определения наилучших в некотором смысле структуры и (или) значений параметров, то такая задача называется **задачей оптимизации**.

Задача синтеза структуры с трудом поддается формализации для решения на ЭВМ. Наиболее трудно автоматизируются задачи синтеза структуры с её последующим анализом для выбора оптимального варианта. Тем не менее, для некоторых видов технических объектов решение этой задачи частично или полностью автоматизировано. Например, на кафедре «Колёсные и гусеничные машины» Санкт-Петербургского государственного политехнического университета решена задача автоматизации синтеза кинематических схем планетарных коробок передач.

Но на нынешнем этапе развития средств автоматизированного проектирования, как правило, ставится задача определения оптимальных значений параметров элементов при заданной структуре объекта. Чтобы подчеркнуть такой характер оптимизации, ее называют **параметрической оптимизацией**.

Кроме параметрической оптимизации при проектировании может иметь место и **задача оптимального управления** — определение функции (закона)

изменения параметра, управляющего процессом функционирования устройства или системы, обеспечивающей наилучшие характеристики. Например, определение оптимального закона нарастания давления масла в гидроприводе включения фрикциона для обеспечения минимального износа рабочих поверхностей дисков; определение закона управления включением тормозов с целью исключения проскальзывания колес автомобиля и обеспечения наибольшей эффективности торможения на скользких участках и так далее.

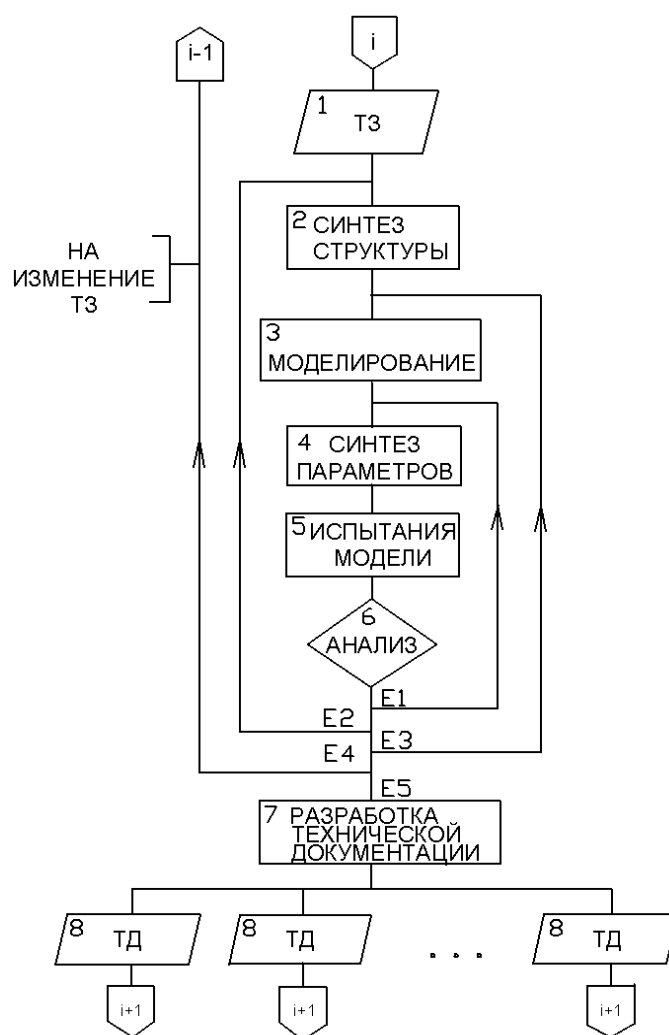


Рис. 2. Схема нисходящего проектирования

Задачи анализа при проектировании являются задачами исследования модели проектируемого объекта. В САПР — это исследование (испытание) *математической модели* — совокупности математических объектов (чи-

сел, переменных, векторов, множеств и т.п.) и отношений между ними, которая адекватно (правдиво) отображает свойства проектируемого объекта, интересующие инженера-проектировщика.

Пример типичной схемы нисходящего проектирования для одного уровня представлен на рис. 2.

При нисходящем проектировании система разрабатывается в условиях отсутствия полной информации, когда еще не спроектированы все ее элементы. При этом могут быть приняты не реализуемые по технологическим, экономическим или иным соображениям параметры системы и ее отдельных элементов, то есть может быть предъявлено невыполнимое ТЗ при переходе с одного уровня на более низкий. Ошибки выявляются при выполнении проектирования на нижнем уровне. Их исправление осуществляется путем возвращения на предыдущие уровни и повторное выполнение этапов (имеют место обратные связи). Отсюда вытекает важная особенность процесса проектирования, заключающаяся в его **итерационном характере** — последовательном приближении при составлении описания с целью получения наилучшего варианта технического объекта. Здесь важно отметить, что получение оптимальной конструкции, или просто конструкции, в полной мере удовлетворяющей заданным требованиям «с первого раза» как правило, не удаётся даже опытным конструкторам.

## 1.2. ИСПОЛЬЗОВАНИЕ ЭВМ В ПРОЦЕССЕ ПРОЕКТИРОВАНИЯ

Проектирование, при котором отдельные преобразования описания объекта и (или) алгоритма его функционирования осуществляется взаимодействием человека с ЭВМ, называется автоматизированным проектированием.

*Система автоматизированного проектирования (САПР) — организационно-техническая система, состоящая из комплекса средств, взаимосвязанного с коллективом специалистов (пользователей системы) и выполняющая автоматизированное проектирование.*

Иногда термином САПР обозначают программные комплексы, выполняющие функции *CAD (Computer Aided Design — система автоматизиро-*



ванного проектирования) и *CAE (Computer Aided Engineering — автоматизированная система определения характеристик проектируемого объекта)*. Но правильнее такие программы называть подсистемами САПР.

Процесс преобразования технического задания в проектную документацию, осуществляемый без участия человека, называется **автоматическим проектированием**. В этом случае за человеком остаются разработка (составление) технического задания (ТЗ) и анализ получаемого проектного решения с последующей корректировкой ТЗ, если оно неудовлетворительно. Реализация автоматического проектирования представляет собой трудновыполнимую задачу и возможна в настоящее время применительно лишь к простым техническим объектам, например, на основе выбора варианта из типового ряда.

Автоматизация проектирования характеризуется применением ЭВМ для выполнения проектных операций и процедур при рациональном использовании творческих и интеллектуальных способностей человека и возможностей вычислительной техники. В настоящее время при проектировании различных объектов автоматизируются следующие операции:

- расчёты, начиная от относительно несложных и кончая расчётными исследованиями математических моделей спроектированных объектов;
- создание трёхмерных параметрических моделей проектируемых объектов с экспресс-анализом их массовых характеристик, возможностей сборки, проверкой кинематических зависимостей;
- оформление расчётно-конструкторской документации;
- поиск справочных данных;
- взаимодействие между участниками проекта посредством телекоммуникационных систем.

Первое применение ЭВМ было направлено на автоматизацию рутинных расчетных операций. В таком режиме ЭВМ нередко используются и сейчас. По заданным «вручную» входным параметрам проводятся поверочные расчеты — определяются выходные характеристики технического объекта. На основе неформализованного анализа полученных выходных характеристик проводится корректировка входных данных, затем снова вводятся в ЭВМ и так

до тех пор, пока не будут найдены приемлемые значения параметров технического объекта. Это соответствует автоматизации этапа 5 «Испытание модели» (рис. 2) и относится к *первому уровню* автоматизации.

К первому уровню автоматизации относится и использование ЭВМ для рутинных работ этапа 7 «Разработка технической документации». Осуществляется это с помощью «лёгких» графических программ (например AutoCAD, Компас) и текстовых редакторов. Автоматизация этого этапа существенно сокращает время на разработку проекта. Анализ работы проектных организаций показывает, что при традиционном «ручном» проектировании до 50% объёма работ связано с разработкой, изготовлением и сопровождением чертежной документации. Автоматизация поиска, записи и хранения справочных данных, данных о предыдущих проектных решениях в виде электронных баз данных и архивов также относится к первому уровню автоматизации. Необходимо отметить, что подобное использование ЭВМ не даёт такого эффекта, как использование современных «средних» и «тяжёлых» программ проектирования, позволяющих автоматически создавать чертежи и спецификации разрабатываемых объектов либо их узлов.

Итак, на первом уровне автоматизировано выполнение отдельных процедур проектирования на каждом отдельном иерархическом уровне без автоматизации связей между ними.

На *втором уровне* автоматизации на ЭВМ перекладываются задачи анализа и синтеза параметров и структуры. В результате анализа может быть несколько исходов (см. рис. 2): E1 — переход к поиску лучшего сочетания значений параметров объекта проектирования; E2 — переход к поиску лучшей структуры; E3 — переход на уточнение математической модели; E4 — переход к предыдущему иерархическому уровню проектирования с целью корректировки технического задания из-за невозможности его выполнения или получения дополнительной информации; E5 — переход к разработке технической документации для найденного наилучшего в определенном смысле проектного решения. Автоматизация этапов 4, 5 и 6 с переходом E1 является оптимизацией параметров объектов проектирования на ЭВМ, а автоматизация этапов 2, 4, 5 и 6 с переходами E1 и E2 — оптимизация структу-

ры и параметров. Ниже рассмотрены некоторые методы и способы автоматизации этих проектных процедур. Одним из достоинств автоматизации второго уровня является исключение пропуска наилучшего варианта, что вполне вероятно может иметь место при традиционном «ручном» проектировании.

**Третий уровень** автоматизации характеризуется автоматизацией связей между отдельными этапами. Так, например, с помощью графических языков программирования возможна автоматическая передача полученного проектного решения (данных о структуре и параметрах) на изготовление чертежей без участия человека. Таким образом, на этом уровне осуществляется интеграция процесса проектирования. Дальнейшая интеграция процесса проектирования в САПР заключается в объединении на данном иерархическом уровне процессов проектирования отдельных блоков  $S_{i,1}, S_{i,2}, \dots, S_{i,n}$  (см. рис. 1), на которые разбивается система при «ручном» проектировании. При такой интеграции нет необходимости в фиксации связей между блоками, что сокращает время на проведение согласования параметров связей и исключает потерю наилучшего варианта проектного решения. Примером такой интеграции может служить автоматизированный выбор параметров зубчатых зацеплений одновременно для всей трансмиссии. Совместная оптимизация параметров зубчатых зацеплений, подшипников сателлитов и других элементов коробки передач или всей трансмиссии является дальнейшей интеграцией процесса проектирования. Объединение в единую САПР соседних иерархических уровней позволяет быстро согласовать изменения в техническом задании и проводить корректировку проектного решения и выпуск технической документации в короткие сроки.

## **2. СОСТАВ САПР**

Составными функциональными частями САПР являются техническое, математическое, программное, информационное, лингвистическое, организационное и методическое обеспечение.

### **2.1. ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САПР**

Одним из основных компонентов технического обеспечения САПР является автоматизированное рабочее место (АРМ). Центральным элементом в комплексе АРМ является вычислительная система, связанная с устройствами ввода-вывода. Кроме того, в состав технического обеспечения САПР могут входить аппаратура передачи данных между удаленными техническими средствами, измерительные приборы и устройства, устройства организационной техники.

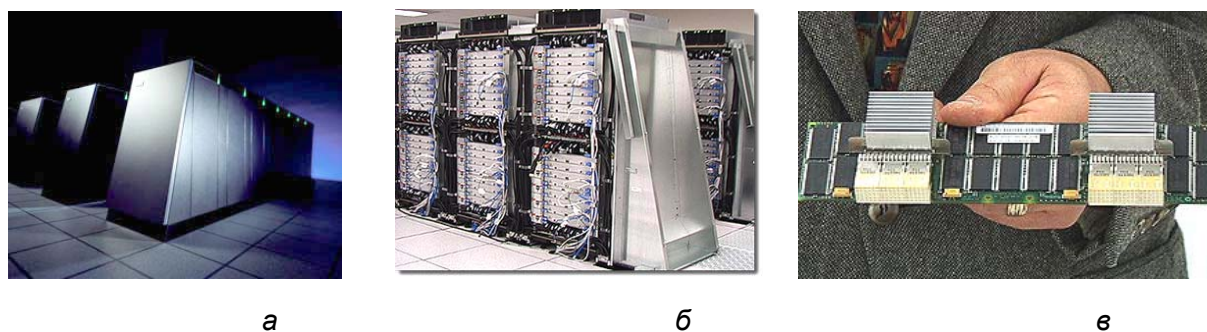
#### **2.1.1. Вычислительные системы**

В настоящее время принята следующая классификация вычислительных систем:

- суперкомпьютеры;
- рабочие станции;
- персональные компьютеры.

**Суперкомпьютеры** — наиболее мощные в данный момент времени вычислительные системы. Для них характерны архитектура, оптимизированная для максимальной производительности при вычислениях с плавающей точкой и лишь самое необходимое программное обеспечение, включающее в себя, как правило, операционную систему и компиляторы FORTRAN и C/C++ с необходимыми средствами отладки.

**Возглавляет рейтинг суперкомпьютеров** (по состоянию на начало 2006 года) векторный суперкомпьютер IBM Blue Gene/L (рис. 3) с производительностью 280,6 триллиона операций с плавающей запятой в секунду (терафлопс). Такой высокой производительности удалось достичь благодаря 131072 процессорам PowerPC 440 с тактовой частотой 700 МГц (в действительности процессоров 65536, но, поскольку они двуядерные, то каждый считается как два), установленным в течение нескольких лет сборки компьютера. Большинство современных вычислительных алгоритмов, основанных на решении систем уравнений, поддается распараллеливанию. Поэтому все современные суперкомпьютеры имеют векторную, или многопроцессорную архитектуру.



*Рис. 3. Суперкомпьютер Blue Gene/L фирмы IBM. а,б — внешний вид; в — вычислительная карта — базовый компонент Blue Gene/L (с сайта terralab.ru)*

Главная особенность Blue Gene/L заключается в высокой степени интеграции: в компьютере применяются встраиваемые центральные процессоры, встраиваемая оперативная память и технология "система-на-чипе", позволяющие объединить в единой интегральной схеме вычислительный процессор, коммуникационный процессор, три уровня кэш-памяти и несколько высокоскоростных сетевых интерфейсов. Благодаря относительно невысокой тактовой частоте процессоров она близка к частоте работы памяти, что также снижает энергопотребление и дает возможность конструировать системы, в которых 1024 блока размещаются в одной серверной стойке.

В качестве примера «среднего» суперкомпьютера можно представить разработку Межведомственного компьютерного центра Российской академии



*Рис. 4. Российский суперкомпьютер MBC-15000BM (с сайта Межведомственного компьютерного центра)*

наук MBC-15000BM (рис. 4) — наиболее мощный в России суперкомпьютер, занимающий 69 строчку в рейтинге наиболее производительных систем. Его производительность составляет около 8 терафлопс и в ближайшее время может быть увеличена примерно вдвое. Компьютер, который, к сожалению, лишь с очень большой натяжкой можно назвать отечественным, содержит 924 микропро-

цессора IBM PowerPC-970+ с тактовой частотой 2,2 ГГц.

**Рабочие станции** представляют из себя персональные вычислительные системы, предназначенные для решения определенных задач. Иногда в качестве рабочих станций позиционируют достаточно мощные персональные компьютеры. Это не совсем верно, так как производительность рабочих станций (как и их цена) существенно отличаются от параметров даже мощных ПК. Так, один из процессоров, используемых в рабочих станциях, UltraSPARC T1, содержит 8 ядер, каждое из которых способно выполнять по четыре потока (технология Chip Multi-Threading). В качестве операционных систем в рабочих станциях используются различные варианты на базе ОС UNIX.

**Из современных рабочих станций** можно привести в качестве примера HP c8000, обеспечивающую очень высокие показатели производительности по операциям с плавающей точкой. Станция функционирует под управлением операционной системы HP-UX 11i TCOE. В ней используются один или два процессора PA-8900 с тактовой частотой 1,1 ГГц с 3 Мб кэш-памяти первого уровня и 64 Мб кэш-памяти второго уровня. Объем оперативной памяти с ECC-коррекцией ошибок может достигать 32 Гб. Предусмотрено размещение одного или двух жестких дисков SCSI Ultra320 максимальной емкостью 1,2 Тб. Подобная система, вполне доступная для средних предпри-

ятий, позволяет решать задачи автоматизированного проектирования практически любой сложности.

