

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Санкт-Петербургский
государственный университет аэрокосмического приборостроения

С. А. Андронов

МЕТОДЫ
ОПТИМАЛЬНОГО
ПРОЕКТИРОВАНИЯ

Текст лекций

Санкт-Петербург
2001

УДК 681.3.06

ББК 32.965

А66

Андронов С. А.

А66 Методы оптимального проектирования: Текст лекций / СПбГУАП. СПб., 2001. 169 с.: ил.

Рассмотрены основные понятия и определения фундаментальных положений теории оптимизации, алгоритмы методов математического программирования, используемых при проектировании приборов и систем. Наряду с теоретическими рассмотрены также вопросы практического применения методов и алгоритмов оптимизации при решении проектных задач.

Текст лекций предназначен студентам специальности 2203 "Системы автоматизации проектирования" в рамках дисциплины "Оптимизация в САПР". При изложении материала значительное внимание уделено выработке у студентов навыков формализации задач, правильному выбору алгоритма решения, численного метода и, наконец, программной реализации методов на ЭВМ.

Рецензенты:

отдел автоматизации ЦНИИ КМ "Прометей";
кандидат технических наук доцент *А. Е. Шадилов*

Утверждено

редакционно-издательским советом университета
в качестве текста лекций

© Санкт-Петербургский
государственный университет
аэрокосмического приборостроения, 2001

© С. А. Андронов, 2001

Предисловие

Жизнь каждого человека заполнена альтернативами, т. е. необходимостью принимать те или иные решения. Выбор наилучшего, оптимального решения имеет определенный смысл: выбрать самое лучшее решение, допускаемое обстоятельствами.

С точки зрения процесса проектирования технических систем задача состоит в том, чтобы выбрать наиболее предпочтительный вариант создаваемой технической системы. Причем, как правило, до заключительных стадий доходят наиболее перспективные варианты, каждый из которых чем-то лучше, а чем-то хуже других. Опасность состоит в том, что лучший вариант будет отброшен, а в производство попадет менее совершенный проект изделия.

Что же мешает разработчику увеличить насколько возможно те характеристики, возрастание которых повышает потребительские качества системы и подавить все ухудшающие ее свойства. Этому препятствует взаимная зависимость между отдельными характеристиками и накладываемые ограничения, т. е. приходится идти на компромисс. Например, требование по уменьшению массы передатчика (без потери мощности) может быть обеспечено за счет увеличения площади антенны.

Методы оптимизации находят широкое применение во многих технических и экономических приложениях, а именно там, где возникают задачи принятия оптимальных решений. Это прежде всего задачи, связанные с проектированием изделий. В числе экономических задач можно назвать например, задачи расчета показателей роста производительности труда с учетом различных факторов, издержек производства при росте объема производства и пр., задачи планирования производства (в основном, методы линейного программирования (ЛП)) при ограничениях на наличные ресурсы, на производственную мощность. Программа выпуска предприятия может формироваться по различным критерияльным признакам: максимизация объема выпуска в стоимостном выражении, максимизация получаемой от реализации прибыли, минимизация издержек производства и т. д. В качестве удельных характеристик

c_i по переменным могут выступать оптовая цена единицы i -го изделия, нормативная трудоемкость его обработки и т. д. К задачам ЛП могут быть сведены задачи формирования расписаний работы поточных линий, оптимизации величин заделов, расписаний работы сборочных цехов и др.

Методы *динамического программирования* могут применяться для решения задач, где необходимо рассматривать процесс производства в пространстве или во времени. Этими методами могут решаться задачи выбора момента времени замены оборудования при условии получения за период эксплуатации наибольшей прибыли, распределения различных видов ресурсов по производствам (например, между выпуском готовых изделий и запасных частей), по различным направлениям во времени, планирования пополнения склада деталями.

Задачи календарного планирования решаются методами *теории расписаний*, дающими оптимальное (дискретное и динамическое программирование) или приближенное решение (эвристические методы). Когда приходится принимать решение в условиях неопределенности, применяются методы *теории игр*.

Проблемы оптимальной регламентации производства продукции различного вида, заготовок, степени их готовности определяют затраты на производство и хранение. Разработкой методов решения этих задач занимается *теория управления запасами*.

Методы *дискретного программирования* применяют для таких задач, как управление перевозками (транспортная) и другие распределительные задачи (о назначении, загрузке), для оптимизации обработки деталей на станках, оптимизации маршрутов следования транспорта (задача коммивояжера) и многих др.

Текст лекций предназначен в помощь студентам специальности 2203 «Системы автоматизированного проектирования», призванных автоматизировать проектный процесс. В этой связи одинаково важными для изучения являются все этапы: от постановки задачи до программной реализации. Этому вопросу посвящена отдельная глава. Наряду с теоретическими вопросами, которые касаются основных понятий и фундаментальных положений теории оптимизации, идеологии методов математического программирования, используемых при проектировании приборов и систем, внимание уделено также вопросам практического применения оптимизационных процедур.

1. ЗАДАЧА ОПТИМАЛЬНОГО ПРОЕКТИРОВАНИЯ В САПР

Идея оптимизации, стремление к оптимальным, а не к любым допустимым вариантам проектируемых систем, глубоко пронизывает современное проектирование.

Процесс проектирования можно представить в виде общей схемы (рис. 1), из которой видно, что он выполняется итеративно с оценкой и перепроектированием до тех пор, пока проект не будет удовлетворять ограничениям и целям (требованиям ТЗ) и не станет в некотором смысле оптимальным. Можно выделить три уровня оптимизации в процессе проектирования.

Первый уровень состоит в выборе наилучшей технической идеи, принципа действия ОП. Это наименее формализованный этап и задача, как правило, решается с использованием экспертных оценок. В качестве программной поддержки могут применяться соответствующие

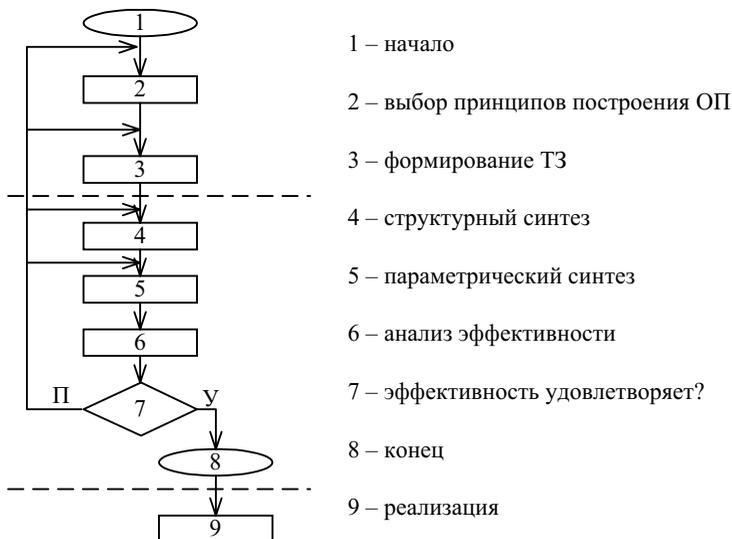


Рис. 1

экспертные системы, включающие базы готовых проектных решений. Отметим, что наряду с техническими факторами необходимо учитывать экономические показатели (прогноз стоимости, сроки изготовления).

Далее формируется первоначальный вариант технического задания (ТЗ), в котором оговариваются требования к выходным показателям ОП (область работоспособности). Заметим, что данная область в процессе проектирования может корректироваться до тех пор, пока не согласуется с возможностями синтеза. При разработке ТЗ также стремятся к оптимальности. В частности, надо найти наилучшие интервалы изменения показателей ОП, которым требуется удовлетворять. Так, например, чтобы обеспечить широкий спрос на создаваемую модель стула, необходимо установить возможно более широкий диапазон приемлемых нагрузок при его проектировании.

1.1. Понятие о структурном и параметрическом синтезе

Процесс внутреннего проектирования включает в себя:

- структурный синтез, который состоит в определении перечня типов компонентов ОП и способа их связи между собой; поиск наилучшей структуры, схемы, а затем и соответствующей им математической модели в рамках выбранного принципа действия;

- параметрический синтез, который заключается в определении числовых значений параметров (допусков на параметры) элементов в рамках заданной структуры, и условий работоспособности на выходные характеристики ОП.

Полученные проектные решения оцениваются с использованием проектных процедур анализа, на основе которых принимается решение о выходе из итерационного процесса или перепроектирования (возможно изменение принципов действия и самого ТЗ).

Кратко остановимся на методах структурной и параметрической оптимизации, используемых в САПР. Важно отметить, что для решения задачи структурного синтеза надо знать оценки качества структур ОП, которые могут быть достоверно получены в результате решения задачи параметрического синтеза. С другой стороны, задача параметрического синтеза может быть решена только при заданной структуре ОП. Выход из противоречия: структурный и параметрический синтез должны рассматриваться в едином вычислительном процессе – в диалоговой процедуре оптимального проектирования.

Задачи структурной оптимизации и методы их решения

Объект проектирования задается множеством элементов и некоторым множеством операций над элементами. Возникают следующие задачи оптимального выбора.

1. Выбор множества элементов, удовлетворяющих принципу построения ОП и требованиям ТЗ.
2. Выбор типа элементов (исходя из их наилучшего сочетания).
3. Выбор формы взаимодействия элементов в ОП (исходя из наличия связей, различной физической природы между элементами).

Возможные пути решения этих задач:

- полный перебор (учитывая трудоемкость оценки эффективности перебора всех комбинаций сочетания элементов) – неприемлем;
- сокращенный перебор (используются методы случайного поиска, однако, здесь неясно когда остановится, поскольку случайный поиск неуправляем);
- экспертные оценки.

Один из вариантов этого подхода состоит в следующем. Применяют обход древовидных структур вида И-ИЛИ деревьев и обработке морфологических таблиц (табл. 1.1 и 1.2).

Таблица 1.1

И	ИЛИ			
	Упругий элемент	Горсион	Пружина	...
	Демпф-элемент	Гидравлический	Фрикционный	...
		

Эксперт назначает коэффициенты предпочтения в табл. 1.2.

Таблица 1.2

	K_1	K_2	...
K_1	K_{11}	K_{12}	...
K_2	K_{21}	K_{22}	K_{23}
K_3	K_{31}	K_{32}	...

Далее, производится полный перебор по первым двум строкам таблицы и выбор оптимального сочетания, например $K_{11}K_{23}$. Этот вариант комбинируется с максимальным коэффициентом из следующей строки и т. д.: формальный подход, учитывающий комбинаторный характер структурного синтеза.

Результат достигается на основе использования методов дискретной оптимизации, которые предполагают, что варьируемые компоненты структуры заданы на дискретном множестве.

Применение методов дискретного программирования (таких как, например, задача о коммивояжере, о покрытии, о назначении и т. д.) связано с высокой вычислительной сложностью переборных задач. Получение точного решения неэффективно, так как трудоемкость поиска экспоненциально растет с размерностью (поэтому применяют приближенные алгоритмы). В качестве иллюстрации рассмотрим следующий пример.

Задача о коммивояжере (разъездном торговце от частной фирмы)

Коммивояжер должен посетить некоторое множество городов (рис. 2, *a* и *b*). Задача состоит в том, чтобы найти кратчайший маршрут, следуя которому он может попасть во все города не более одного раза и вернуться в исходную точку (город *a* – исходный пункт).

При лобовом способе решения этой задачи вначале генерируют все возможные перестановки городов. Если получается комбинация с городами, которые не имеют прямых путей между собой (в некоторые попадем дважды или более раз), тогда устраним эти комбинации и среди оставшихся найдем кратчайший маршрут. Сложность этого алгоритма $O(n!)$.

С целью уменьшения сложности алгоритма стремятся найти приближенное решение, а именно, в данной задаче: начиная с *a*, следующим берем ближайший город, т. е. *b*, а затем *e*. Далее, если выберем *d*, то маршрут найти нельзя. Иначе приходим к оптимальному решению, в котором длина пути равна 16 (необязательно оптимальное). Такой алгоритм имеет уже полиномиальную сложность.

Задачи параметрической оптимизации

К задачам параметрической оптимизации относятся следующие основные задачи:

- определение оптимальных значений параметров;
- назначение оптимальных допусков на параметры по математической модели и заданным ограничениям на показатели качества;
- параметрическая идентификация (уточнение параметров в модели блока объекта проектирования на основе данных испытания).