При решении задач средней сложности возможно использование **персональных компьютеров**. Здесь надо учитывать следующее:

- процессоры фирмы AMD обеспечивают лучшую производительность на операциях с плавающей точкой по сравнению с процессорами фирмы Intel аналогичного класса;
- наряду с тактовой частотой процессора большое значение имеют его архитектура и размер встроенной КЭШ-памяти;
- большое влияние на производительность при решении задач автоматизированного проектирования оказывает размер оперативной памяти. Желателен объём ОЗУ не менее 1 Гб;
- размер экрана монитора по диагонали должен быть не менее 17" с разрешением не менее 1024×768 точек.

Решение несложных учебных задач с использованием программы SolidWorks 2001 может быть осуществлено и на менее мощных компьютерах — например, Pentium-II с объёмом ОЗУ от 32 Мб. Ещё менее требовательны к ресурсам компьютера «электронные чертёжные доски» — КОМПАС, AutoCAD и т.п.

### 2.1.2. Устройства ввода-вывода информации

Несмотря на прогресс в развитии систем *CAM (Computer Aided Manufacturing)*, позволяющих передавать информацию от программного обеспечения САПР непосредственно к управляющим устройствам станков с ЧПУ, потребность во вводе в компьютер созданных ранее чертежей и выводе чертежей на бумагу достаточно велика.

Для ввода графической информации в компьютер чаще всего используются **сканеры**. В настоящее время наиболее распространены планшетные сканеры формата А4. Но для ввода больших чертежей они неудобны — изображение приходится «склеивать» при помощи графических программ. Поэтому АРМ конструктора желательно оснащать рулонным сканером, позволяющим сканировать изображения, по ширине соответствующие формату А1

либо А0, длина которых ограничивается лишь ресурсами компьютера.

Следует учитывать, что при помощи сканера графические изображения вводятся в растровом формате, то есть в виде совокупности точек. Так как программы САПР оперируют с изображениями, представленными в векторном формате (то есть совокупностью графических примитивов — отрезков, окружностей и т.п.), то после сканирования введённое изображение необходимо *векторизовать*. Для векторизации растровых изображений служат программы, принцип действия которых аналогичен принципу действия программ распознавания текста. Однако при работе этих программ возможны искажения введённого изображения. Поэтому часто введённые растровые изображения «обводят» вручную непосредственно в графических программах САПР.

Используются для ввода графической информации также и **дигитайзеры**, представляющие собой планшет, на котором располагается вводимый чертёж. Под планшетом располагается координатная сетка из натянутых тонких проволок с достаточно малым шагом (как правило — 0,1 мм). Для ввода координаты точки с этой точкой совмещается считывающее устройство, как правило, представляющее из себя линзу с перекрестием посередине. При нажатии кнопки координата точки заносится в программу. Заранее указав тип примитива (отрезок, окружность и т.п.) и его характерные точки (начало и конец отрезка, точку на окружности и её центр и т.п.), можно ввести примитив в программу.

Здесь необходимо иметь в виду, что ввод изображений с бумаги чреват внесением ошибок и неточностей черчения, что бывает допустимо на сборочных чертежах, но крайне нежелательно на рабочих чертежах деталей. Поэтому рабочие чертежи деталей желательно всё же заносить в компьютер вручную, в точном соответствии с номинальными размерами.

Для вывода чертежей на бумагу долгое время применялись **планшетные графопостроители (плоттеры)**. Они представляли собой планшет размером от А3 до А0 и более, на котором закреплялся лист бумаги. Над планшетом перемещалась головка с пером. Использовались перья разной толщины либо разных цветов. Недостаток таких графопостроителей — невысокая скорость, особенно при изображении таких элементов, как штриховка или



текст, а также пониженная точность. В настоящее время они не выпускаются.

Принцип действия современных **рулонных плоттеров** аналогичен принципу действия струйных принтеров. Эти устройства позволяют выводить с высокой скоростью и точностью не только чертежи, но и любые полноцветные изображения. Размер изображения ограничивается его шириной (обычно 594 или 841 мм) и длиной рулона бумаги.

## 2.2. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ САПР

*Программное обеспечение (ПО) САПР* — совокупность программ и программной документации к ним, необходимых для реализации автоматизированного проектирования. Программное обеспечение САПР подразделяется на общесистемное (общее) и специализированное (специальное), или проблемно-ориентированное.

*Общесистемное ПО* предназначено для управления вычислительным процессом; решения общематематических задач; подготовки специализированных программ к исполнению; контроля и диагностики работы вычислительной системы; хранения и поиска информации, необходимой при проектировании. *Специальное ПО* включает в себя прикладные программы, предназначенные для решения проектных задач. Такие программы делят на объектно-ориентированные (объектные) и инвариантные (объектно-независимые).

**Объектно-ориентированные системы** служат для проектирования конкретных узлов и агрегатов. Наряду с оптимизацией их параметров либо структуры такие системы могут осуществлять генерацию их изображения для последующего импорта в графическую программу. Примером здесь могут служить программа выбора оптимального зубчатого зацепления, программа синтеза схем планетарных коробок передач, программа выбора оптимальных параметров автомобильного гидроамортизатора и т.п.

**Инвариантные подсистемы** не зависят от того, какой тип объекта проектируется. Они применимы к проектированию любых машиностроительных изделий, а возможно, и к объектам из других областей техники. Примером инвариантных подсистем могут служить системы автоматизации чертежно-графических работ AutoCAD и КОМПАС-График, САД-системы Mechani-

cal Desktop, КОМПАС-3D, SolidWorks и Pro/Engineer, CAE-системы ANSYS и COSMOS.

**Системы автоматизации чертежно-графических работ или электронные чертёжные доски** предназначены преимущественно для создания 2-мерных чертежей любой сложности. Хотя последние версии этих систем обладают достаточно развитыми возможностями для 3-мерного проектирования, создавать 3-мерные модели в них гораздо менее удобно, чем в специально приспособленных для этого САД-системах. Преимуществом простых графических программ является их низкая требовательность к ресурсам компьютера и простота освоения.

**САД-системы (Computer Aided Design** – система автоматизированного проектирования) используются в основном для процесса собственно проектирования, заменяя собой не только карандаш и бумагу, но и пластилин, дерево и другие средства создания 3-мерных моделей. Как правило, такая модель представляет собой сочетание твердого тела и набора поверхностей. Процесс её построения основывается на создании элементарных геометрических примитивов и выполнения различных операций над ними. Модель строится из стандартных элементов и может быть отредактирована путём добавления (удаления) этих элементов, либо путём изменения их характерных параметров. Изображение деталей, узлов и агрегатов в виде трёхмерных моделей позволяет определить их объём, массу, момент инерции, проверить возможность сборки, а также кинематику механизма. САД-системы позволяют осуществлять параметрическое проектирование, то есть автоматическое изменение размеров проектируемой детали соответственно изменению характерных размеров. Чертежи со всеми необходимыми видами, разрезами и сечениями, а также спецификации сборок генерируются в таких программах автоматически. Причём чертежи обладают двунаправленной ассоциативностью с моделью, то есть, изменения, сделанные в модели, автоматически отображаются на чертежах и наоборот.

**CAE-системы (Computer Aided Engineering** – автоматизированная система определения характеристик проектируемого объекта) позволяют рассчитать кинематические, динамические, прочностные, тепловые, аэро- и гид-

родинамические характеристики проектируемой конструкции. Наиболее широко известные CAE-системы — ANSYS и COSMOS. Расчетные модули могут встраиваться непосредственно в CAD-систему. Высокую оценку разработчиков заслужил расчётный модуль, встроенный в PRO/Engineer. Несложные расчёты на прочность трёхмерных моделей позволяют расчётные модули, встроенные в Mechanical Desktop и SolidWorks.

### 2.3. МАТЕМАТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САПР

*Математическое обеспечение* — совокупность математических методов, моделей и алгоритмов, необходимых для автоматизированного выполнения проектных процедур. Выполнение проектных операций и процедур в САПР основано на оперировании с математическими моделями. С их помощью прогнозируются характеристики и оцениваются возможности различных вариантов структур и параметров.

Разработка математического обеспечения является самым сложным этапом создания САПР, от которого (при использовании условно одинаковых технических средств) в наибольшей степени зависят производительность и эффективность функционирования САПР в целом. Какое влияние на время решения задачи может оказать правильный метод решения может проиллюстрировать следующий пример.

При решении относительно небольшой системы из 30 линейных уравнений методом Крамера требуется вычислить 31 определитель порядка 30. Если делать это «в лоб», то для решения системы понадобится  $31 \times 30! \times 29$  умножений и приблизительно столько же сложений. Если принять, что одно умножение — это одна «операция с плавающей точкой» или FLOP, то только для вычисления мультипликативной части решения данной системы на наиболее мощном суперкомпьютере типа «Blue Gene/L» понадобится около 30 триллионов лет. Однако, используя более совершенные методы, можно решить системы намного более высоких порядков на гораздо менее мощных вычислительных системах.

К математическому обеспечению САПР можно отнести и стандартные библиотеки подпрограмм, реализованных на языках высокого уровня

(ФОРТРАН, а в последнее время — и С++). До сих пор широко используется библиотека SSP фирмы IBM, содержащая подпрограммы из всех разделов вычислительной математики. В настоящее время в сложных научных расчётах применяются более современные библиотеки, например — LINPACK, охватывающая методы линейной алгебры, которые широко используются, в частности, при расчётах методами конечных элементов. Хороший набор вычислительных подпрограмм на языках ФОРТРАН и С++ расположен на сайте Библиотеки вычислительного анализа НИВЦ МГУ (URL <http://www.srcc.msu.su>).

## 2.4. ПРОЧИЕ ВИДЫ ОБЕСПЕЧЕНИЯ САПР

**Информационное обеспечение САПР** представляется в виде баз данных, содержащих сведения, необходимые для выполнения проектирования. Эти базы данных содержат справочные данные об унифицированных элементах, нормалах, ГОСТах; сведения и чертежи типовых и предыдущих проектных разработок; результаты предыдущих этапов проектирования и т.п. Техническая документация для крупных технических проектов может достигать сотен тысяч листов. Упорядоченное хранение этих материалов, обеспечение к ним быстрого доступа со стороны разработчиков является сложной технической задачей. Для её автоматизации используются **PDM-системы (Product Data Management — система управления данными проектирования)**.

**Лингвистическое обеспечение САПР** — совокупность языков для записи алгоритмов, описания исходных данных и результатов, обмена информацией между проектировщиком и ЭВМ в процессе проектирования. При создании расчётных программ используются алгоритмические языки программирования высокого уровня (FORTRAN, С++, реже — PASCAL и др.). Для описания процесса вычерчивания графических отображений объектов проектирования используются встроенные в САД-системы языки программирования. Обычно, это язык С, но в программе AutoCAD фирмы AutoDesk применяется язык AutoLisp.

Для передачи сведений о проектируемом объекте (структура, конструктивные ограничения, условия производства и эксплуатации) разрабаты-

вается язык описания объекта (ЯОО). Например, при помощи ЯОО могут быть описаны правила кодировки кинематической схемы коробки передач, правила задания габаритных и конструктивных ограничений при оптимизации параметров зубчатых зацеплений и т.п. Язык описания заданий (ЯОЗ) служит для задания САПР сведений о том, какая или какие задачи должны быть решены и в каких условиях. Записи на ЯОЗ представляют собой директивы, задаваемые в начале или в процессе проектирования. Так, на ЯОЗ может быть поставлена задача САПР — провести только поверочный расчет или часть расчетов, а после анализа можно ввести в систему директиву графического отображения.

**Организационное обеспечение САПР** — совокупность положений, инструкций, устанавливающих состав и функции подразделений проектной организации, правила и режим работы проектировщиков и обслуживающего персонала, формы документов и т.п.

**Методическое обеспечение САПР** — совокупность документов, в которых отражен состав, правила отбора и эксплуатации средств автоматизации проектирования. В частности, к методическому обеспечению относят описание технологических маршрутов процесса проектирования, т.е. типовых последовательностей выполнения проектных операций и процедур. В каждой САПР описание технологии проектирования конкретизируется и содержит описание последовательности операций, использование различных сочетаний программ с целью рационального распределения функций между человеком и ЭВМ.

### 3. ГРАФИЧЕСКОЕ ОТОБРАЖЕНИЕ ОБЪЕКТОВ ПРОЕКТИРОВАНИЯ

Используются два принципа кодирования графической информации — растровый и векторный.

В **растровом формате** изображение представлено в виде прямоугольника, представляющего из себя матрицу точек с определенной размерностью по горизонтали и вертикали. Цвет каждой точки кодируется определённым числом. Таким образом можно сказать, что файл, содержащий графическое изображение в растровом формате, представляет собой совокупность чисел, описывающих каждую точку. Размер такого файла зависит от количества точек и от глубины цвета. При кодировании чёрно-белого изображения без полутонов каждая точка занимает один бит информации (например, 0 — белый цвет, 1 — чёрный). Для хранения полутоновых чёрно-белых изображений с 256 градациями серого цвета, информация об яркости точки занимает уже 8 бит, или 1 байт. Для хранения цветных изображений объём памяти, отводимой на одну точку, колеблется от 16 до 48 бит. Для хранения столь значительных объёмов информации большинство растровых форматов применяют различные способы сжатия данных. Некоторые растровые форматы, например, jpeg, позволяют весьма значительно уменьшать размер файла не только за счёт сжатия данных, но и за счёт некоторого ухудшения качества изображения. Растровые форматы незаменимы для хранения полутоновых художественных изображений.

Для представления графических отображений объектов проектирования (чертежей, геометрических моделей и т.п.) в САПР используются **векторные форматы**. При таком способе кодирования графической информации любой

чертёж формируется из элементарных графических элементов (графических примитивов). По аналогии с алфавитом это набор геометрических символов, позволяющий создавать любой графический «текст». Минимальный набор (база) графических примитивов включает в себя точку, отрезок прямой (вектор), окружность, дугу. Более сложные примитивы (отрезки прямых со стрелками на концах для простановки размеров; буквы, и другие символы) формируются из базового набора элементарных примитивов. Положение графического элемента на поле чертежа (плоскости) или в объемном пространстве и его размеры определяются заданием координат опорных точек и необходимых размеров. Например, отрезок прямой может быть задан: координатами начальной и конечной точек; координатами начальной точки, углом наклона к горизонтали и длиной. Окружность может быть задана: координатами центра и радиусом; координатами центра и любой точки, лежащей на окружности; координатами любых трех точек, лежащих на окружности. Перед отображением (вычерчиванием) графического элемента задается тип линии (основная, тонкая, пунктирная, штрихпунктирная), а также цвет отображения.

Характерной особенностью векторного изображения является сохранение качества при любом увеличении масштаба. Пример, приведённый на рис. показывает, что качество векторного изображения ограничивается только разрешающей способностью экрана либо графопостроителя.

Геометрические модели для графического отображения объектов проектирования могут быть детерминированными и параметрическими; 2-мерные (на плоскости) и 3-мерные (объемные). **Детерминированная геометрическая модель** — статическое изображение традиционного чертежа, представленное в ЭВМ в виде двоичных кодов, описывающих команды управления процессом его отображения на экране дисплея или твердом носителе (бумаге) в том или ином масштабе. При **параметрических геометрических моделях** путём задания значений параметров можно получать графические отображения с любым соотношением размеров его элементов — например, чертежи болтов с различной длиной при одинаковых головках и так далее. Наиболее важным преимуществом параметрических геометрических моделей является простота внесения изменений и более удобная инте-

грация системы автоматизированного проектирования (САД-системы) с расчетной подсистемой САПР (САЕ-системой).

Создание параметрических моделей имеет много общего с написанием программ. Имея параметрическую модель со всеми размерными зависимостями, которые могут быть выражены в виде математических формул любой сложности, можно без труда модифицировать её, меняя значения размеров и вид зависимостей в соответствующей базе данных. Можно сравнить разработку параметрических моделей в САД-системах с разработкой программ в современных системах визуального программирования.

Трёхмерное представление объекта проектирования дает большие дополнительные возможности для анализа конструкторских решений и автоматического получения отображений проекции на плоскости, разрезов и сечений. В настоящее время имеются программные средства, которые позволяют автоматически получить отображение в любой плоскости сечения, «посмотреть» на объект с любых сторон, «заглянуть» внутрь его, рассмотреть различные его слои. Для качественного объёмного изображения современные САД-системы имеют развитые средства рендеринга — визуализации трёхмерного объекта с учётом материала, расположения и интенсивности источников освещения.