1.2. Примеры постановок задач параметрической оптимизации

1. Пусть требуется спроектировать контейнер в форме прямоугольного параллелепипеда объемом $V = 1 \text{ м}^3$ и израсходовать минимум материала. При постоянной толщине стенок площадь должна быть минимальна, т. е. имеем задачу

$$S = 2(x_1x_2 + x_2x_3 + x_1x_3) \rightarrow \min_{x_1x_2x_3}.$$

Из ограничения-равенства можем исключить одну переменную

$$V = x_1x_2x_3 = 1; \quad x_2 = \frac{1}{x_1x_3}; \quad S = 2\left(x_1x_2 + \frac{1}{x_1} + \frac{1}{x_3}\right).$$

В результате решения системы нормальных уравнений получим

$$x_1^* = x_2^* = 1 \text{ м}; \quad x_3^* = 1 \text{ м}.$$

(т. е. оптимальной формой является куб).

Можно усложнить задачу, например, чтобы контейнер был не менее 2 м длиной (тогда появится ограничение: $x_1 \geq 2$).

2. Запуск ракеты (пример решения задачи синтеза системы управления)

$$\int_0^T |n(t)| dt \rightarrow \min_n.$$

Ограничения: при $\ddot{y}(t) + mg = n(t), t[0, T]$

$$|n(t)| \leq b, \quad y(T) = \bar{y}, \quad y(0) = 0,$$

где \bar{y} – высота, на которую должна подняться ракета за время T ; $n(t)$ – сила, действующая на ракету в вертикальном направлении.

Приведем две возможных постановки задачи: определить минимальную энергию, которую надо затратить на выведение ракеты на высоту $y(T)$ за счет выбора управления $n(t)$ и соответствующей ему траектории $y(t)$

$$f(y, n) \rightarrow \min_k;$$

$$\sum_{k=1}^K |n_k| \rightarrow \min_k.$$

Если разбить отрезок $[0, T]$ на k интервалов длины d , то можно аппроксимировать задачу при условии

$$y_{1,k} - y_{1,k-1} = y_{2,k-1}, \quad k = \overline{1, K};$$

$$y_{2,k} - y_{2,k-1} = u_k - mg; \quad |u_k| \leq b; \quad y_{1,0} = y_{1,2} = 0; \quad y_{1,k} = \overline{y};$$

и найти такое управление $n(t)$, при котором высота подъема y будет максимальной

$$f(y, n) = -\overline{y} \rightarrow \min_n (" - ", \text{ так как } f = -\max(-f)).$$

3. Параметрическая идентификация:

Рассмотрим уравнение маятника

$$m l^2 \ddot{\alpha} + k \dot{\alpha} + c \alpha = -mgl \sin(\alpha).$$

Приводя к нормальной форме имеем

$$y'_1 = \dot{\alpha} = y_2;$$

$$y'_2 = f(m, l, k, c).$$

Пусть сняты экспериментальные данные (рис. 2, а). Задача в том, чтобы оценить неизвестные параметры на основе данных эксперимента.

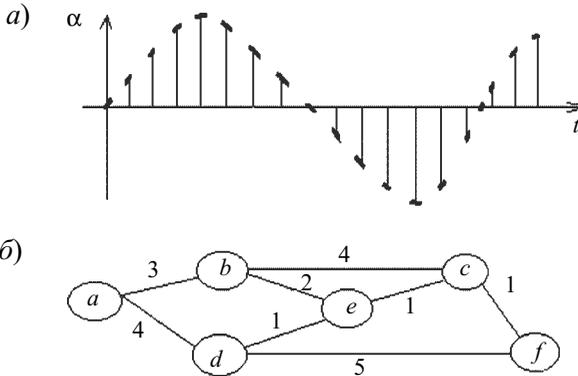


Рис. 2

Пусть m, l заданы, а k и c – неизвестны. Формально должна быть решена задача

$$J = \sum_{i=1}^N (y_1(t_i, k, c) - \tilde{\alpha}_i)^2 \rightarrow \min_{k, c}.$$

Заметим, что последняя задача решается при наличии экспериментальных данных, в частности, в условиях существования опытных готовых образцов ОП, которые подвергаются испытаниям. Понятно, что если в результате удастся уточнить модель ОП, то на модели можно поставить такие эксперименты, которые трудно или невозможно реализовать на опытном образце.

Рассмотрим подробнее последовательность выполнения проектных процедур, связанных с оптимальным параметрическим синтезом при проектировании приборов (рис. 3). Рассмотрим комментарии к приведенной схеме: блоки:

- 1 – начало;
 - 2 – составление ТЗ;
 - 3 – выбор структуры неизменяемой части прибора;
 - 4 – построение априорной математической модели (аналитические выражения, значения параметров);
 - 5 – моделирование необходимо ?
 - 6 – моделирование и анализ;
 - 7 – необходима ли корректирующее устройство (КУ)?;
 - 8 – синтез КУ (структурная и параметрическая оптимизации);
 - 9 – синтез КУ возможен ?;
 - 10 – нужен анализ параметрической чувствительности (выбор основных параметров)?;
 - 11 – анализ чувствительности;
 - 12 – оптимизация параметров;
 - 13 – найдено оптимальное решение?;
 - 14 – все блоки прибора изготовлены;
 - 15 – определение показателя эффективности (ПЭ) всего прибора;
 - 16 – показатель эффективности (ПЭ) соответствует ТЗ?
 - 17 – конец;
 - 18 – разработка и изготовление блоков прибора;
 - 19 – определение ПЭ блока;
 - 20 – ПЭ соответствует ТЗ?
 - 21 – параметрическая идентификация.
- Под ПЭ понимается

$$\text{ПЭ} = \frac{\phi}{S} = \frac{\text{Функциональная эффективность}}{\text{Стоимость}}.$$

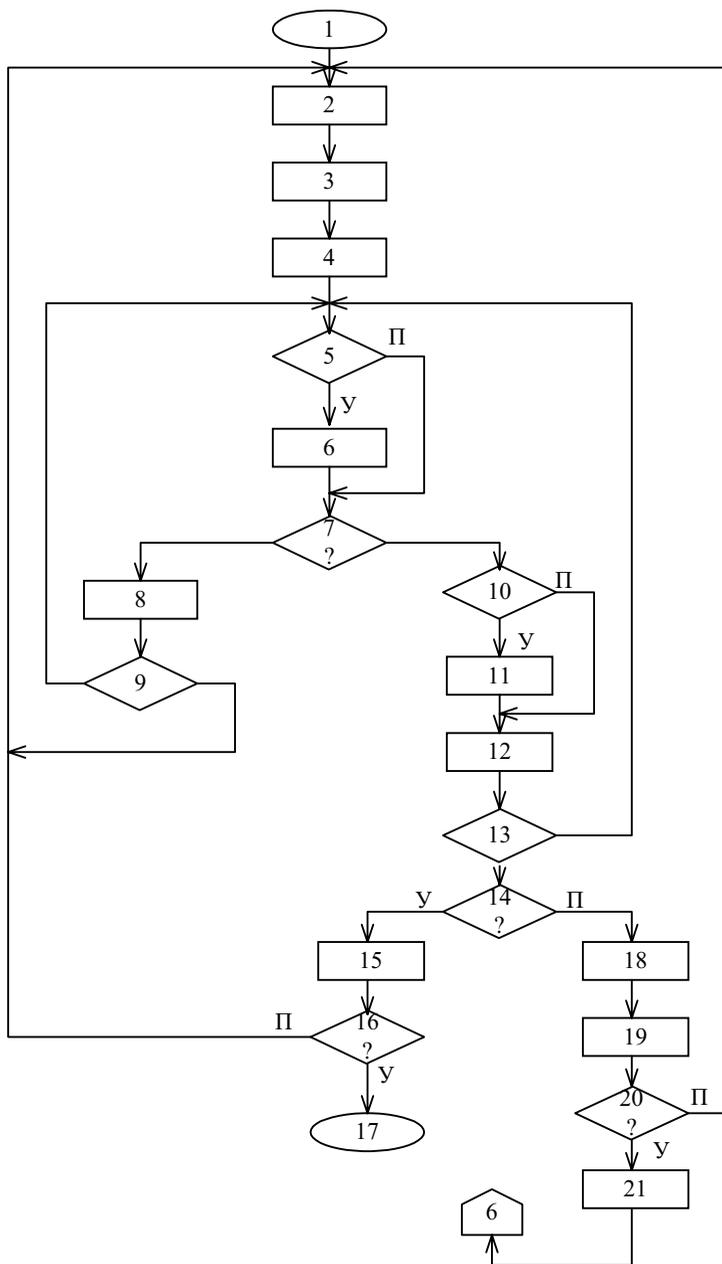


Рис. 3

1.3. Формализация процесса принятия оптимальных решений. Математическая модель ОП

Разработчик обычно имеет дело с построением двух типов моделей ОП:

1. Функционально-структурной (состав и логика соединений компонент ОП). Основой для создания такой модели являются принципиальная схема, конфигурация, которые будем считать выбранными на более ранних этапах;

2. Математической моделью (ММ), которая также исходит из структурных описаний, существенных свойств, соответствующих целям проектирования. Здесь следует особо отметить слово “существенных”, поскольку чрезмерное усложнение ММ за счет учета в ней несущественных для конкретной цели исследования факторов приводит не только неоправданным вычислительным затратам, но и к невозможности трактовки результатов.

Свойства объекта, которые могут варьироваться, называются управляемыми параметрами. Эти параметры объединяются в вектор

$$\mathbf{x} = (x_1, \dots, x_n)^T,$$

(“управляемые”, так как можем их варьировать, они влияют на характеристики). Остальные параметры могут быть постоянными или случайными величинами. Вектор \mathbf{x} характеризует систему с точки зрения разработчика ОП.

Свойства внешней среды, влияющие на ОП называются внешними параметрами. Внешние параметры, имеющие в общем случае, случайную природу, сведем в вектор

$$\xi = (\xi_1, \dots, \xi_l)^T,$$

например, шумовые воздействия, температура и т. д.

Свойства, характеризующие количественные значения показателей ОП, называются характеристиками

$$\varphi = (\varphi_1, \dots, \varphi_m).$$

В качестве примера можно назвать такие параметры, как потребляемая мощность, габариты, стоимость, точность и т. д. Их можно измерять, но непосредственно изменять нельзя (например, это угол отклонения маятника от вертикали в процессе колебаний). Вектор φ характеризует систему с точки зрения заказчика.

Под ММ ОП будем понимать отображение между двумя множествами параметров

$$\{x, \xi\} \text{ и } \{\Phi\},$$

которое может быть задано различными способами (аналитически, экспериментальная зависимость, алгоритмически и т. д.), в частности, это функциональные соотношения:

$$\begin{aligned}\Phi_1 &= \Phi_1(x_1, \dots, x_n; \xi_1, \dots, \xi_l); \\ \Phi_m &= \Phi_m(x_1, \dots, x_n; \xi_1, \dots, \xi_l).\end{aligned}$$

Заметим, что для каждого ОП можно построить несколько ММ различной степени детализации в зависимости от цели (задачи) проектирования, отражающие различные свойства, существенные в данной конкретной ситуации. Поскольку решения в САПР принимаются на основе ММ, адекватность модели реальному ОП определяет достоверность результатов, полученных в процессе проектирования.

1.4. Формализация технико-эксплуатационных требований, предъявляемых к объекту проектирования

Требования, вытекающие из ТЗ, условия физической и схемной реализуемости, условия функционирования и другие условия могут быть заданы в виде

$$\begin{aligned}x_j^- &\leq x_j \leq x_j^+, j = \overline{1, n}; \\ x_j^-(x_k) &\leq x_j \leq x_j^+(x_k), j \neq k; \\ \Phi_i^- &\leq \Phi_i(x) \leq \Phi_i^+, i = \overline{1, m}.\end{aligned}$$

Последняя группа неравенств записывается также в виде

$$g(x) \geq 0,$$

где

$$g_i(x) = \begin{cases} \Phi_i(x) - \Phi_i^-, & \Phi_i^- \leq \Phi_i(x); \\ \Phi_i^+ - \Phi_i(x), & \Phi_i(x) \leq \Phi_i^+. \end{cases}$$

Ограничение типа равенства $g(x) = 0$, если необходимо, могут быть включены в состав неравенств в виде

$$\begin{cases} g_k(x) \geq 0; \\ -g_k(x) \geq 0. \end{cases}$$

С другой стороны, неравенство $\varphi_i(x) \leq \varphi_i^+$ может быть преобразовано к равенству введением дополнительной переменной $g_i(x) = \varphi_i(x) - \varphi_i^+ + z_i^2 = 0$.

В процессе проектирования представляют интерес только те значения вектора x , которые принадлежат множеству D (допустимая область, область работоспособности)

$$\begin{aligned} D &= D_x \cap D_y; \\ D_x &= \{x \mid x_i^- \leq x_j \leq x_i^+, j = \overline{1, n}\}; \\ D_y &= \{x \mid y_i(x) \geq 0, j = \overline{1, m}\}. \end{aligned}$$

Любой вектор $x \in D$ – работоспособный вариант ОП. Запишем некоторые определения, важные при описании области D .