## **4. МАТЕМАТИЧЕСКОЕ ОПИСАНИЕ ОБЪЕКТОВ ПРОЕКТИРОВАНИЯ**

### **4.1. ПОНЯТИЕ МАТЕМАТИЧЕСКОЙ МОДЕЛИ**

При автоматизации проектирования специфика проектируемых объектов проявляется, прежде всего, в их математических моделях. Знание особенностей математических моделей, методов и алгоритмов решения проектных задач необходимо инженеру для постановки задачи, правильной формулировки исходных данных и интерпретации получаемых результатов, а также при принятии решений об использовании тех или иных компонентов математического обеспечения в процессе решения проектных задач.

Несмотря на то, что математические модели разных объектов различны, имеется ряд общих положений, справедливых для многих технических объектов. На каждом иерархическом уровне используются свои математические модели, сложность которых согласуется с возможностями анализа. Так, например, при анализе плавности хода автомобиля амортизатор задаётся своей статической характеристикой. Но при проектировании самого амортизатора для анализа происходящих в нем процессов используются математические модели, учитывающие сжатие жидкости, вибрации клапанов и другие явления. По характеру отображаемых свойств проектируемого объекта модели разделяют на функциональные и структурные.

**Структурные модели** отображают состав элементов и взаимосвязи между ними. Эти модели могут иметь форму матриц, графов, списков векторов и выражать взаимное расположение элементов и наличие отношений между ними. Структурные модели обычно используют в случаях, когда задачи

синтеза структур удастся ставить и решать, абстрагируясь от особенностей физических процессов в объекте проектирования.

Назначение структурных моделей — решение задачи синтеза возможных структур с последующим анализом и выбором наилучшей с учетом характеристик функционирования объекта.

**Функциональные модели** отображают процессы функционирования элементов и (или) объекта проектирования в целом. Например, процессы накопления повреждений до разрушения от изгиба, трения, кручения т.п., которые определяют в заданных условиях срок службы деталей и технического объекта. Модели могут описывать как отдельные стороны функционирования отдельных элементов, рабочих процессов (сгорание газов в цилиндре двигателя, нарастание давления в гидроцилиндре и др.) так и функционирование технического объекта в целом. Для построения функциональной модели должна быть известна структура объекта.

Назначение функциональных моделей — оценка характеристик качества функционирования элементов и технического объекта, которые определяются как структурой объекта так и параметрами элементов и внутренних процессов.

Поскольку структурные и функциональные свойства объектов тесно взаимосвязаны (совместно определяют качества будущего объекта), для реализации автоматизированного проектирования высокого уровня требуются модели с отображением особенностей, как структуры объекта, так и характера физических процессов, происходящих в нем. Это требование реализуется в **обобщенных функциональных моделях**, создаваемых путем соединения отдельных моделей с учетом структурных свойств. Получение функциональной модели для конкретного варианта структуры может производиться и путем усечения обобщенной модели автоматизированным способом по данным, кодирующим заданную структуру.

**Основными требованиями математической модели** являются требования: адекватности отображения основных свойств объекта (процесса), интересующих проектировщика; универсальности — применимости к широкому классу проектируемых объектов; экономичность с точки зрения затрат на

разработку и вычислительных ресурсов (машинного времени и объема оперативной памяти); адаптации — легко подвергаться модификации по мере накопления знаний и выявлении новых свойств и условий; простота и понятность пользователю — залог успеха применения и использования; легко упрощаема и в тоже время легко усложняема. Перечисленные требования находятся в противоречии. Поэтому разработка модели требует определенного искусства, способности анализа, выделения основных свойств. Часто модель можно сделать проще и она дает тот же результат, что и сложная, на которую было затрачено много времени и средств.

Искусство составления матмодели во многом заключается в обоснованном принятии *допущений*, или отказа от учета тех или иных физических явлений. Например, при расчетах на плавность хода транспортных средств, как правило, не учитываются деформации корпуса.

#### 4.2. ПАРАМЕТРЫ МАТЕМАТИЧЕСКОЙ МОДЕЛИ

Фигурирующие в математических моделях объектов величины называют параметрами, которые можно разделить на три основные группы:

$Y = (y_1, y_2, \dots, y_m)$  — *вектор выходных параметров, которые характеризуют различные стороны качества технического объекта;*

$X = (x_1, x_2, \dots, x_n)$  — *вектор внутренних параметров — параметры, характеризующие элементы и внутренние процессы;*

$Q = (q_1, q_2, \dots, q_k)$  — *вектор внешних параметров, которые представляют условия производства и эксплуатации (применения) технического объекта и связи его с другими объектами.*

Целью проектирования является определение структуры и значений вектора внутренних параметров  $X$ , по которым можно будет изготовить технический объект в заданных условиях. Следует отметить, что при переходе к новому уровню проектирования внутренние параметры могут стать выходными, и наоборот, также, как и внешние — внутренними.

Связь между параметрами математической модели может быть представлена в виде

$$Y = F(X, Q), \quad (1),$$

что представляет собой краткую запись в векторной форме совокупности следующего набора функциональных зависимостей:

$$\begin{aligned} y_1 &= f_1(x_1, x_2, \dots, x_n, q_1, q_2, \dots, q_k); \\ y_2 &= f_2(x_1, x_2, \dots, x_n, q_1, q_2, \dots, q_k); \\ y_m &= f_m(x_1, x_2, \dots, x_n, q_1, q_2, \dots, q_k). \end{aligned}$$

Часто связь между выходными, внутренними и внешними параметрами известна не в виде явной зависимости  $\mathbf{Y}$  от  $\mathbf{X}$  и  $\mathbf{Q}$ , а задается в алгоритмической форме, например, через численное решение системы уравнений относительно  $x_1, x_2, \dots, x_n$  при заданных значениях  $y_1, y_2, \dots, y_m$  и  $q_1, q_2, \dots, q_k$ .

Связи между элементами внутри системы (например, условия геометрической совместимости деталей) и с внешними системами представляются в виде функциональных зависимостей (2) и неравенств (3), которые описывают возможные варианты технического объекта по допустимым структурным и функциональным свойствам.

$$\begin{aligned} w_1(\mathbf{X}, \mathbf{Q}) &= 0; \\ w_2(\mathbf{X}, \mathbf{Q}) &= 0; \\ &\dots \dots \dots \\ w_r(\mathbf{X}, \mathbf{Q}) &= 0. \end{aligned} \tag{2}$$

$$\begin{aligned} \psi_1(\mathbf{X}, \mathbf{Q}) &> 0; \\ \psi_1(\mathbf{X}, \mathbf{Q}) &> 0; \\ &\dots \dots \dots \\ \psi_p(\mathbf{X}, \mathbf{Q}) &> 0. \end{aligned} \tag{3}$$

В векторной записи описание связей имеет вид:

$$\mathbf{\Omega}(\mathbf{X}, \mathbf{Q}) = \mathbf{0}; \tag{4}$$

$$\mathbf{\Psi}(\mathbf{X}, \mathbf{Q}) = \mathbf{0}, \tag{5}$$

где  $\mathbf{0} = (0, 0, \dots, 0)$  — ноль-вектор с нулевыми значениями проекций.

Функциональную зависимость, записанную в виде

$$\mathbf{X} = F(\mathbf{Y}, \mathbf{Q}) \tag{6}$$

называют *проектировочной моделью*, которая позволяет прямым способом решить главную задачу проектирования без многовариантного анализа. Получение моделей вида (6) представляет собой трудновыполнимую задачу,

так как обратная зависимость функции (1), как правило, многозначна и даже при возможности определения всех вариантов сочетаний значений  $\mathbf{X}$  не ясно, какой вариант наилучший. Кроме того, получаемые решения, как правило, для технических систем не удовлетворяют условиям (4) и (5). Поэтому в САПР применяются в основном *поверочные математические модели* (1), (4), (5). Наличие их является достаточным для автоматизированного решения главной задачи проектирования — определения наилучшей структуры системы и значений параметров ее элементов. Что же касается начального значения вектора  $\mathbf{X}$ , то оно может быть получено в результате элементарных проектировочных расчётов, интуиции проектировщиков или даже сгенерировано случайным образом.

### 4.3. ЧУВСТВИТЕЛЬНОСТЬ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ

При разработке математической модели или использовании готовых для обеспечения эффективного проектирования важным является выявление существенно значимых взаимосвязей параметров  $\mathbf{Y}$ ,  $\mathbf{X}$ , и  $\mathbf{Q}$  между собой, которые оцениваются степенью влияния изменения одного из них на величину другого. Это производится путем проведения анализа чувствительности, который заключается в определении матрицы чувствительности  $\mathbf{A}$ , элементами которой являются коэффициенты чувствительности (коэффициенты влияния)

$$a_{j,i} = \partial y_i / \partial x_j.$$

Чувствительность к изменению внешних параметров  $q_k$  определяется аналогично. Для исключения влияния размерности на значение  $a_{j,i}$  параметры перед вычислением матрицы чувствительности необходимо нормировать. Нормирование заключается заменой абсолютных значений параметров, зависящих от размерности, относительными приращениями этих параметров относительно некоего номинального значения, например

$$y_{i \text{ норм}} = \Delta y_i / y_{i \text{ ном.}}, \text{ где}$$

$\Delta y_i$  – абсолютное приращение параметра;

$y_{i \text{ ном}}$  – номинальное значение.

В практике проектирования наиболее широко распространен метод

анализа чувствительности, основанный на численном дифференцировании и называемый методом приращений. Его преимущества — простота алгоритма, универсальность; недостатки — невысокая точность вычисления производных; сравнительно большие затраты машинного времени. Эти недостатки в определенной мере устраняются в довольно сложных методах анализа чувствительности, основанных на решении дополнительных систем однородных дифференциальных уравнений — прямой и вариационный методы [9]. Для анализа чувствительности применяют также регрессионный метод, в котором данные для вычисления матрицы чувствительности получают в процессе статистических испытаний по методу Монте-Карло [16].

Сравнивая значения элементов матрицы  $A$  между собой определяют значимость того или иного внутреннего или внешнего параметра. Часть параметров может быть исключена из рассмотрения при проектировании. Например, можно считать, что для транспортных средств, максимальная скорость которых не превышает 70 км/ч, коэффициент аэродинамического сопротивления не играет большой роли и может быть исключен из рассмотрения при проектировании.

Следует иметь в виду, что для большинства математических моделей технических объектов проектирования чувствительность выходных параметров к изменению внутренних или внешних зависит от положения точки в пространстве  $X$  либо  $Q$ . Так значение  $j$ -го абсолютного коэффициента чувствительности к изменению  $i$ -го внутреннего параметра в точке 1 может значительно отличаться от значения в точке 2:

$$a_{ij}|_{X=X_1, Q=Q_1} \gg a_{ij}|_{X=X_2, Q=Q_2}$$

Результаты анализа чувствительности используются для задания величины шага при поиске оптимального проектного решения.

## 5. ПОСТАНОВКА ЗАДАЧ ОПТИМАЛЬНОГО ПРОЕКТИРОВАНИЯ

### 5.1. ОСНОВНЫЕ ПОНЯТИЯ

Одна из важнейших задач САПР — определить такую структуру и значения параметров ее элементов, которые обеспечивают получение оптимального технического объекта, то есть найти **оптимальное проектное решение**. За оптимальное проектное решение (технический объект) в САПР принимается такое решение (технический объект), для которого выбранный определенным образом **критерий качества** имеет наибольшее (наименьшее) значение при выполнении заданных ограничений и условий. Поиск наилучшего значения критерия качества зависит от цели оптимизации, определяемой смыслом выбранного критерия. Например, если за критерий оценки качества проектируемого объекта взять к.п.д., то целью оптимизации будет вариант, для которого значение его будет наибольшим, а для обеспечения высокой проходимости — вариант с наименьшим давлением на грунт.

Для реализации поиска оптимального проектного решения на ЭВМ в автоматическом режиме (без участия человека) необходимо определить один и только один оценочный параметр, по которому ЭВМ будет осуществлять сравнение вариантов технических объектов, и выбирать наилучший. Сведение качества объекта к одному единственному параметру — сложная операция, так как человек обычно оценивает объект по многим параметрам. Но требование о единственности оценочного параметра — критерия качества обусловлено особенностями реализации процессором ЭВМ логических операций сравнения и принятия решения (рис. 5). Оператор 1 на схеме соответ-

ствуется определению значения критерия качества  $K$  для текущего варианта, в операторе 2 значение  $K$  рассматриваемого варианта сравнивается с  $K^*$  — наибольшим значением критерия качества, достигнутым на данный момент проектирования при рассмотрении предыдущих вариантов. Если для рассматриваемого варианта  $K$  имеет большее значение, чем  $K^*$ , то текущий вариант является наилучшим из всех рассмотренных и в операторе 3 производится запоминание этого значения и соответствующих ему значений внутренних параметров  $X^*$  и осуществляется переход к синтезу следующего варианта. Если критерий качества не увеличился, то текущий вариант бракуется и производится

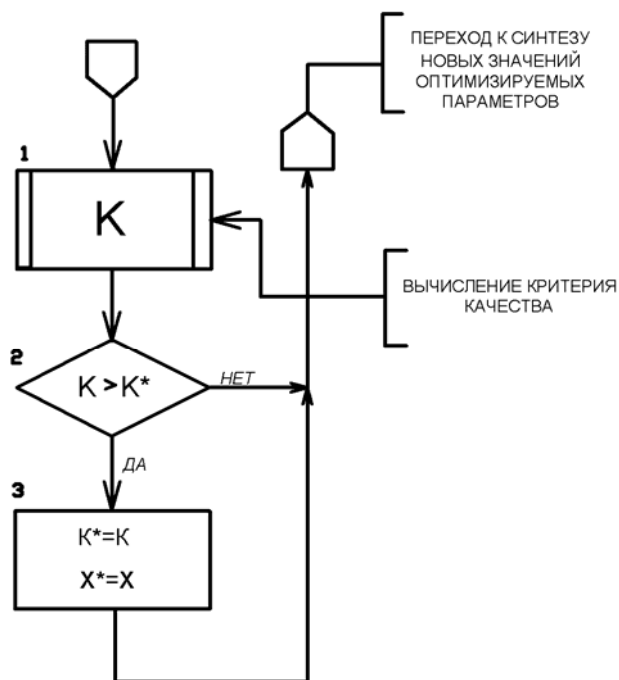


Рис. 5. Общий алгоритм поиска оптимального решения

дится переход к синтезу следующего варианта.

Цель оптимизации формализуется в виде *целевой функции* — математических зависимостей, реализованных в алгоритме определения по управляемым параметрам значения критерия качества

$$K = \Phi(\mathbf{X}), \quad (7)$$

где  $\mathbf{X} = (x_1, x_2, \dots, x_n)$  — вектор управляемых параметров.

Вектор управляемых параметров  $\mathbf{X}$  представляет собой, как правило, набор внутренних параметров оптимизируемого технического объекта. В определенных задачах управляемыми параметрами

могут быть и внешние параметры — при рассмотрении оптимизации условий производства и эксплуатации технического объекта. Чаще всего вектор внешних параметров, описывающий условия производства и эксплуатации, а также связи с другими объектами, определен в техническом задании на проектирование.

Задача оптимального проектирования разбивается на две задачи: задачу



оптимизации структуры и задачу оптимизации параметров ее элементов. Обычно одновременная оптимизация структуры и параметров не реализуется из-за сложности формализации задачи синтеза и задачи анализа структуры. Синтез структур ряда технических объектов поддается формализации, например, синтез кинематических схем планетарных передач. Но формулировка понятия оптимальной структуры на уровне кинематических схем остается трудно формализуемой. Поэтому решение задач оптимизации технического объекта чаще всего реализуется путем неформализованного синтеза ряда структур. Для каждой структуры ЭВМ решает задачу оптимизации параметров ее элементов. Затем проектировщик путем анализа полученных значений критерия качества в сочетании с другими неформализованными оценками качества структуры на основании опыта и интуиции принимает решение об оптимальном сочетании структуры и параметров ее элементов.