1. Выпуклость множества D .

Множество точек, образующих область допустимых решений D , называется выпуклым множеством, если для любой пары точек $x^{(1)}, x^{(2)} \in D$ отрезок прямой, соединяющий их $x = \alpha x^{(1)} + (1 - \alpha)x^{(2)}$, также принадлежит области D (рис. 4, а, б), т. е., если для любых $x^{(1)}$ и $x^{(2)} \in D$ любая точка отрезка прямой принадлежит D , то D – выпуклая область.

2. Многосвязное множество.

Множество, которое состоит из нескольких частей (рис. 4, в) называется многосвязным

$$\begin{cases} y_1(x) = -0,25 + x_2 - 1 \geq 0; \\ y_2(x) = -x_2 + x_1^2 - 4x_1 + 4 \geq 0, \text{ при } x_1 \geq 0, x_2 \geq 0. \end{cases}$$

Если в систему ограничений входят характеристики g_i , зависящие от некоторого параметра n (время, температура и т. д.), т. е. $g_i(x, v) \geq 0$ для любого $v \in [v^-, v^+]$, то после разбиения равномерной сетью $v^- = v_1 < \dots < v_s = v^+$ приходим к задаче

$$g_i(x) = \min_{1 \leq k \leq s} g_i(x, v_k) \geq 0,$$

т. е. $g_i(x)$ не зависит от v .

1.5. Математические модели принятия оптимальных решений

1. Пусть объект проектирования описывается с помощью детерминированной ММ вида $\Phi = \Phi(x)$. Если в области D имеется только одно значение вектора параметров x , то проблемы принятия решения не возникают (например, если $x = \text{const}$). В тех случаях, когда работоспособный вариант не единственный, для сравнения вариантов и выбора из них наилучшего (в некотором смысле) необходимо ввести функции цели (критерий оптимальности $f = f(\Phi(x)) = f(x)$ – функционал), например

$$f(x) = \int_a^b x(t) dt,$$

Экстремальное значение целевой функции численно характеризует свойство технико-экономического показателя ОП. Этот критерий показывает относительное предпочтение одного варианта по отношению к другому, определяет цель проектирования и вместе с вектором x и областью D образует математическую модель принятия оптимального решения, которое является задачей параметрической оптимизации $f(x)^* = \min_{x \in D} f(x)$ (в качестве f часто используются такие характеристики, как масса, мощность, стоимость перевозок грузов, прибыль и т. д.).

Таким образом, решение задачи сводится к выбору управляемых параметров x^* , принадлежащих допустимой области и доставляющих минимум критерию оптимальности $f(x)$ (заметим, что $\max f = -\min(-f)$).

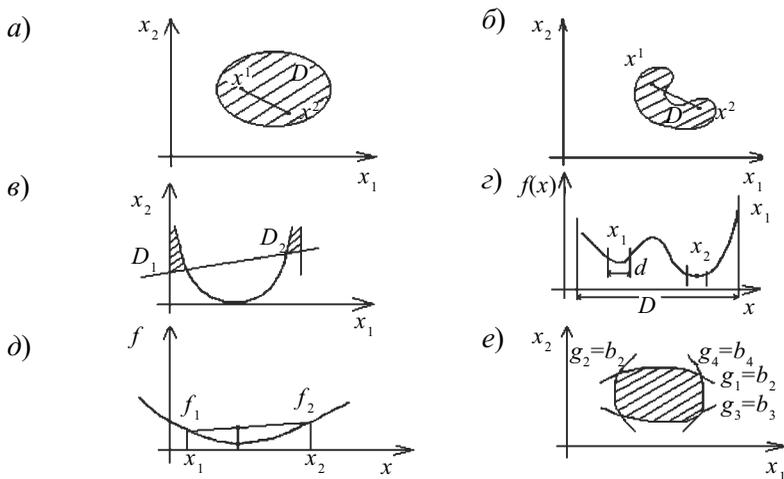


Рис. 4

2. В зависимости от вида $f(x)$ оптимальное решение x^* может быть точкой локального или глобального минимума.

Вектор x^* называется точкой локального (относительного) минимума, если для всех точек x , принадлежащих ε -окрестности $d(x^*, \varepsilon)$ этой точки, функция $f(x)$ не принимает меньшего значения, т. е. $f(x^*) \leq f(x)$ для всех $x \in d(x^*, \varepsilon)$.

В случае глобального минимума (абсолютного) $f(x^*) \leq f(x)$ для всех $x \in D$, т. е. глобальный минимум – это наименьший из всех локальных (рис. 4, з).

3. Если необходимо получить наилучшие значения одновременно для нескольких критериев, рассматривают векторный критерий оптимальности $f(x) = (f_1(x), \dots, f_s(x))$ (здесь s целей проектирования). Задача векторной оптимизации

$$f(x^*) = \min_{x \in D} f(x) = \min_{x \in D} f_1(x); \quad \min_{x \in D} f_2(x), \dots, \min_{x \in D} f_s(x).$$

Оптимальное решение этой задачи, в общем случае, не является точкой минимума ни для одного из частных критериев (как правило, f_i противоречивы). Оптимальное решение выбирается так, чтобы обеспечить компромисс между частными критериями, так как уменьшение одних из них приводит к увеличению других. Вектор x^* в этом случае называется рациональным решением. Один из путей решения этой задачи состоит в сведении ее к однокритериальной.

4. При рассмотрении вероятностной модели ОП (т. е. присутствуют случайные внешние факторы ξ) модель принятия решения нуждается в уточнении (например, технологический разброс допусков на параметры). Если у вектора относительно ξ известно, что он принадлежит некоторой области D_ξ (случай неопределенности), тогда

$$\min_{x \in D} f(x) = \min_{x \in D} \left\{ \max_{\xi \in D_\xi} f(x, \xi) \right\},$$

где $D = \left\{ x \mid g_i(x) = \left(\min_{\xi \in D_\xi} g_i(x, \xi) \right) \geq 0, \quad i = \overline{1, n} \right\}$, т. е. избавляемся от случайности. Задача состоит в получении наилучшего результата по x в наихудшем случае по неопределенности ξ . Заметим, что сужение области ограничений за счет ξ может только увеличить минимум функции.

1.6. Классификация экстремальных задач, описывающих процесс принятия оптимальных решений

В зависимости от числа n управляемых параметров x , структуры области допустимых решений D и вида критерия оптимальности $f(x)$, задача оптимизации приводится к различным классам экстремальных задач.

1. Если $n = 1$ – одномерная задача.

$$f(x) \rightarrow \min_{a \leq x \leq b} \text{ (поиск минимума заданной кривой на интервале).}$$

Если $n \geq 2$ – многопараметрическая задача.

2. Если $f(x)$ имеет в области D единственный локальный минимум, то задача $f(x) \rightarrow \min_{a \leq x \leq b}$ называется унимодальной (одноэкстремальной)

задачей оптимизации, в противном случае – многоэкстремальной задачей.

3. При отсутствии ограничений на управляемые параметры x и характеристики $\varphi(x)$ приходим к задаче безусловной оптимизации $\min_{x \in R^n} f(x)$.

Многопараметрическая задача оптимизации $f(x) \rightarrow \min_{x \in D}$ с ограничениями типа линейных неравенств, где

$$D_x = \{x \mid x_j^- \leq x_j \leq x_j^+, j = \overline{1, n}\},$$

может быть сведена к задаче безусловной оптимизации относительно переменных $z_j, j = \overline{1, n}$ с помощью преобразования

$$x_j = x_j^- + (x_j^+ - x_j^-) \sin^2 z_j, \quad j = \overline{1, n}.$$

При наличии нелинейных ограничений задача параметрической оптимизации называется задачей нелинейного программирования (в смысле планирования)

$$\min_x f(x), \text{ при } g_i(x) \geq 0, i = \overline{1, m}.$$

В некоторых частных случаях такую задачу удастся свести к задаче безусловной оптимизации с помощью функции преобразования $x_i = r_i(z_i)$ (z_i – новые переменные). Например, если область

$$D = \left\{ x \mid \sum_{i=1}^n x_i^2 = 1 \right\}, \text{ то } x_i = r_i(z) = \frac{z_i}{\alpha}, i = \overline{1, n-1}, x_n = r_n(z) = \frac{1}{\alpha},$$

где $\alpha = \left(1 + \sum_{i=1}^{n-1} z_i^2\right)^{-\frac{1}{2}}$.

Исходную задачу сведем к задаче безусловной оптимизации, решение которой находится проще (ряд преобразований можно найти в [4]).

Частным случаем задачи нелинейного программирования является задача с ограничениями типа равенств

$$\min_x f(x), \text{ при } g_i(x) = b_i, \quad i = \overline{1, m} < n.$$

В задаче нелинейного программирования тип оптимального решения (это точка локального или глобального минимума) зависит не только от вида $f(x)$, но и от того, является ли D выпуклым множеством.

4. Задача нелинейного программирования, связанная с минимизацией выпуклой функции $f(x)$, которая задана на выпуклом множестве D , называется задачей выпуклого программирования. Оптимальное решение x^* задачи выпуклого программирования задается в локальном минимуме, который является в то же время глобальным минимумом.

Функция $f(x)$ выпуклая на выпуклой области D , если для любых двух точек $(x_1, x_2) \in D$ выполняется соотношение $f(\alpha x_2 + (1 - \alpha)x_1) \leq \alpha f(x_2) + (1 - \alpha)f(x_1)$.

В частности, в случае функции одной переменной функция выпукла (вниз), если лежит ниже хорды, соединяющей любые точки графика (рис. 4, д).

Доказано, что если допустимая область задана неравенствами $g_i(x) \leq b_i$ и $g_i(x)$, $i = \overline{1, m}$ – выпуклые функции, то допустимая область выпуклая (рис. 4, е). Пересечение выпуклых множеств есть выпуклое множество (докажите!).

5. Задача выпуклого программирования (функция выпукла на выпуклом множестве) называется задачей линейного программирования (ЛП), если критерий и ограничения – линейные формы от управляемых параметров x

$$\min_x \left\{ \sum_{i=1}^n c_i x_i \right\}, \text{ при } \sum_{i=1}^n a_{ki} x_i \leq b_k, \quad x_i \geq 0, \quad k = \overline{1, m}, \quad i = \overline{1, n}$$

В задаче ЛП область D всегда выпукла (докажите!).

Обобщением задачи ЛП является задача сепарабельного программирования, в которой критерий оптимальности и ограничения имеют вид сумм из n одномерных функций только одного переменного

$$\min_x \sum_{i=1}^n r_i(x_i), \text{ при } \sum_{i=1}^n h_{ki}(x_i) \leq b_k, \quad x_i \geq 0; \quad k = \overline{1, m}; \quad i = \overline{1, n}.$$

В ряде случаев можно осуществить переход от задачи НП к задаче сепарабельного программирования.

6. Задача параметрической оптимизации с дополнительным требованием, чтобы управляемые параметры x принимали только дискретные значения, называется задачей дискретной оптимизации (D – дискретное множество). Если все управляемые параметры x , $i = \overline{1, n}$ могут быть только целыми неотрицательными числами, то задача дискретной оптимизации называется задачей целочисленного программирования. В качестве примера приведем задачу о ранце.

Задача целочисленного линейного программирования с булевыми переменными (0 или 1) называется задачей о рюкзаке (о загрузке), если

$$f(x) = \left\{ \sum_{i=1}^n c_i x_i \right\} \rightarrow \max_x \text{ при } \sum_{i=1}^n a_i x_i \leq b, \quad i = \overline{1, n}; \quad x_i = 0 \text{ или } 1.$$

Трактовка может быть, например, такая. Рюкзак загружается предметами n различных типов. Предмет i -го типа характеризуется весом a и стоимостью (полезностью) c . Максимальная грузоподъемность рюкзака (вес) b . Требуется загрузить рюкзак предметами так, чтобы максимизировать функцию полезности f и не превысить допустимую грузоподъемность. В данном случае, если $x_i = 0$, то нет предметов i -го типа, $x_i = 1$ – есть.

1.7. Примеры схем оптимального параметрического синтеза

1. Оптимизация характеристик ОП

$$f_1^{(1)}(x) = \varphi(x) \rightarrow \min_{x \in D}, \quad (1.1)$$

где в качестве характеристик могут выступать, например, потребляемая мощность, частота среза и др.

$$f_1^{(2)}(x) = \sum_i^N \varphi(x, p_i) \rightarrow \min_{x \in D}, \quad (1.2)$$

где p_i – параметр (время, частота, температура).

2. Задача аппроксимации.