Кроме формулировки понятия оптимальности и описания целевой функции в постановку задачи оптимизации входит задание описания ограничений по допустимости значений ряда параметров и условий возможности существования того или иного варианта объекта по геометрической совместности элементов, физическим, технологическим и другим соображениям. Описание области возможных проектных решений задается в виде функций ограничений типа равенства (8) и неравенства (9)

$$\Omega(\mathbf{X}) = 0; \quad (8)$$

$$\Psi(\mathbf{X}) > 0. \quad (9)$$

Из математического описания ограничений в виде (8) или (9) формально следует, что любая функция ограничений есть зависимость от всех управляемых параметров:

$$\mathbf{X} = (x_1, x_2, \dots, x_n).$$

В конкретном случае функция ограничений может представлять зависимость лишь от одного или части параметров. В этом случае условно считают, что в функцию входят и остальные параметры в виде  $a_i \cdot x_i$  со значением коэффициента  $a_i=0$ .

Частным случаем ограничений типа равенства является фиксация значения управляемого параметра  $x_i = \text{const}$ . Частным случаем ограничений типа

неравенства являются прямые ограничения, имеющие вид  $x_{i,\min} < x_i < x_{i,\max}$ , где  $x_{i,\min}$  и  $x_{i,\max}$  — предельно допустимые значения параметра  $x_i$ , например, по возможности физического функционирования объекта.

Таким образом, итоговая формулировка задачи параметрической оптимизации для САПР имеет вид:

найти экстремум целевой функции

$$K = \Phi(\mathbf{X}), \quad (10)$$

при обеспечении выполнения ограничений:

$$\begin{aligned} w_1(\mathbf{X}) &= 0; \\ w_2(\mathbf{X}) &= 0; \end{aligned} \quad (11)$$

.....

$$\begin{aligned} w_l(\mathbf{X}) &= 0; \\ \psi_1(\mathbf{X}) &> 0; \\ \psi_2(\mathbf{X}) &> 0; \end{aligned} \quad (12)$$

.....

$$\psi_p(\mathbf{X}) > 0.$$

Задача оптимизации в такой постановке есть задача *математического программирования*. Если целевая функция и функции ограничений имеют некоторый специальный вид, то задача оптимизации относится к задачам одного из разделов математического программирования. Так, при линейности всех функций (10), (11) и (12) имеем *задачу линейного программирования*; если хотя бы одна из функций нелинейна — *задачу нелинейного программирования* (частными случаями задачи нелинейного программирования являются задачи *квадратичного, выпуклого, геометрического программирования*); если все (или часть) управляемых параметров являются дискретными величинами — задачу *дискретного (или частично дискретного) программирования*. Дискретное программирование называется *целочисленным*, если  $\mathbf{X}$  принадлежит множеству целых чисел.

**Решением задачи оптимизации** является оптимальная точка  $\mathbf{X}^*$ , в которой достигается наибольшее (при максимизации) или наименьшее (при минимизации) значение критерия качества (целевой функции)

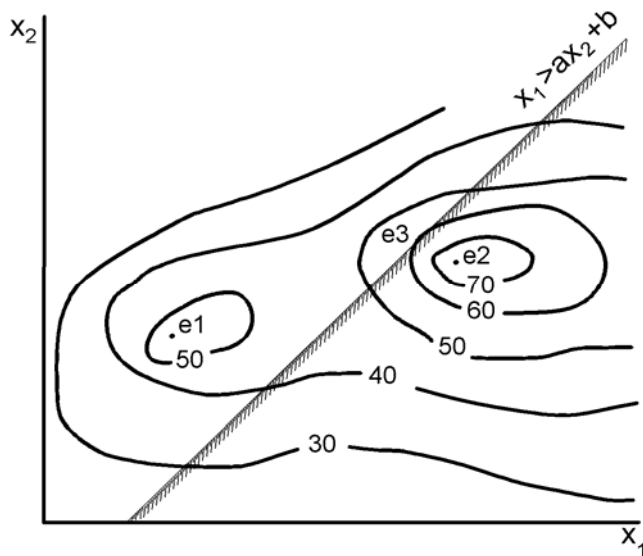


Рис 6. Целевая функция

$$\mathbf{X} \in \{\mathbf{X}\} .$$

При изучении вопросов оптимизации удобны геометрические представления. Так, множество векторов (точек)  $\mathbf{X}$  рассматривается как **пространство оптимизации**, или пространство управляемых параметров. Функцию цели представляют гиперповерхностями равного уровня. В точках конкретной гиперповерхности равного уровня целевая функция равна постоянной величине. Уравнение гиперповерхности имеет вид

$$\Phi(\mathbf{X}) = \text{const} .$$

В частном случае двумерного пространства оптимизации

$$\mathbf{X} = (x_1, x_2)$$

гиперповерхности равного уровня вырождаются в линии равного уровня. На рис. 6 приведен пример целевой функции от двух управляемых параметров, имеющей при отсутствии ограничений два экстремума: в точках  $\mathbf{X}_{e1}$  и  $\mathbf{X}_{e2}$ . При этом точка  $e_2$  — глобальный экстремум, так как  $K_{e2} > K_{e1}$ . При отсутствии ограничений задача оптимизации есть задача безусловной оптимизации, при наличии ограничений — задача условной оптимизации.

Если в примере, изображенном на рис. 6, ввести ограничение  $x_2 > ax_1 + b$ , то областью допустимых значений будет множество точек, лежащих слева от прямой. В этом случае целевая функция будет иметь два экстремума: в точках  $\mathbf{X}_{e1}$  и  $\mathbf{X}_{e3}$ . При этом точка  $e_3$  будет являться наименьшим значением и решени-

$$\Phi(\mathbf{X}^*) = K_{\max} \text{ либо } K_{\min}$$

при выполнении ограничений:

$$\Omega(\mathbf{X}^*) = 0;$$

$$\Psi(\mathbf{X}^*) > 0.$$

Множество сочетаний значений, для которых выполняются функции ограничений, называется областью допустимых значений (ОДЗ) и обозначается  $\{\mathbf{X}\}$ . Таким образом, кратко, выполнение ограничений записывается:

ем задачи оптимизации при наличии данного ограничения является оптимальная точка  $\mathbf{X}^* = \mathbf{X}_{e3}$ .

Из рис. 6 видно, что положение оптимальной точки  $\mathbf{X}^*$  и достигаемое максимальное значения критерия качества  $K^*$  зависит от поставленных ограничений. В данном примере оптимальная точка лежит на (вблизи) границы области допустимых значений (ОДЗ). Для технических объектов проектирования расположение оптимального решения на границе ОДЗ является наиболее распространенным случаем.

Таким образом, в САПР оптимальность проектного решения (технического объекта) определяется как критерием качества и математическим описанием его (целевой функцией), так и поставленными ограничениями. Изменение ограничений может вызвать и изменение значений оптимальных параметров и характеристик качества технических объектов. Поэтому часто при проектировании рассматривают решение ряда задач оптимизации с различными вариантами задаваемых ограничений.

Путем преобразования целевой функции задача минимизации может быть сведена к задаче максимизации и наоборот:

$$K' = -K = -\Phi(\mathbf{X}), \quad (13)$$

где  $K'$  — преобразованный критерий качества.

Далее рассматривается ряд методов постановки задачи оптимального проектирования.

## **5.2. ПОСТАНОВКА ЗАДАЧИ ОПТИМАЛЬНОГО ПРОЕКТИРОВАНИЯ ПО ПРИНЦИПУ РАВНОПРОЧНОСТИ**

Кроме постановки задач оптимизации с формализацией оценки оптимальности технического объекта в виде критерия качества (целевой функции) используют постановки задач проектирования оптимальных технических объектов на основе эвристических соображений и принципов. Примером эвристической формулировки является постановка задачи оптимального проектирования по принципу равнопрочности.

Согласно принципу равнопрочности полагается, что технический объект будет оптимальным, если возникающие механические напряжения во всех элементах (точках элемента) будут одинаковы и равны допускаемым. В

этом случае постановка задачи формализуется в виде системы уравнений:

$$\begin{aligned} f_1(x_1, x_2, \dots, x_n) &= [\delta]; \\ f_2(x_1, x_2, \dots, x_n) &= [\delta]; \\ &\dots\dots\dots \\ f_n(x_1, x_2, \dots, x_n) &= [\delta], \end{aligned}$$

где  $f_i(x_1, x_2, \dots, x_n)$  — функция, определяющая действующие напряжения в  $i$ -м элементе (точке);  $[\delta]$  — предельно допустимое значение напряжения.

Решение задачи нахождения оптимального варианта сводится к решению данной системы уравнений. Полученное решение  $X^*$  считается оптимальным с точки зрения минимального веса (объема) технического объекта при выполнении условия прочности. Однако область применения принципа равнопрочности ограничивается, потому что не учитывается: различное время действия напряжений в различных элементах; элементы имеют различные виды напряженного состояния (изгиб, кручение, смятие и т.д.). Кроме того, при наличии ограничений решение системы уравнений может не входить в ОДЗ:  $X^* \in \{X\}$  и тогда принцип равнопрочности не реализуется.

### 5.3. ПОСТАНОВКА ЗАДАЧИ ОПТИМАЛЬНОГО ПРОЕКТИРОВАНИЯ ПО ПРИНЦИПУ РАВНОДОЛГОВЕЧНОСТИ

Аналогичной эвристической постановкой задачи проектирования оптимального технического объекта является постановка по принципу равнодолговечности:

$$\begin{aligned} f_1(x_1, x_2, \dots, x_n) &= R; \\ f_2(x_1, x_2, \dots, x_n) &= R; \\ &\dots\dots\dots \\ f_n(x_1, x_2, \dots, x_n) &= R, \end{aligned}$$

где  $R$  — номинальное значение ресурса работы технического объекта.

Постановка по принципу равнодолговечности учитывает различия по времени величины и нерегулярности нагружения различных элементов, но сохраняются остальные недостатки, имеющие место в постановке по принципу равнопрочности. Обе приведенные постановки не учитывают случайный характер нагружения и рассеяния характеристик точности и сопротивления усталости материалов.

## 5.4. ПОСТАНОВКА ЗАДАЧИ ОПТИМИЗАЦИИ ПО ЧАСТНОМУ КРИТЕРИЮ

Данная постановка задачи оптимизации заключается в том, что из множества выходных параметров

$$\mathbf{Y} = (y_1, y_2, \dots, y_n),$$

оценивающих различные стороны качества технического объекта, выбирается наиболее существенный, важный и по его значению производится сравнение и выбор наилучшего варианта  $K = y_i = f_i(\mathbf{X})$ .

Для остальных или части выходных параметров устанавливаются прямые ограничения по допустимости значений. Например:

$$y_1 = f_1(\mathbf{X}) > y_{1,\min};$$

$$y_2 = f_2(\mathbf{X}) < y_{2,\max};$$

$$y_3 = f_3(\mathbf{X}) = y_{3,\text{nom}}.$$

В совокупности с другими ограничениями (11), (12) они определяют область допустимых значений управляемых (оптимизируемых) параметров.

## 5.5. МЕТОД ВЕСОВЫХ КОЭФФИЦИЕНТОВ

За критерий качества здесь принимается сумма частных критериев с учетом веса (значимости) влияния их на качество технического объекта в целом. Такой критерий является обобщенным критерием и носит название аддитивный. Он вычисляется по формуле

$$K = c_1 \cdot y_1 + c_2 \cdot y_2 + \dots + c_m \cdot y_m$$

где  $c_i$  — весовые коэффициенты.

Для исключения зависимости частных критериев  $y_i$  и весовых коэффициентов от размерности используют относительные безразмерные критерии  $y_i/y_{i,n}$ , где  $y_{i,n}$  — номинальное значение частного критерия. Значения  $y_{i,n}$  определяются, как номинальные для прототипа или как требуемые (желаемые) для проектируемого технического объекта. Тогда постановка задачи будет иметь вид:

$$\max K = \max \sum_{i=1}^m c_i y_i / y_{i,n},$$

где  $y_i = f_i(\mathbf{X})$ .

Главная сложность этого метода заключается в определении значений весовых коэффициентов  $c_i$ , которые устанавливаются проектировщиком на основании анализа опыта эксплуатации и проектирования предшествующих технических объектов данного типа. Коэффициенты  $c_i$  являются эвристическими оценками значимости частных характеристик в общем качестве объекта. Для формализации обобщения опыта большого числа специалистов значения весовых коэффициентов определяют методом экспертных оценок, что уменьшает степень субъективного характера оценок.

## 5.6. МАКСИМИННАЯ И МИНИМАКСНАЯ ПОСТАНОВКИ ЗАДАЧИ ОПТИМИЗАЦИИ

В качестве целевой функции при максиминной постановке берется функция минимума:

$$F(\mathbf{X}) = \min (Z_1(\mathbf{X}), Z_2(\mathbf{X}), \dots, Z_m(\mathbf{X})),$$

где  $Z_i(\mathbf{X})$  есть функция, определяющая работоспособность по тому или иному виду (элементу) функционирования. Постановка задачи оптимизации будет иметь вид:

$$\max K = \max \min Z_i(\mathbf{X}),$$

где  $i=1 \dots m$ .

Примером максиминной постановки является проектирование с обеспечением максимального значения ресурса работы наиболее слабого элемента —  $\max \min (R_1, R_2, \dots, R_m)$ , где  $R_i = f_i(\mathbf{X})$  — ресурс работы  $i$ -го элемента.

В процессе решения задачи оптимизации в такой постановке наиболее слабым элементом является тот элемент, ресурс которого имеет по сравнению с остальными наименьшее значение. Таким образом, в зависимости от конкретных значений вектора управляемых параметров  $\mathbf{X}$ , наиболее слабым элементом может быть любой элемент. В этом случае полученное решение максимальным образом приближается к обеспечению равной долговечности всех элементов и при отсутствии ограничений реализуется принцип равнодолговечности.

Аналогичное содержание имеет и постановка задачи оптимизации с обеспечением минимального значения максимального механического напря-

жения из всех возникающих в элементах (точках) технического объекта —  $\min \max (\delta_1, \delta_2, \dots, \delta_n)$ , где  $\delta_i = f_i(\mathbf{X})$  — функция величины действующего механического напряжения в  $i$ -м элементе (точке). Такая постановка называется минимаксной.

## 5.7. ПОСТАНОВКА ЗАДАЧИ ОПТИМИЗАЦИИ ПО КРИТЕРИЮ ВЕРОЯТНОСТИ БЕЗОТКАЗНОЙ РАБОТЫ

За критерий качества принимается вероятность безотказной работы технического объекта за номинальный срок эксплуатации  $R_N$ . Оптимальным техническим объектом в этом случае будет тот, у которого при заданных ограничениях значение вероятности безотказной работы будет максимальным. Для решения задачи в такой постановке необходимо установить математическое описание функции плотности распределения вероятности появления отказа для каждого элемента технического объекта и вида отказа. Если  $f_i(\mathbf{X}, L)$  — плотность распределения вероятности появления отказа  $i$ -го элемента (вида), то вероятность безотказной работы этого элемента в течение требуемого срока эксплуатации  $R_N$  определяется выражением:

$$P_i = 1 - \int_0^{R_N} f_i(\mathbf{X}, L) dL.$$

Теория надежности [7] позволяет всю совокупность оценок вероятностей безотказной работы отдельных элементов (видов)  $P_1, P_2, \dots, P_i, \dots, P_n$  свести к одной обобщенной оценке, определяющей безотказность работы технического объекта в целом как по отказам отдельных элементов, так и по различной физической природе возникновения отказа и видам выполняемых элементами функций в системе при ее эксплуатации.

Если выход из строя какого-либо элемента приводит к отказу всей системы (технического объекта), то вероятность безотказной работы системы в целом определяется произведением вероятностей безотказной работы отдельных элементов. В этом случае  $P$  есть мультипликативный критерий:

$$P = P_1 \cdot P_2 \cdot \dots \cdot P_m = \prod_{i=1}^m P_i. \quad (14)$$

Это соответствует последовательному соединению элементов в функ-



циональной схеме технического объекта. При этом реальное, физическое соединение элементов или последовательность выполняемых функций при эксплуатации может иметь любую структуру. При параллельном соединении (с дублированием) элементов в функциональной схеме технического объекта, когда при отказе какого-либо элемента производится включение дублирующего и обеспечивается таким образом продолжение функционирования системы, вероятность безотказной работы определяется выражением:

$$P = 1 - 2 \prod_{i=1}^m (1 - P_i). \quad (15)$$

Для функциональных схем со смешанным соединением вероятность безотказной работы определяют путем последовательного применения зависимостей (14), (15).