Отображение степени близости оптимизируемой величины к некоторой желаемой (в частности, к экспериментальной)

$$f_2^{(1)}(x) = (\varphi(x) - \varphi_{ж})^2 \rightarrow \min_{x \in D}; \quad (1.3)$$

$$f_2^{(2)}(x) = \frac{1}{N} \sum_{i=1}^N \omega_i (\varphi(x, p_i) - \varphi_{ж})^2 \rightarrow \min_{x \in D}, \quad (1.4)$$

где ω_i – весовые коэффициенты (рис. 5, а).

3. Принадлежность некоторому коридору

$$f_3^{(1)}(x) = \begin{cases} 0, & \varphi_{ж}^H \leq \varphi_B(x) \leq \varphi_{ж}^B, \\ (\varphi(x) - \varphi_{ж}^H)^2, & \varphi(x) > \varphi_{ж}^H \rightarrow \min_{x \in D}, \\ (\varphi_{ж}^H - \varphi(x))^2, & \varphi(x) < \varphi_{ж}^H, \end{cases} \quad (1.5)$$

$$f(x) = \max_{i=1, N} \omega_i |\varphi(x, p_i) - \varphi_{iж}| \rightarrow \min_{x \in D} \quad (\text{рис. 5, б}). \quad (1.6)$$

Заметим, что функционалы типа (1.3), (1.4) обладают свойством “гладкости”. Если $\varphi(x, p)$ дважды непрерывно дифференцируема по x , то этим же свойством обладает и зависимость $f(x)$, что существенно облегчает процедуру оптимизации.

Ограниченная точность аппроксимации отдельных слагаемых, т. е. плохая аппроксимация в некоторых точках по i , может компенсироваться хорошей точностью в других точках. Этот недостаток устранен в критерии (1.6), который, к сожалению, не сохраняет гладкости функции $\varphi(x, p)$ (сначала ищем максимум, т. е. не можем дифференцировать по x), что требует привлечения специальных методов оптимизации. Поэтому часто используют гладкие среднестепенные аппроксимации минимаксного критерия $f_4(x)$

$$f_5(x) = \sum_{i=1}^N \varphi_i^v(x) \rightarrow \min_{x \in D}, \quad v = 2, 3, \dots, \quad (1.7)$$

где $\varphi_i(x) = \omega_i |\varphi(x, p) - \varphi_{im}|$.

При достаточно больших значениях v решения задач (1.6) и (1.7) почти совпадают, так как справедливо

$$\left(\sum_{i=1}^N \varphi_i^v \right)^{\frac{1}{v}} \rightarrow \max \varphi_i, \quad \varphi_i \geq 0.$$

Операция извлечения корня не влияет на локализацию точки минимума.

Функционал f_5 совмещает достоинства f_2 и f_4 . Являясь гладким как f_2 , он в то же время не допускает значительных отклонений точности аппроксимации в отдельных точках (обычно увеличивают v , начиная с $v = 2$).

1.8. Способ построения функционала при проектировании

ММ, описывающие динамику поведения многих приборов, часто могут быть представлены в виде систем обыкновенных дифференциальных уравнений (СОДУ), поведение решений которых близко поведению решений линейной СОДУ. Пусть ММ имеет вид

$$\dot{y} = \mathbf{F}(y, t, p), \quad y \in R^n, \quad p \in R^m,$$

где \mathbf{p} – управляемые параметры, которые нас интересуют.

Если ставится задача аппроксимации этой системой некоторой желаемой траектории $y_m(t)$ за счет выбора вектора \mathbf{p} , то в ряде случаев удастся уйти от задачи численного интегрирования исходной системы в процессе оптимизации, используя алгебраизацию задачи. При этом удастся автоматизировать построение оптимизируемого функционала.

Пусть вид движения в исходной системе близок желаемому движению. Тогда, задавшись аппроксимацией желаемой траектории,

$$y_i(t) = \sum_{k=1}^{r_i} a_{ik} \varphi_{ik}(t) = \sum_{k=1}^{r_i} a_{ik} e^{\alpha_{ik} t} (A_{ik} \cos \omega_{ik} t + B_{ik} \sin \omega_{ik} t).$$

Здесь $b_{ik} = (A_{ik}, B_{ik}, a_{ik}, \alpha_{ik}, \omega_{ik})$ – элементы вектора желаемых характеристик $b = (b_1, \dots, b_s)$ считаются известными числами.

После подстановки y_i в СОДУ получаем систему алгебраических уравнений – невязок $\Phi_i(b, p, t) \approx 0, i = 1, n$. После исключения времени одним из способов (по методу коллокаций в этой системе фиксируются

моменты времени $t = \left(0, \frac{\pi}{\omega}, \frac{\pi}{2\omega}, \frac{\pi}{3\omega}, \dots \right)$ получим переопределенную систему $\Phi_j(b, p) \approx 0, j = 1, N$.

Теперь решение задачи аппроксимации может быть сведено к минимизации суммы квадратов невязок, т. е. $\sum_i \Phi_i^2(b, p) \rightarrow \min_p$.

1.9. Функции многих переменных

Рассмотрим задачу $f(x) \rightarrow \min, x \in R^n$. Разложение в ряд Тейлора для функции $f(x)$ в точке \bar{x} имеет вид

$$f(x) = f(\bar{x}) + \nabla f(x)\Delta x + \frac{1}{2} \Delta x^T \nabla^2 f(x) \Delta x + o(\Delta x),$$

где $\Delta x = \bar{x} - x$;

$$\nabla f = \left[\frac{df}{dx_1}, \dots, \frac{df}{dx_n} \right]^T - \text{градиент } f(x);$$

$$\nabla^2 f = \left[\frac{d^2 f}{dx_i dx_j} \right] - \text{матрица Гессе (Гессиан)}$$

(симметрическая матрица, $n \times n$);

$$\Delta f = f(x) - f(\bar{x}) \geq 0.$$

Если это неравенство выполняется для всех $x \in R$, то \bar{x} – точка глобального минимума, если лишь в некоторой δ – окрестности точки, то \bar{x} – точка локального минимума x^* .

Как известно, для наличия в точке \bar{x} локального минимума (оптимального решения) необходимо и достаточно, чтобы $\nabla f(\bar{x}^*) = 0$ и $\nabla^2 f(\bar{x}^*)$ была положительно-определенной матрицей, т. е. квадратичная форма $Q = \Delta x^T \nabla^2 f(x) \Delta x > 0$ для любых $\Delta x \neq 0$.