При оптимизации по критерию вероятности безотказной работы обеспечивается нахождение варианта технического объекта с максимальной надежностью его работы при заданных габаритных, весовых, конструктивных, технологических и других ограничениях. На рис. 8 приведен пример задачи оптимального распределения заданной величины суммы  $x_1 + x_2 = m$  (например, выделенной массы металла) между двумя элементами. В этом случае увеличение одного из параметров приводит к повышению надежности одного элемента и снижению величины вероятности безотказной работы другого:

$$\uparrow P_1 = f(x_1) \Rightarrow \uparrow x_1 \Rightarrow \downarrow P_2 = f(x_2) = f(m - x_1).$$

При проектировании с обеспечением равнодолговечности элементов решением является

$$X' = (x_1', x_2'),$$

а с обеспечением максимальной безотказности

$$X^* = (x_1^*, x_2^*).$$

Последнее решение обеспечивает значение вероятности безотказной работы за требуемый срок эксплуатации (см. рис. 7):

$$P^* = P_1^* \cdot P_2^* = 0,9999 \cdot 0,5 \approx 0,5,$$

что существенно больше, чем решение по принципу равнодолговечности:

$$P' = P_1' \cdot P_2' = 0,6 \cdot 0,6 = 0,36.$$

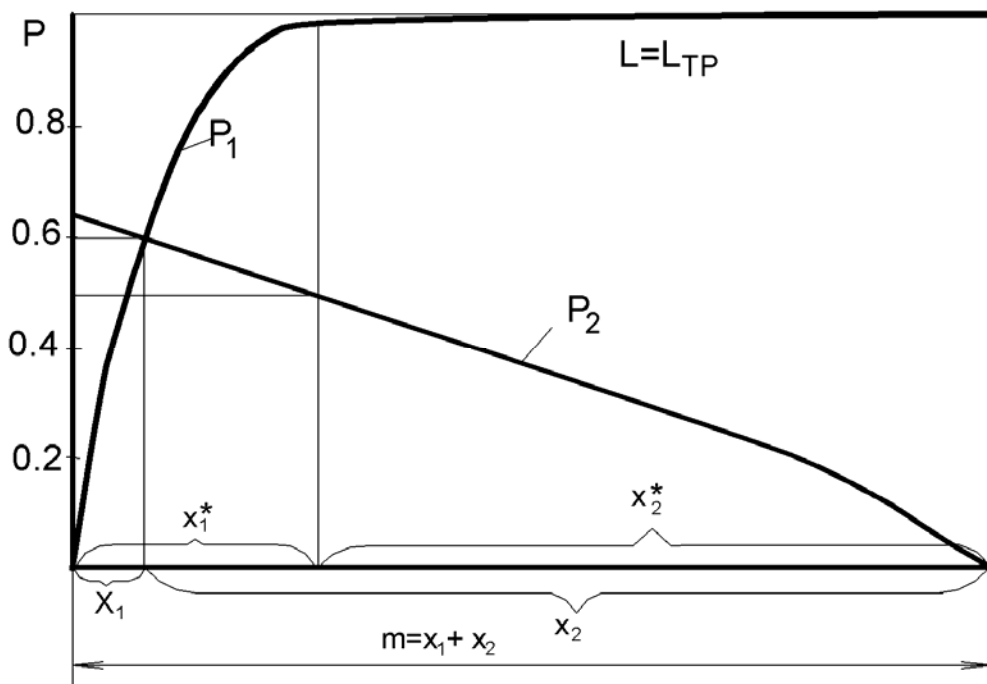


Рис. 7. Пример задачи оптимального распределения заданной массы

Проектирование по принципу равнодолговечности часто приводит к тому, что количество «слабых элементов» в системе становится больше и надежность работы системы в целом уменьшается по сравнению с проектированием по принципу максимальной безотказности.

Недостатком постановки задачи оптимизации по критерию вероятности безотказной работы является малая чувствительность целевой функции к изменению управляемых параметров при значениях вероятностей близких к единице. При больших запасах работоспособности всех элементов ( $P_i=1$ ) может возникнуть ситуация, когда для любого сочетания значений управляемых параметров в пространстве оптимизации функция цели имеет одинаковое значение  $P = F(\mathbf{X}) = 1$ . В этом случае применяют приемы искусственного повышения чувствительности целевой функции с обеспечением сохранения положения точки экстремума  $\mathbf{X}^*$  в пространстве оптимизации.

В постановке задачи оптимизации по критерию вероятности безотказной работы остаются и вопросы описания ограничений по возможности существования технического объекта с теми или иными значениями управляемых параметров. Полученное таким образом математическое описание пред-

ставляет собой обобщенную математическую модель, объединяющую все элементы системы и различные стороны их функционирования в одном критерии качества. На основе такого подхода осуществима интеграция процесса проектирования — объединение отдельных этапов синтеза параметров по элементам в задачу одновременного синтеза всех параметров системы с учетом взаимосвязи между ними.

Существуют и другие постановки задач оптимального проектирования.

## **6. МЕТОДЫ РЕШЕНИЯ ЗАДАЧ ОПТИМАЛЬНОГО ПРОЕКТИРОВАНИЯ**

### **6.1. ОСНОВНЫЕ ПОНЯТИЯ**

Значительная часть задач оптимизации технических объектов транспортного машиностроения относится к задачам нелинейного программирования с ограничениями. Математическое описание целевой функции и ряда функций ограничений в этих задачах, как правило, представляет собой алгоритмические модели, что обуславливает необходимость применения поисковой (шаговой) оптимизации. Классические методы нахождения экстремумов целевых функций аналитическим путём непосредственно в САПР практически не применяются, так как случаи аналитического задания функций крайне редки.

Поисковая оптимизация заключается в расчете функций ограничений и целевой функции для последовательного ряда точек в направлении приближения к достаточно малой окрестности оптимальной точки  $X^*$ , значение критерия качества в которой имеет наибольшее (наименьшее) значение и выполняются заданные ограничения. Последовательность рассмотрения точек определяется методом поиска наибольшего (наименьшего) значения при заданных ограничениях. Основные требования к методу поиска — сходимость алгоритма поиска, обеспечение нахождения глобального экстремума и малые потери на поиск при приемлемой точности нахождения экстремума. Потери на поиск — время, затрачиваемое на синтез вариантов сочетаний значений управляемых параметров и вычисление критерия качества в процессе поиска.

Эффективность алгоритма поиска определяется числом обращений к модели технического объекта (число вариантов анализа) и затратами времени на выбор направления поиска — задание точки перехода от рассмотренной.

Потери на поиск и точность определения экстремума зависят от особенности конкретной целевой функции и функций ограничений и выбранного метода поиска. Метод эффективный, надежный для решения одной задачи, может оказаться неэффективным для решения других задач или не иметь сходимости вообще. Поэтому на практике используется множество алгоритмов, реализующих различные методы оптимизации. Сущность метода определяется тем, как выбираются направления поиска в различных точках пространства оптимизации. Каждый метод может иметь несколько вариантов алгоритмической реализации, отличающихся способами нормализации переменных (переходом к относительным безразмерным параметрам), определением условия окончания поиска, заданием шага смещения в выбранном направлении и так далее. При этом смена варианта алгоритмической реализации в рамках одного метода может привести к большим изменениям показателей эффективности поиска, чем смена самого метода.

Методы поиска минимума можно классифицировать по следующим признакам. В зависимости от характера экстремума, на поиск которого ориентирован метод, различают методы условной и безусловной, локальной и глобальной оптимизации. В зависимости от числа управляемых параметров — методы одномерного и многомерного ( $n$ -мерного) поиска. В зависимости от характера информации (испытания модели), используемой для выбора направления, различают методы нулевого, первого и второго порядка. В методах первого порядка используются первые частные производные целевой функции по управляемым параметрам, которые составляют вектор — градиент целевой функции. Поэтому эти методы называют также градиентными методами. В методах второго порядка, кроме первых производных, используются вторые, составляющие матрицу Гессе[5]. В методах нулевого порядка при выборе направления поиска используются только значения целевой функции.

Общая схема поисковой оптимизации представлена на рис. 8 .

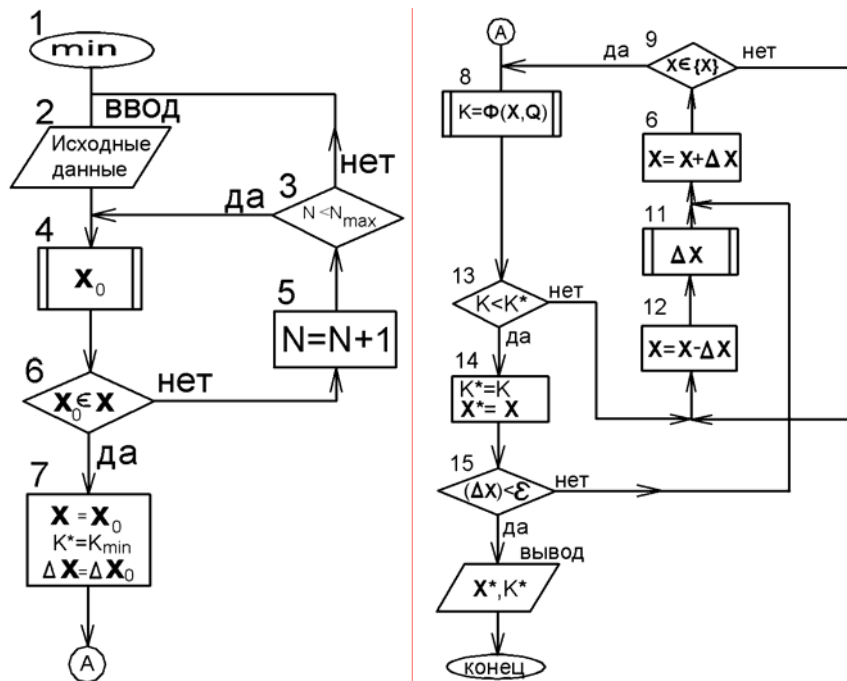


Рис 8. Общий алгоритм пошагового поиска

Перед началом поиска экстремума производится формализация и ввод технического задания — оператор 2. В исходных данных задается: описание рассматриваемой структуры технического объекта; значения вектора внешних параметров  $Q$  и константы функций ограничений; требуемая точность нахождения экстремума  $\varepsilon$ . В зависимости от степени универсальности конкретной реализации алгоритма ряд данных может определяться в самом алгоритме, а часть задаваться в виде исходных данных. Так вектор коэффициентов чувствительности целевой функции  $A$  может быть определен путем испытания  $F(X)$  на ЭВМ отдельно. Более эффективно предусмотреть анализ чувствительности в процессе поиска экстремума, так как чувствительность функции цели к изменению управляемых параметров может существенно зависеть от положения рассматриваемой точки в пространстве оптимизации. Результаты анализа чувствительности используются для определения шага поиска:

$$\Delta X = (h/a_1, h/a_2, \dots, h/a_n), \quad (16)$$

где  $h$  — коэффициент рабочих шагов, задаваемый в исходных данных.

Таким образом, смещение по любой координате вызывает примерно одинаковое изменение целевой функции, что обеспечивает эффективность

оптимизации и исключение «глушения» каким-либо значимым параметром других. В градиентных методах учет различия чувствительности по параметрам обеспечивается автоматически.

Для любого метода поисковой оптимизации требуется перед началом поиска задать (определить) какую-либо точку начальных значений управляемых параметров  $\mathbf{X}_0$ , для которой выполняются все ограничения вида  $\mathbf{X}_0 \in \{\mathbf{X}\}$ .

Вектор  $\mathbf{X}_0$  называют начальным проектным решением. При решении задач совершенствования (модернизации) технического объекта за начальное проектное решение может быть взято сочетание значений его параметров. При проектировании новых технических объектов или совершенствований существующих с изменением ограничений и внешних параметров  $\mathbf{Q}$  определение начального вектора  $\mathbf{X}_0$  часто представляет собой сложную трудоемкую задачу. Поэтому целесообразно в САПР реализовывать автоматизацию решения задачи определения  $\mathbf{X}_0$ , принадлежащего ОДЗ, например, методом Монте-Карло.

На рис. 9 операторы 3–6 осуществляют поиск начальной точки:  $\mathbf{X}_0 \in \{\mathbf{X}\}$ . В операторе 4 производится синтез сочетания значений  $\mathbf{X}_0$ , в операторе 6 — проверка выполнения ограничений. Если для текущего варианта значений ограничения выполняются, то они принимаются за начальные и осуществляется переход к оптимизации. Если  $\mathbf{X}_0 \notin \{\mathbf{X}\}$ , то производится переход к синтезу следующего сочетания значений  $\mathbf{X}_0$  до тех пор, пока число рассмотренных вариантов не достигнет заданного достаточно большого значения  $N_{\max}$ . При  $N=N_{\max}$  с определенной вероятностью (степенью уверенности) можно считать, что для заданных ограничений область допустимых значений не существует  $\mathbf{X}=0$ , то есть данное техническое задание принципиально не выполнимо. В этом случае осуществляется переход на изменение исходных данных.

Собственно поиск минимума критерия качества  $K^*$  и соответствующих оптимальных значений управляемых параметров  $\mathbf{X}^*$  начинается с оператора 8, где для текущей точки  $\mathbf{X}$  определяется значение целевой функции — критерия качества. В операторе 14 осуществляется запоминание варианта с наименьшим на данный момент поиска значением критерия качества — удачный шаг. После удачного шага процесс поиска продолжается с тем же вектором смещения  $\Delta\mathbf{X}$  — следующая точка задается оператором 10. Если переход в точку  $\mathbf{X} = \mathbf{X} + \Delta\mathbf{X}$

не приводит к нарушению ограничений ( $\mathbf{X} \in \{\mathbf{X}\}$ ), то в операторе 8 опять производится определение значения критерия качества. При неудачном шаге — значение критерия качества не уменьшается или нарушаются ограничения, в операторе 12 производится возврат — смещение на один шаг назад, а затем в операторе 11 определяется (генерируется) новое направление поиска  $\Delta\mathbf{X}$ . Если определенное достаточно большое число раз генерации нового направления не привело ни к одному удачному шагу, то в операторе 11 производится уменьшение длины шага поиска, например методом половинного деления

$$\Delta\mathbf{X} = \Delta\mathbf{X}/2 .$$

Процесс поиска прекращается, если длина вектора смещения достигла значения меньше заданной точности  $|\Delta\mathbf{X}| < \varepsilon$ . В этом случае в операторе 16 осуществляется вывод:  $\mathbf{K}^*$  — наименьшего значения критерия качества из всех рассмотренных в процессе поиска и соответствующих ему значений управляемых параметров  $\mathbf{X}^*$ .

Основные различия одного метода поисковой оптимизации от другого заключаются в алгоритме определения направления поиска вектора смещения  $\Delta\mathbf{X}$ . В большинстве методов он определяется по выражению (16) с уменьшением длины  $\Delta\mathbf{X}_2$  до  $\varepsilon$  вблизи экстремума. Значение  $h$  задается в исходных данных перед началом поиска. Слишком малые значения длины шага увеличивают потери на поиск, слишком большие приводят к большому числу неудачных шагов — «перешагиванию» через экстремум и границы ОДЗ, что тоже увеличивает потери на поиск. Длина шага обычно выбирается такой, чтобы приращение любого из параметров не превышало 5..10% от диапазона его возможных значений. По мере приближения к экстремуму шаг уменьшается путем последовательных делений пополам.

## 6.2. МЕТОД ПОЛНОГО ПЕРЕБОРА (МЕТОД СЕТОК)

Данный метод заключается в рассмотрении значений целевой функции и функции ограничений во всех точках заданной области. Алгоритм перебора точек представляет собой последовательное изменение значений управляемых параметров в диапазонах от  $x_{i,\min}$  до  $x_{i,\max}$  с шагом  $\Delta x_i$ . Выбор диапазона и шага определяется предельной областью возможных значений управляемых



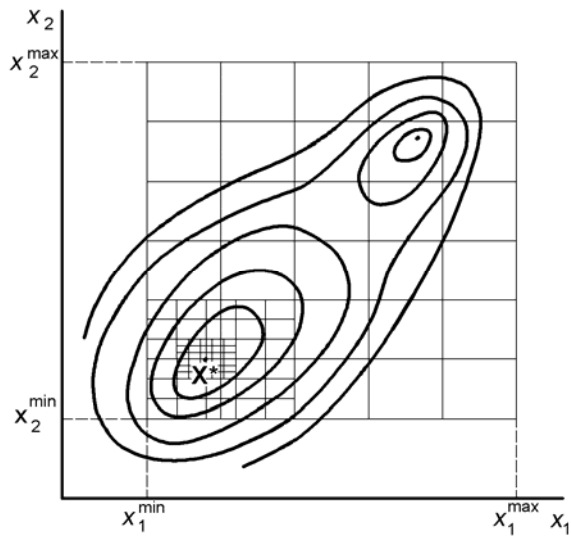


Рис. 9. Схема реализации метода сеток с заданной точностью

параметров и чувствительностью целевой функции к их изменению. Для двухпараметрической функции на рис. 9 приведено геометрическое представление рассматриваемых точек — узлов сетки. Преимуществами данного метода являются простота реализации и нахождение минимального в заданной области значения целевой функции (метод является глобальным). Однако для обеспечения необходимой точности в заданной области число узловых точек может быть слишком велико. Поэтому на

практике этот метод применяют для определения характера целевой функции и нахождению достаточно грубого приближения точки минимума. Для дальнейшего уточнения оптимального решения может быть применен *метод сеток с заданной точностью*, когда в окрестности узловой точки с минимальным значением целевой функции выделяется область и «строится» сетка с уменьшенными размерами ячейки.

Следует отметить, что в силу степенной зависимости затрат времени от размерности задачи, данный метод нецелесообразно применять для решения задач с размерностью, превышающей 4...5.

### 6.3. МЕТОД ПОКООРИНАТНОГО СПУСКА (ГАУССА-ЗЕЙДЕЛЯ)

Выбирается начальная точка  $X_0$  и в ней вычисляется значение  $K$ . Затем определяется новая точка, отстоящая от начальной на величину шага  $\Delta X = (\Delta x_1, 0, 0, \dots, 0)$  вдоль первой координатной оси (рис. 10). Обычно принимают  $\Delta x_1 = \Delta x_2 = \dots = \Delta x_n = h$ , хотя в случае существенно разной чувствительности целевой функции к различным параметрам шаги по координатам могут быть различными. Если попытка была удачной (значение критерия качества уменьшилось по сравнению с начальной точкой), то следующие точки отклады-

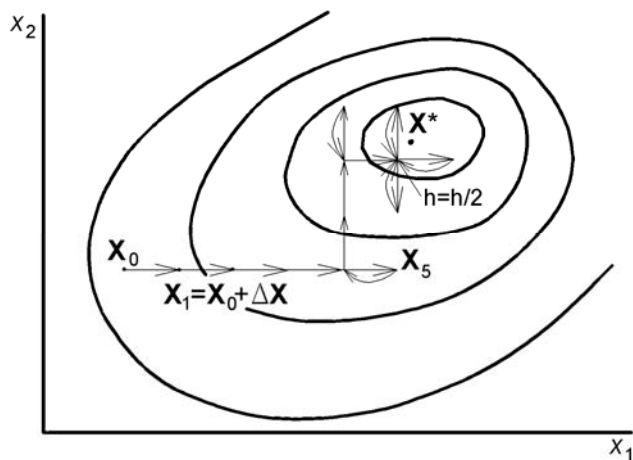


Рис. 10. Схема метода покоординатного спуска

деляется точка, отстоящая от начальной на величину шага  $\Delta X = (-\Delta x_1, 0, 0, \dots, 0)$  вдоль той же координатной оси

После цикла смещений вдоль всех  $n$  осей производится уменьшение шага (обычно он делится пополам) и цикл повторяется. При попадании в достаточно малую окрестность экстремума ( $h < \varepsilon$ ) процесс поиска прекращается.

#### 6.4. ПОИСК ПО ОБРАЗЦУ (МЕТОД ХУКА-ДЖИВСА)

Метод Хука-Дживса, схема которого приведена на рис. 11, напоминает метод покоординатного спуска с той разницей, что в нем для каждой точки, называемой *базисной*, рассчитываются  $2n$  значений целевой функции (в точках на координатных осях, отстоящих от базисной на величину шага  $h$ ). Из этих точек та, в которой значение критерия качества  $K$  наименьшее, назначается новой базисной точкой. Если во всех исследуемых точках значение  $K$  больше, чем в базисной, то шаг делится пополам. И так до тех пор, пока значение шага не станет меньше

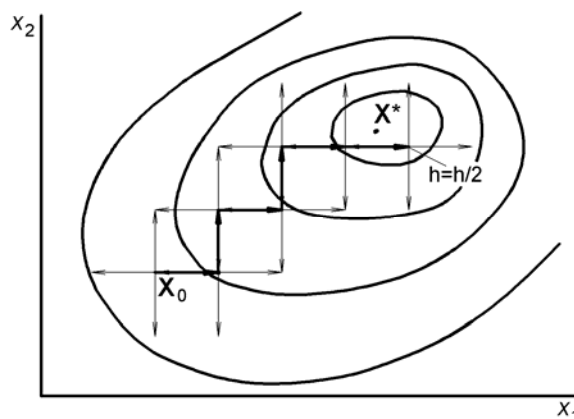


Рис. 11. Схема метода Хука-Дживса (поиска по образцу)

заданной точности. Один из вариантов алгоритма метода Хука-Дживса приведен на рис. 12.

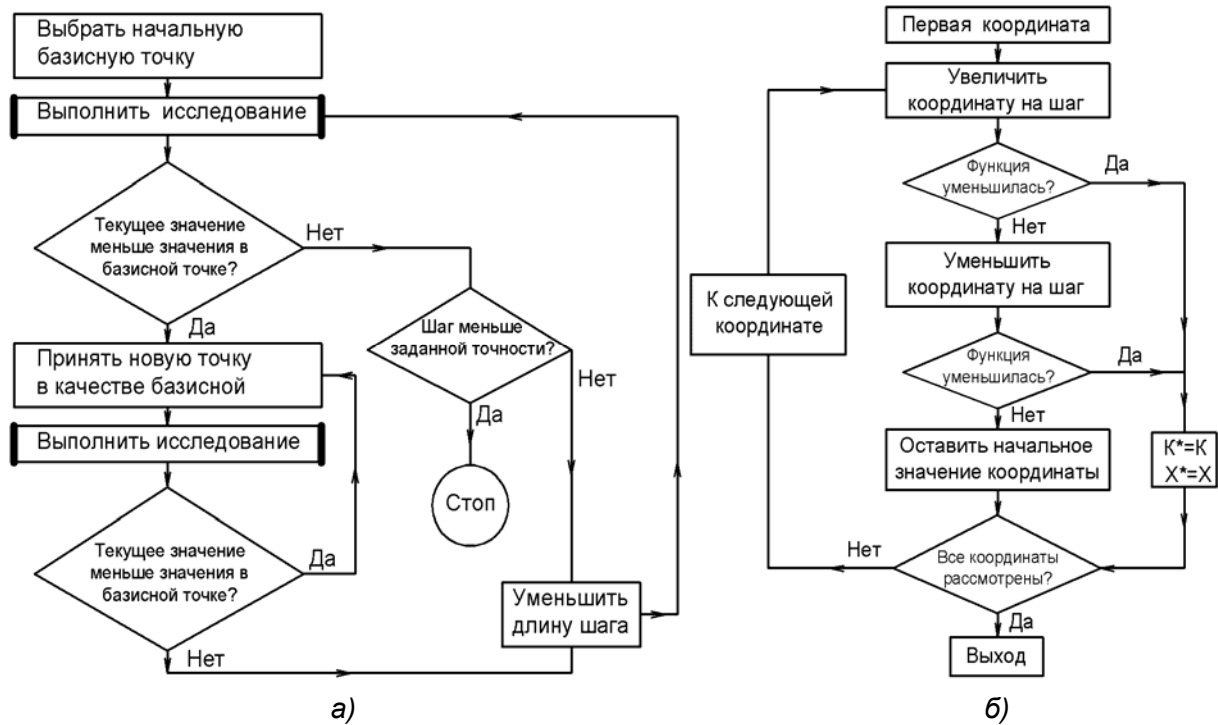


Рис. 12. Алгоритм метода Хука-Дживса.

а – общий алгоритм; б – подробное описание шага «выполнить исследование»

## 6.5. МЕТОД ГРАДИЕНТА

Известно, что направление градиента в каждой точке совпадает с направлением наибыстрейшего возрастания целевой функции, то есть в задачах максимизации локально наилучшим является градиентное направление. В задачах минимизации целевой функции перспективным направлением поиска в текущей точке является направление антиградиента, противоположное направлению градиента. Это свойство градиента и используется в данном методе. В соответствии с методом в каждой рассматриваемой точке  $X_k$  рассчитывается градиент целевой функции:

$$\text{grad } \Phi(X) = \left[ \frac{\partial \Phi(X)}{\partial x_1} \Big|_{X=X_k}, \frac{\partial \Phi(X)}{\partial x_2} \Big|_{X=X_k}, \dots, \frac{\partial \Phi(X)}{\partial x_n} \Big|_{X=X_k} \right], \quad (17)$$

а также определяется вектор  $g$  единичной длины, совпадающий по направлению с градиентным:

$$g = \frac{\text{grad } \Phi(X)}{|\text{grad } \Phi(X)|},$$

где  $|\text{grad } \Phi(X)| = \sqrt{\sum_{i=1}^n \left[ \frac{\partial \Phi(X)}{\partial x_i} \Big|_{X=X_k} \right]^2}$  — норма (длина) вектор-градиента.

Вычисление частных производных производится аналитически или приближенно по формуле

$$\frac{\partial \Phi(X)}{\partial x_i} \approx \frac{\Phi(x_1, \dots, x_i, \dots, x_n) - \Phi(x_1, \dots, x_i + \Delta x_i, \dots, x_n)}{\Delta x_i}$$

Далее осуществляется шаг в градиентном направлении в новую отображающую точку, при максимизации:

$$X_{k+1} = X_k - h \cdot g_k$$

Если  $\Phi(X_{k+1}) > \Phi(X_k)$ , то определяется  $\text{grad } \Phi(X_{k+1})$  и вектор  $g_{k+1}$  уже в точке  $X_{k+1}$  и делается новый шаг.

Если  $\Phi(X_{k+1}) < \Phi(X_k)$ , то такая ситуация отождествляется с «перешагиванием» экстремума, что требует возврата в предыдущую точку

и продолжения поиска с уменьшенным шагом  $h=h/2$ . На рис. 13 показана траектория поиска максимума некоторой функции двух управляемых параметров с помощью градиентного метода. При использовании геометрических представлений градиентное направление в рассматриваемой точке является нормальным к линиям (при  $n=3$  — к поверхностям, при  $n>3$  — к гиперповерхно-

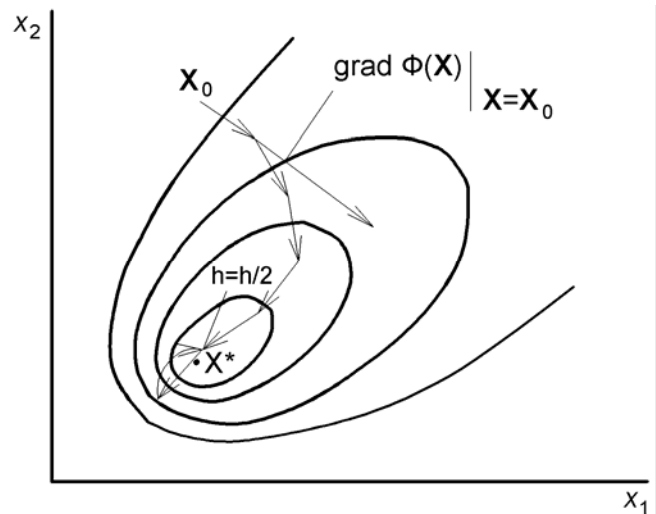


Рис. 13. Схема метода градиентов

стям) равного уровня. Именно это свойство градиента и определило вид изображенной траектории.

## 6.6. МЕТОД НАИСКОРЕЙШЕГО СПУСКА

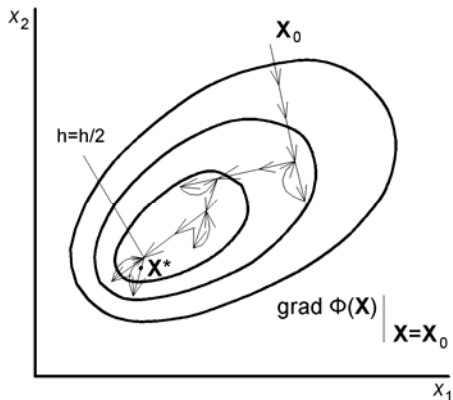


Рис. 14. Схема метода наискорейшего спуска

Разновидностью градиентного метода является метод наискорейшего спуска (при решении задач максимизации — подъема). В отличие от предыдущего градиентного метода определение градиента производится не в каждой точке. В градиентном направлении движение осуществляется с шагом  $\Delta X$  до тех пор, пока критерий качества уменьшается. При первом неудачном шаге делается возврат и определяется градиент, по которому снова начинается движение. Если первый же шаг в градиентном направлении является неудачным, производится уменьшение длины шага (рис. 14). По сравнению с методом градиента число вычислений частных производных уменьшается, зато траектория движения к экстремуму получается длиннее.

По сравнению с методом градиента число вычислений частных производных уменьшается, зато траектория движения к экстремуму получается длиннее.

## 6.7. МЕТОД МОНТЕ-КАРЛО

Существует много разновидностей алгоритмов, реализующих случайный поиск. В них определение координат точек осуществляется с помощью специальных алгоритмов — генераторов случайных чисел. В составе математической библиотеки практически любой системы программирования имеются функции для получения случайных значений величины, равномерно рас-

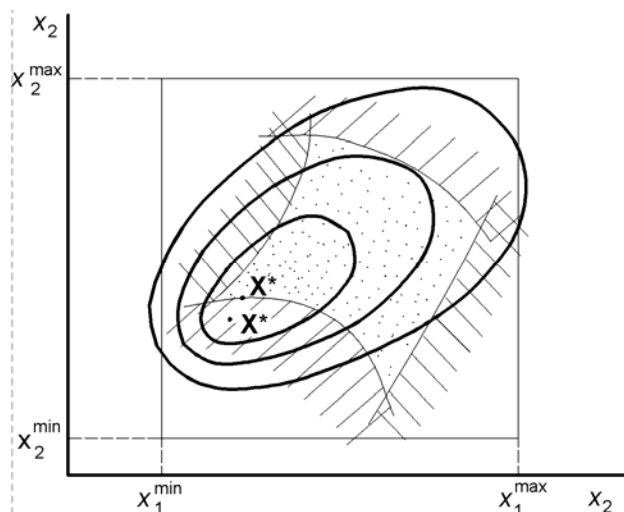


Рис. 15. Пример применения метод Монте-Карло для задачи с ограничениями.  
 $X^*$  — решение задачи безусловной оптимизации;  
 $X''^*$  — решение задачи с данными ограничениями

пределенной в интервале  $[0,1]$ . Для получения равномерно распределенной случайной величины  $x$  в интервале  $[a, b]$  используется формула

$$x = a + (b - a) \cdot \alpha,$$

где  $\alpha$  — случайная величина, равномерно распределенная на интервале  $[0,1]$ .

Значения  $x_1, x_2, \dots, x_n$  можно рассматривать, как координаты точки в  $n$ -мерном пространстве (рис. 15). После генерации случайной точки, лежащей в заданном интервале  $[X^{\min}, X^{\max}]$ , она проверяется на принадлежность к ОДЗ. При позитивном результате проверки в точке вычисляется критерий качества  $K$ . Если данная точка — первая из попавших в ОДЗ, то полученное значение  $K$  и координаты точки запоминаются в качестве оптимальных ( $K^*=K, X^*=X$ ). Иначе  $K$  сравнивается с  $K^*$ , и если вновь вычисленное значение критерия качества меньше, то теперь уже эта точка принимается в качестве оптимальной.

Преимуществом этого метода кроме простоты программной реализации является возможность прервать его в любое время с последующим возобновлением. Кроме того, метод является глобальным (сразу находит глобальный минимум) и легко решает задачи с ограничениями. Наряду с другими случайными методами он незаменим там, где размерность задачи  $n$  достаточно велика ( $n > 10$ ).

Погрешность метода для координаты  $x_i$  при  $N$  вычисленных точках может быть оценена как  $(x_i^{\max} - x_i^{\min})/N$ .

## 6.8. МЕТОД ПОИСКА ПО НАИЛУЧШЕЙ ПРОБЕ

Данный метод является дальнейшим развитием идеи метода Монте-Карло с целью уменьшить число точек, в которых будет вычислено значение целевой функции при той же точности. Схема этого метода приведена на рис. 16. В области допустимых значений берется начальная точка  $X_0$  и окружается гиперкубом (на плоскости — квадратом) со стороной, равной удвоенному шагу  $h$ . Затем внутри квадрата проводится поиск минимума аналогично методу Монте-Карло. Если полученное решение находится достаточно близко от границы гиперкуба, то есть основания полагать, что снаружи гиперкуба имеются точки с еще меньшим значением параметра критерия качества  $K$ . Тогда най-

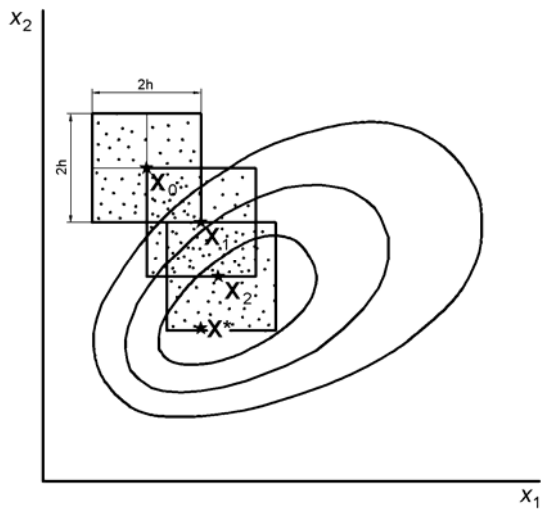


Рис. 16. Схема поиска по наилучшей пробе (поиска внутри гиперкуба)

денная оптимальная точка окружается гиперкубом и все повторяется сначала. Если оптимальная точка лежит достаточно далеко от границ гиперкуба, то она является решением задачи.

Данный метод несмотря на перекрытие гиперкубов и вытекающее отсюда повторное вычисление точек в перекрывающихся областях, в ряде случаев позволяет получить решение значительно быстрее, чем метод Монте-Карло. Но не следует забывать, что как

и все пошаговые методы, он является *локальным*, то есть обнаруживает ближайшее к нему экстремальное значение целевой функции, которое вовсе не обязательно будет являться глобальным минимумом.

## 6.9. МЕТОД ПОИСКА ПО ПОВЕРХНОСТИ ГИПЕРШАРА

Для повышения эффективности алгоритмов случайного поиска в начале движения применяют генерацию случайных векторов смещения с постоянной длиной шага. При этом случайные пробы, взятые в базисной точке, будут равномерно распределены по поверхности гипершара — гиперсфере (в случае двух управляемых параметров — по окружности). Если для каждого сгенерированного случайного вектора  $\Xi = (\xi_1, \xi_2, \dots, \xi_n)$  применить уравнение гиперсферы в  $n$ -мерном пространстве с радиусом  $h$ :

$$(a \cdot \xi_1)^2 + (a \cdot \xi_2)^2 + \dots + (a \cdot \xi_n)^2 = h^2$$

и из него определить величину

$$a = \frac{h}{\sqrt{\sum_{j=1}^n \xi_j^2}},$$

то вектор  $\Delta X = a \cdot \Xi$  будет иметь одно и тоже значение длины  $|\Delta X| = h$  при любой генерации сочетания  $\xi_1, \xi_2, \dots, \xi_n$ , и с равномерным распределением в ин-

тервале  $[-1, 1]$ .

Если в текущей базисной точке  $\mathbf{X}_k$  рассмотреть  $m$  проб вектора смещения

$$\Delta \mathbf{X}_1 = a_1 \cdot \mathbf{E}_1, \Delta \mathbf{X}_2 = a_2 \cdot \mathbf{E}_2, \dots, \Delta \mathbf{X}_m = a_m \cdot \mathbf{E}_m,$$

то все точки  $\mathbf{X}_i = \mathbf{X}_k + \Delta \mathbf{X}_i$ , где  $i = 1, 2, \dots, m$ , будут равномерно распределены по поверхности гипершара с центром в точке  $\mathbf{X}_k$  и радиусом равным  $h$ . На рис. 17 для двумерной задачи начало движения из точки  $\mathbf{X}_0$  определяется по четырем точкам, равномерно распределенным по окружности. После дачного шага снова производится выбор нового направления из 4 проб. При попадании в окрестность экстремума  $\mathbf{X}^*$ , когда все пробы неудачны (значения критерия качества  $K$  в них больше, чем в базисной точке) шаг делится пополам. Необходимая точность нахождения экстремума обеспечивается уменьшением  $h$  до допустимой погрешности  $\varepsilon$  в окрестности минимального значения функции.

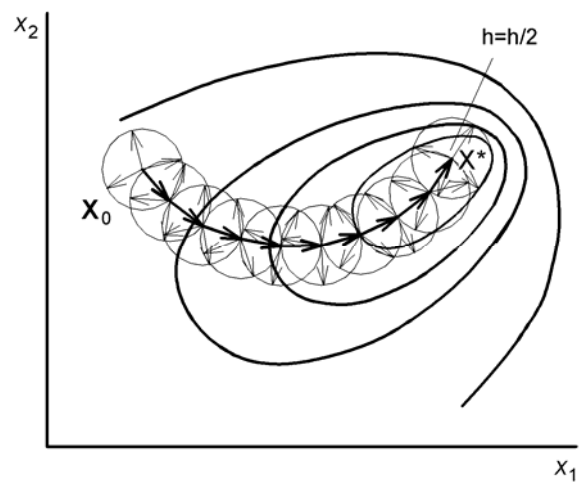


Рис. 17. Схема поиска по наилучшей пробе на поверхности гипершара

## 6.10. СРАВНЕНИЕ МЕТОДОВ ОПТИМИЗАЦИИ

Одним из важнейших показателей эффективности методов оптимизации являются потери на поиск, которые складываются из затрат на определение вектора смещения в каждой текущей точке и затрат на движение, длина траектории которого зависит от метода, вида целевой функции и положения начальной точки. Один и тот же метод может быть эффективным для одной целевой функции и неэффективным для других. На рис.18 при движении из начальной точки  $\mathbf{X}_{0,1}$  по методу Гаусса-Зейделя траектория имеет по сравнению с методом градиента меньшую длину, и наоборот — при движении из точки  $\mathbf{X}_{0,2}$ . Однако, большая длина траектории не означает наличие меньшей



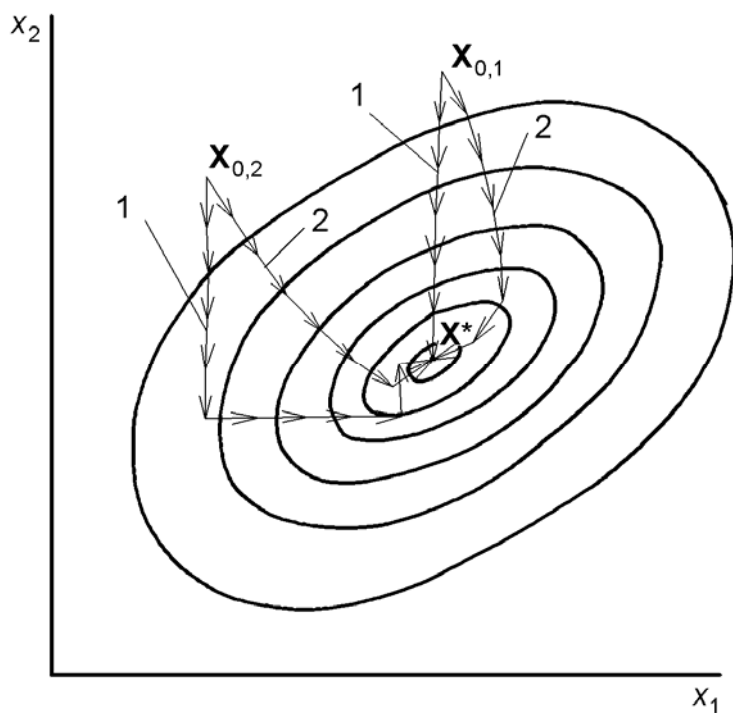


Рис. 18. Сравнение метода Гаусса-Зейделя и метода градиентов

эффективности. Так при движении из начальной точки  $x_{0,2}$  методом Гаусса-Зейделя окрестность экстремума достигнута за 11 шагов, а методом градиента — за 7. Но при поиске по методу градиента в каждой точке определялись значения частных производных, что требует для двух управляемых параметров, например по формуле (17), 3 обращения к вычислению целе-

вой функции. Общее число вычислений целевой функции для траектории 2 (методом градиента) составит 21, что превышает число обращений при движении по траектории 1, несмотря на более ее короткую длину.

Для многопараметрических целевых функций при большом числе управляемых параметров наиболее эффективными являются алгоритмы случайного поиска экстремума. По данным работы [12] затраты метода случайного поиска растут, как  $\sqrt{n}$ , где  $n$  — число управляемых параметров, в то время, как затраты метода градиента возрастают линейно, а метода сеток — в соответствии со степенной функцией с показателем степени  $n$  (рис.19).

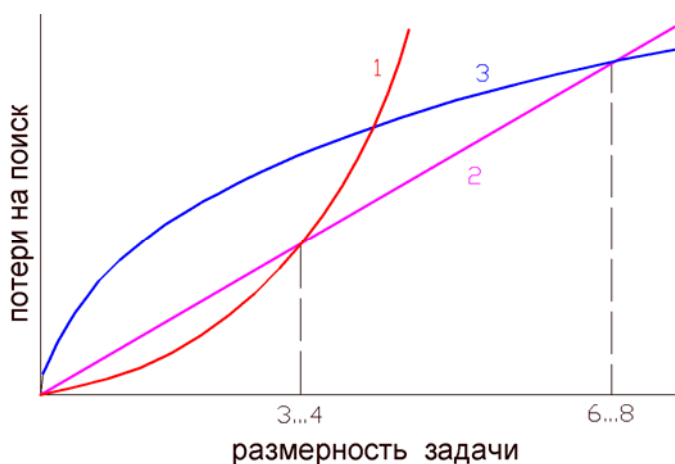


Рис. 19. Потери на поиск при различных методах оптимизации. 1— метод полного перебора (метод сеток); 2 — градиентные методы; 3 — вероятностные методы

Как правило, при  $n > 6 \dots 8$  алгоритмы случайного поиска имеют большую эффективность по сравнению с методами первого порядка. Метод сеток наиболее удобен при размерности задачи  $n < 4$ .

Важным свойством метода оптимизации является возможность движения к экстремуму вдоль линии границы области допустимых значений. Несмотря на возможность сведения задачи оптимизации с ограничениями к безусловной удобно иметь универсальный метод, не требующий проведения предварительных преобразований по сведению задачи условной оптимизации к безусловной (см. п. 6.7 метод штрафных функций). В алгоритмах случайного поиска перспективным направлением является любое, приводящее к увеличению (уменьшению) целевой функции, что и обеспечивает возможность движения вдоль границы области допустимых значений.

В других методах вводятся дополнительные приемы, обеспечивающие поиск условных экстремумов. Так для функций ограничений типа равенства применим метод проекции градиента, реализующий перемещение в гиперплоскости, касательной к гиперповерхности функции ограничений. Метод градиента применим и к решению задач с ограничениями типа неравенств, для этого вводится проверка принадлежности точки смещения к ОДЗ и если ограничения нарушаются, то делается шаг в направлении противоположном направлению проекции градиента функции нарушенных ограничений. Но это требует дополнительных затрат по определению частных производных от функций ограничений, что резко увеличивает потери на поиск.

К сожалению, отсутствуют эффективные методы, позволяющие легко установить число экстремумов у целевой функции. Зная координаты какой-либо экстремальной точки  $\mathbf{X}^*$ , нельзя сделать никаких заключений о локальном или глобальном характере этого экстремума, не исследуя  $\Phi(\mathbf{X})$  во всех точках области допустимых значений. Исключением являются задачи, где целевая функция — выпуклая при максимизации или вогнутая при минимизации. Выпуклость (или вогнутость) есть достаточный признак единственности экстремума. Существуют методы определения этого признака для аналитически заданных функций. Но в задачах проектирования технических объектов целевые функции  $\Phi(\mathbf{X})$ , как правило, заданы в алгоритмической форме, и проверка на

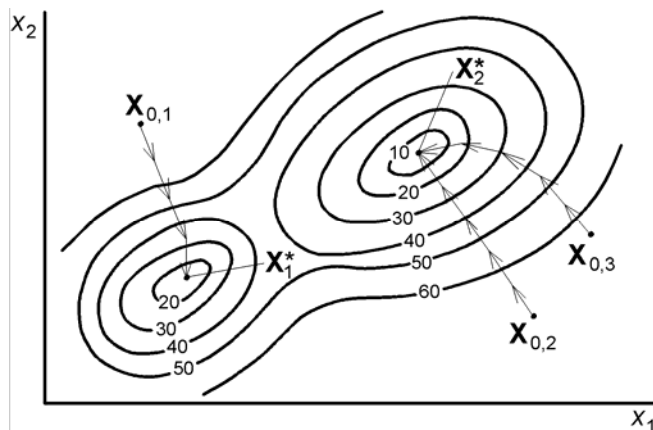


Рис. 20. Пример многоэкстремальной задачи

ние  $X_i^* = X^*$ , то с некоторой степенью уверенности можно говорить о глобальном характере экстремума. Таким образом повторное решение задачи оптимизации из нескольких положений начальной точки в пространстве оптимизации, повышает вероятность нахождения глобального экстремума. На рис. 20 для функции с двумя экстремумами поиск из начальной точки  $X_{0,1}$  приводит к решению  $X_1^*$  — локальному экстремуму, а из  $X_{0,2}$ ,  $X_{0,3}$  — к глобальному экстремуму  $X^* = X_2^*$ .

Аналогом метода полного перебора является метод Монте-Карло, реализующий в заданной области генерацию заданного количества равномерно распределенных точек. Программная реализация метода Монте-Карло намного проще по сравнению с программной реализацией метода сеток.

Кроме рассмотренных методов и алгоритмов параметрической оптимизации существует большое количество и других методов: методы второго порядка с определением вторых частных производных; методы и дополнения к рассмотренным при «овражных» целевых функциях; алгоритмы поиска с самообучением и так далее.

## 6.11. РЕШЕНИЕ ЗАДАЧ УСЛОВНОЙ ОПТИМИЗАЦИИ

Большинство методов оптимизации ориентировано на поиск безусловных локальных экстремумов. Их использование в задачах условной оптимизации возможно, так как существуют методы сведения задач условной (с ог-

выпуклость (вогнутость) невозможна.

Наиболее простой способ проверки полученного решения  $X^*$  на глобальность заключается в задании другой начальной точки  $X_{0,i}$  и проведение поиска экстремума из этой точки. Если полученное решение

раничениями) оптимизации к безусловной.

При наличии в постановке задачи оптимизации ограничений типа равенства производят уменьшение числа управляемых параметров. Для этого на каждом шаге поиска экстремума значения ряда управляемых параметров определяются как решение уравнения

$$w_j(\mathbf{X}) = 0$$

относительного одного из параметров

$$x_j = w_j^{-1}(\mathbf{X}'),$$

где  $w_j^{-1}$  — аналитическое решение уравнения функции ограничения относительно переменной  $x_j$ ;  $\mathbf{X}'$  — новый вектор управляемых параметров, не содержащий  $x_j$ .

Таким образом, целевая функция преобразуется в функцию (18) от меньшего числа управляемых параметров  $\mathbf{X}'$ , изменение значений которых при поиске может производиться независимо друг от друга.

$$K = \Phi(w_1^{-1}(\mathbf{X}'), w_2^{-1}(\mathbf{X}'), \dots, w_p^{-1}(\mathbf{X}'), \mathbf{X}'), \quad (18)$$

где  $\mathbf{X}' = (x_{p+1}, x_{p+2}, \dots, x_n)$  — новый вектор независимых управляемых параметров.

В случае невозможности получения аналитических решений используют численные методы определения вектора значений

$$\langle \mathbf{X}_j \rangle = (\langle x_{1,i} \rangle, \langle x_{2,i} \rangle, \dots, \langle x_{p,i} \rangle)$$

на каждом  $i$ -м шаге поиска экстремума. Вектор  $\langle \mathbf{X}_j \rangle$  есть решение системы уравнений (18) при фиксированных (заданных) значениях вектора независимых управляемых параметров  $\mathbf{X}_i'$  в  $i$ -ой точке пространства оптимизации.

$$\begin{cases} w_1(x_1, x_2, \dots, x_p, X_i^1) = 0 \\ w_2(x_1, x_2, \dots, x_p, X_i^1) = 0 \\ \dots \\ w_p(x_1, x_2, \dots, x_p, X_i^1) = 0 \end{cases} \quad (19)$$

Если зависимость значений вектора  $\langle \mathbf{X} \rangle$  от значений независимых управляемых параметров обозначить через функцию-оператор  $\langle \mathbf{X} \rangle = R(\mathbf{X})$ , то целевая функция будет представлять собой функцию (20) от  $n-p$  независимых параметров оптимизации.

$$K = \Phi(R(\mathbf{X}'), \mathbf{X}') \quad (20)$$

Таким образом решение задачи оптимизации с ограничениями может быть сведено к решению задачи безусловной оптимизации.

**Метод штрафных функций** сводит задачу оптимизации с ограничениями к задаче безусловной оптимизации, после чего становятся применимыми методы поиска безусловных экстремумов. Первоначальная целевая функция  $\Phi(\mathbf{X})$  заменяется на новую целевую функцию

$$K' = \Phi(\mathbf{X}) + \Theta(\mathbf{X}),$$

где  $\Theta(\mathbf{X})$  — функция штрафа, обладающая следующими свойствами:

$$\Theta(\mathbf{X}) \begin{cases} =0, & \text{если } \mathbf{X} \in \{\mathbf{X}\} \\ >0, & \text{если } \mathbf{X} \notin \{\mathbf{X}\}. \end{cases}$$

Таким образом, при нарушении хотя бы одного ограничения «налагается» штраф на целевую функцию. Примером функции штрафа может служить функция:

$$\Theta(\mathbf{X}) = \sum_{i=1}^m \gamma_i \cdot (\Psi_i(\mathbf{X}))^2,$$

$$\text{где } \gamma_i = \begin{cases} \lambda_i, & \text{если } \psi_i(\mathbf{X}) < 0 \\ 0, & \text{если } \psi_i(\mathbf{X}) \geq 0, \end{cases}$$

где  $\lambda_i$  — коэффициент штрафа за нарушение  $i$ -го ограничения.

Иллюстрацией к методу штрафных функций в одномерном случае может служить пример, показанный на рис. 21, где даны функции  $\Phi(\mathbf{X})$ ,  $\Psi(\mathbf{X})$  и новая целевая функция  $K' = \Phi(\mathbf{X}) + \Theta(\mathbf{X})$ . В интервале ОДЗ, где ограничение  $\Psi(\mathbf{X}) > 0$  выполняется, функции  $\Phi(\mathbf{X})$  и  $\Phi(\mathbf{X}) + \Theta(\mathbf{X})$  совпадают. При нарушении ограничения функция  $\Phi(\mathbf{X})$  резко

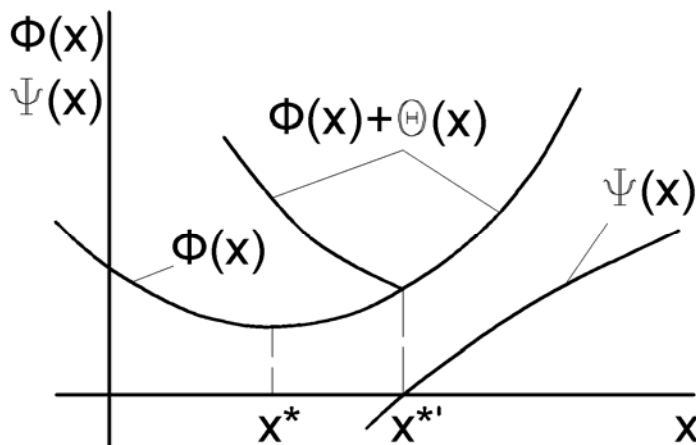


Рис. 21. Метод штрафных функций для одномерной задачи

может служить пример, показанный на рис. 21, где даны функции  $\Phi(\mathbf{X})$ ,  $\Psi(\mathbf{X})$  и новая целевая функция  $K' = \Phi(\mathbf{X}) + \Theta(\mathbf{X})$ . В интервале ОДЗ, где ограничение  $\Psi(\mathbf{X}) > 0$  выполняется, функции  $\Phi(\mathbf{X})$  и  $\Phi(\mathbf{X}) + \Theta(\mathbf{X})$  совпадают. При нарушении ограничения функция  $\Phi(\mathbf{X})$  резко

возрастает. В результате условный минимум функции  $\Phi(X)$  совпадает с безусловным минимумом функции  $\Phi(X)+\Theta(X)$  в точке  $X^*$ , которая и является решением задачи оптимизации  $\Phi(X)$ , с ограничением  $\Psi(X)>0$ .

## 6.12. ТЕСТОВЫЕ ФУНКЦИИ ДЛЯ АЛГОРИТМОВ ОПТИМИЗАЦИИ

Для проверки методов решения задач оптимизации, а также для отладки и тестирования соответствующего программного обеспечения применяются специальные функции. В отличие от реальных задач эти функции относительно просты, задаются аналитически и их решение очевидно. Кроме того, для проверки метода иногда требуются функции специального вида. Например, для проверки градиентных методов обычно применяют овражные функции, где данные методы сталкиваются с определенными трудностями (рис. 22).

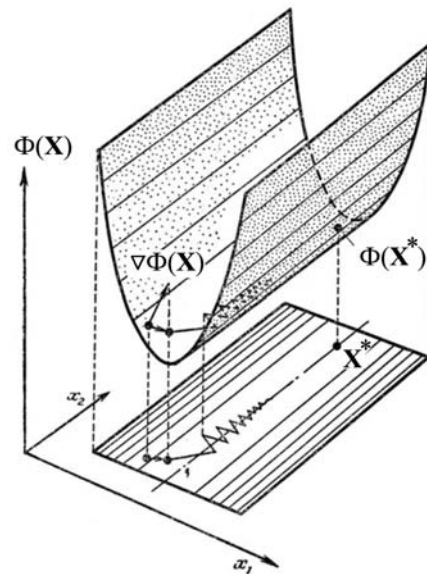


Рис. 22. Поведение градиентных методов в случае овражных функций

Наиболее часто в качестве тестовой функции используется *эллиптическая функция* (рис. 23). Для произвольного числа размерностей  $n$  она запишется в следующем виде:

$$\Phi(X) = a_0 + \sum_{i=1}^n a_i (x_i - x_i^*), \text{ где}$$

$a_0 \dots a_n$  и  $x_1^* \dots x_n^*$  — заданные константы.

Нетрудно видеть, что минимальное значение функции будет равно  $a_0$ , и достигнуто оно будет в точке  $X^* = \{x_1^* \dots x_n^*\}$ .

В качестве примера овражной функции используется двумерная *функция Розенброка* (рис. 24). Она задается формулой

$$\Phi(X) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2.$$

Минимум этой функции  $\Phi(X^*)=0$  при  $X^*=\{1,1\}$ .

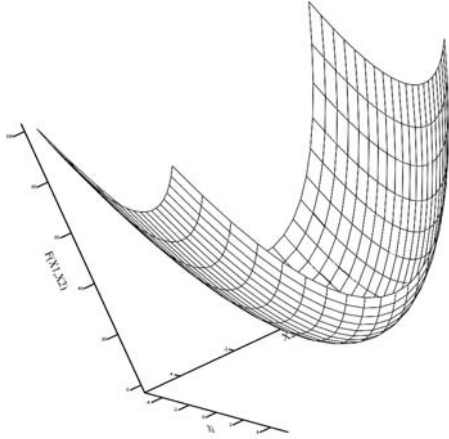


Рис. 23. Эллиптическая функция

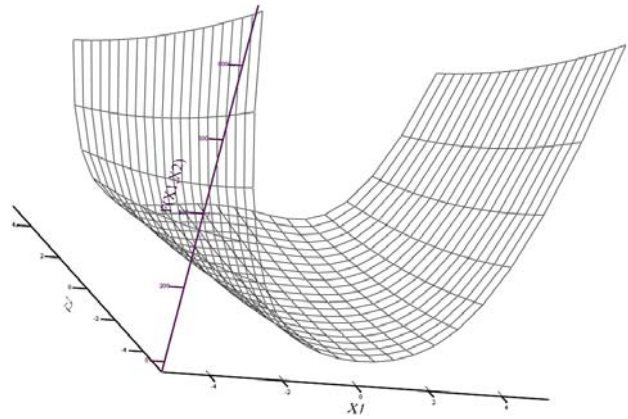


Рис. 24. Функция Розенброка

Обе приведенных функции являются *вогнутыми*, то есть имеют один экстремум. Для тестирования локальных методов применяется, в частности, четырехмерная функция Пауэлла, имеющая несколько локальных экстремумов. Эта функция записывается следующим образом

$$\Phi(X) = (x_1 + 10x_2)^2 + 5(x_2 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4,$$

её решение, при котором  $\Phi(X^*)=0$  будет наблюдаться при  $X^*=\{0,0,0,0\}$ .

Существует большое количество и других тестовых функций, познакомиться с которыми можно, в частности, на сайте компании ОРЦ по адресу [http://www.orc.ru/~pulsar/test/test\\_r.html](http://www.orc.ru/~pulsar/test/test_r.html).

## ЛИТЕРАТУРА

1. Банди Б. Методы оптимизации. Вводный курс. М.: Радио и связь, 1988. — 126 с.
2. Гардан И., Люка М. Машинная графика и автоматизация конструирования. — М.: Мир, 1987. — 272 с.
3. ГОСТ 22487-77. Проектирование автоматизированное. Термины и определения.
4. ГОСТ 34.003-90. Автоматизированные системы. Термины и определения.
5. Камаев В.А. Методы нелинейного программирования в транспортном машиностроении. Учебное пособие. — Волгоград: Изд. Волгоградского политехнического института, 1984. — 86 с.
6. Корячко В.П. и др. Теоретические основы САПР: Учебник для вузов. — М.: Энергоатомиздат, 1987. — 400 с.
7. Кос И.И., Зорин В.А. Основы надежности дорожных машин. — М., Машиностроение, 1978. — 165 с.
8. Кунву Ли. Основы САПР (CAD/CAM/CAE) СПб: Питер, 2004. — 560 с.
9. Норенков И.П. Введение в автоматизированное проектирование технических устройств и систем: Учебное пособие для вузов, 2-ое изд., перераб. и доп. — М.: Высшая школа, 1986. — 304 с.
10. Норенков И.П. Разработка систем автоматизированного проектирования. Учебник для вузов. — М: Изд-во МГТУ им. Баумана, 1994. — 207 с.
11. Пшеничный Б.Н., Данилин Ю.М. Численные методы в экстремальных задачах. — М.: Наука, 1975. — 260 с.
12. Растрингин Л.А. Теория и применение случайного поиска. — Рига: Знание, 1969. — 260 с.
13. Реклейтис Г., Рейвиндран А., Рэгсдел К. Оптимизация в технике. Кн. 1,2. М.: Мир, 1986.
14. Сироткин Я.А. Промышленные САПР машиностроения, элементы геометрического моделирования. СПб, Изд-во СПбГТУ, 2000. — 116 с.
15. Системы автоматизированного проектирования. В 9 кн. /Под ред. Норенкова И.П. — М.: Высшая школа, 1986.
16. Соболев И.М. Численные методы Монте-Карло. — М.: Наука, 1973. — 310 с.



## СОДЕРЖАНИЕ

<b>1. ОБЩИЕ ОСНОВЫ ПРОЕКТИРОВАНИЯ .....</b>	<b>3</b>
1.1. ОСНОВНЫЕ ПОНЯТИЯ .....	3
1.2. ИСПОЛЬЗОВАНИЕ ЭВМ В ПРОЦЕССЕ ПРОЕКТИРОВАНИЯ .....	8
<b>2. СОСТАВ САПР .....</b>	<b>12</b>
2.1. ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САПР .....	12
2.1.1. Вычислительные системы.....	12
2.1.2. Устройства ввода-вывода информации .....	15
2.2. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ САПР.....	17
2.3. МАТЕМАТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САПР.....	19
2.4. ПРОЧИЕ ВИДЫ ОБЕСПЕЧЕНИЯ САПР .....	20
<b>3. ГРАФИЧЕСКОЕ ОТОБРАЖЕНИЕ ОБЪЕКТОВ ПРОЕКТИРОВАНИЯ.....</b>	<b>22</b>
<b>4. МАТЕМАТИЧЕСКОЕ ОПИСАНИЕ ОБЪЕКТОВ ПРОЕКТИРОВАНИЯ.....</b>	<b>25</b>
4.1 ПОНЯТИЕ МАТЕМАТИЧЕСКОЙ МОДЕЛИ .....	25
4.2. ПАРАМЕТРЫ МАТЕМАТИЧЕСКОЙ МОДЕЛИ.....	27
4.3. ЧУВСТВИТЕЛЬНОСТЬ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ.....	29
<b>5. ПОСТАНОВКА ЗАДАЧ ОПТИМАЛЬНОГО ПРОЕКТИРОВАНИЯ... 31</b>	<b>31</b>
5.1. ОСНОВНЫЕ ПОНЯТИЯ .....	31
5.2. ПОСТАНОВКА ЗАДАЧИ ОПТИМАЛЬНОГО ПРОЕКТИРОВАНИЯ ПО ПРИНЦИПУ РАВНОПРОЧНОСТИ.....	36
5.3. ПОСТАНОВКА ЗАДАЧИ ОПТИМАЛЬНОГО ПРОЕКТИРОВАНИЯ ПО ПРИНЦИПУ РАВНОДОЛГОВЕЧНОСТИ .....	37
5.4. ПОСТАНОВКА ЗАДАЧИ ОПТИМИЗАЦИИ ПО ЧАСТНОМУ КРИТЕРИЮ .....	38

5.5. МЕТОД ВЕСОВЫХ КОЭФФИЦИЕНТОВ.....	38
5.6. МАКСИМИННАЯ И МИНИМАКСНАЯ ПОСТАНОВКИ ЗАДАЧИ ОПТИМИЗАЦИИ .....	39
5.7. ПОСТАНОВКА ЗАДАЧИ ОПТИМИЗАЦИИ ПО КРИТЕРИЮ ВЕРОЯТНОСТИ БЕЗОТКАЗНОЙ РАБОТЫ .....	40
<b>6. МЕТОДЫ РЕШЕНИЯ ЗАДАЧ ОПТИМАЛЬНОГО ПРОЕКТИРОВАНИЯ.....</b>	<b>44</b>
6.1. ОСНОВНЫЕ ПОНЯТИЯ .....	44
6.2. МЕТОД ПОЛНОГО ПЕРЕБОРА (МЕТОД СЕТОК).....	48
6.3. МЕТОД ПОКООРДИНАТНОГО СПУСКА (ГАУССА-ЗЕЙДЕЛЯ) .....	49
6.4 ПОИСК ПО ОБРАЗЦУ (МЕТОД ХУКА-ДЖИВСА).....	50
6.5. МЕТОД ГРАДИЕНТА.....	51
6.6. МЕТОД НАИСКОРЕЙШЕГО СПУСКА .....	53
6.7 МЕТОД МОНТЕ-КАРЛО.....	53
6.8. МЕТОД ПОИСКА ПО НАИЛУЧШЕЙ ПРОБЕ .....	54
6.9 МЕТОД ПОИСКА ПО ПОВЕРХНОСТИ ГИПЕРШАРА .....	55
6.10. СРАВНЕНИЕ МЕТОДОВ ОПТИМИЗАЦИИ .....	56
6.11. РЕШЕНИЕ ЗАДАЧ УСЛОВНОЙ ОПТИМИЗАЦИИ .....	59
6.12. ТЕСТОВЫЕ ФУНКЦИИ ДЛЯ АЛГОРИТМОВ ОПТИМИЗАЦИИ ...	62
<b>ЛИТЕРАТУРА .....</b>	<b>64</b>



Иван Михайлович Герасимов  
Валерий Дмитриевич Кощев

ОСНОВЫ  
АВТОМАТИЗИРОВАННОГО  
ПРОЕКТИРОВАНИЯ  
Учебное пособие

ООО «Творческий Центр «Победа»  
Лицензия ИД № 03330

Отпечатано в ООО «Творческий Центр «Победа»  
Подп. в печать 21.03.2006 Заказ № 2733  
Бумага офсетная.  
Формат 60x84 1/16. Усл. печ.л. 4,25  
Тираж 100 экз.