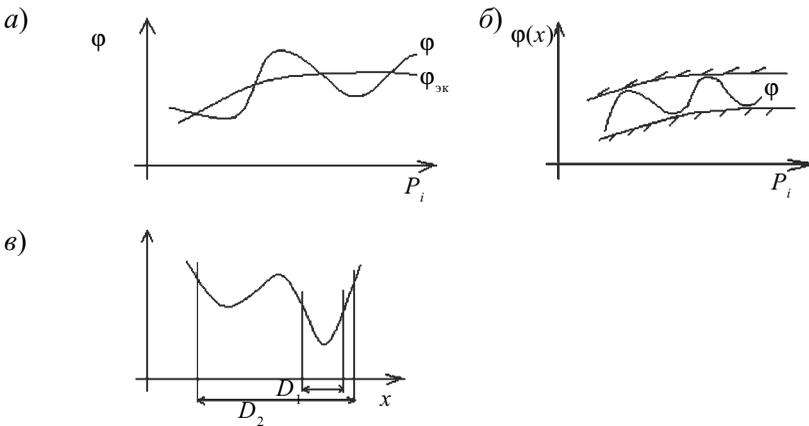


Рис. 5

Обычно довольствуются нахождением локального минимума, но если удастся показать, что $Q > 0$ для всех x , то $f(x)$ называется выпуклой функцией, а локальный минимум оказывается глобальным (рис. 5, в). Для выяснения характера функции надо исследовать характер квадратичной формы. Доказано, что характер формы совпадает с характером матрицы формы, т. е. $\mathbf{A} = \frac{1}{2} \nabla^2 f(x)$.

1.10. Критерии положительной определенности матриц

1. Матрица \mathbf{A} называется положительно-определенной, если все ее собственные значения $\lambda_i > 0$ (или если положительны все корни ее характеристического уравнения).

Пример

$$f(x_1, x_2) = 2x_1 + 6x_2 - 2x_1^2 - 3x_2^2 + 4x_1x_2;$$

$$\mathbf{A} = \frac{1}{2} \nabla^2 f(x) = \frac{1}{2} \begin{bmatrix} -4 & 4 \\ 4 & -6 \end{bmatrix} = \begin{bmatrix} -2 & 2 \\ 2 & -3 \end{bmatrix};$$

$$\det(\mathbf{A} - \lambda \mathbf{I}) = \det \begin{bmatrix} -2 - \lambda & 2 \\ 2 & -3 - \lambda \end{bmatrix} = 0;$$

$$\lambda^2 + 5\lambda + 2 = 0;$$

корни $\lambda_{1,2} = \frac{1}{2}(-5 \pm \sqrt{17})$ — отрицательны, следовательно, функция вогнутая.

Отметим, что для квадратичной функции матрица Гессе не зависит от точки, в которой она вычислена.

2. Критерий Сильвестра выглядит следующим образом. Симметричная матрица \mathbf{A} является положительно-определенной, если все угловые главные миноры положительны.

$\mathbf{A} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$, $\Delta_1 = 2 > 0$, $\Delta_2 = 3 > 0$ и отрицательно-определенной, если знаки угловых главных миноров чередуются, причем знаки нечетных миноров отрицательны.

3. Приведение квадратичной формы $Q(x)$ к сумме полных квадратов (форме Лагранжа)

$$1. Q(x) = a x_1^2 + 2 b x_1 x_2 + c x_2^2 = (a^2 x_1^2 + 2 a b x_1 x_2) + c x_2^2 =$$

$$\begin{aligned}
&= \frac{1}{a}(a^2 x_1^2 + 2 a b x x) + c x_2^2 = \\
&= \frac{1}{a}(a^2 x_1^2 + 2 a b x_1 x_2 + b^2 x_2^2 - b^2 x_2^2) + c = \\
&= \frac{1}{a}(a x_1 + b x_2)^2 + (a c - b^2) x_2^2;
\end{aligned}$$

$$q(x) > 0, \text{ если } \frac{1}{a} > 0;$$

$$a c - b^2 > 0.$$

2. Способ, основанный на **LU** – разложении:

$$\mathbf{A} = \mathbf{L} \mathbf{D} \mathbf{U},$$

где **U** – верхняя треугольная матрица; **D** – диагональная матрица. Для симметричной матрицы $\mathbf{A} = \mathbf{U}^T \mathbf{D} \mathbf{U}$

$$Q(x) = \mathbf{x}^T \mathbf{A} \mathbf{x} = \mathbf{x}^T \mathbf{U}^T \mathbf{D} \mathbf{U} \mathbf{x} = (\mathbf{U} \mathbf{x})^T \mathbf{D} (\mathbf{U} \mathbf{x}).$$

Вводим новые переменные $\mathbf{z} = \mathbf{U} \mathbf{x}$, $Q(x) = Q(z) = \mathbf{z}^T \mathbf{D} \mathbf{z} = \sum_{i=1}^n d_{ii} z_i^2$

(таким образом, необходимо найти матрицы **U** и **D**).

Рассмотрим задачу преобразования **A** к верхней треугольной матрице.

Пусть

$$\mathbf{A} = \begin{bmatrix} a & b \\ b & c \end{bmatrix}.$$

Первая строка делится на a , результат умножается на b и полученная строка вычитается из второй строки, т. е.

$$\begin{bmatrix} 1 & \frac{b}{a} \\ 0 & c - \frac{b^2}{a} \end{bmatrix}.$$

После деления второй строки на диагональный элемент получаем матрицу **U**

$$\mathbf{U} = \begin{bmatrix} 1 & \frac{b}{a} \\ 0 & 1 \end{bmatrix}.$$

Из элементов, на которые делили, получаем диагональную матрицу

$$\mathbf{D} = \begin{bmatrix} a & 0 \\ 0 & c - \frac{b^2}{a} \end{bmatrix}; \quad \mathbf{z} = \mathbf{U}\mathbf{x} = \begin{pmatrix} 1 & \frac{b}{a} \\ 0 & 1 \end{pmatrix}.$$

Теперь учитывая, что $Q(x) = \sum_{i=1}^n d_{ii} z_i^2$,

$$Q(x) = a \left(x_1 + \frac{b}{a} x_2\right)^2 + \left(c - \frac{b^2}{a}\right) x_2^2.$$

Итак, форма Q положительно-определенная, если знаки элементов матрицы \mathbf{D} больше нуля. Таким образом, при аналитическом задании функции $f(x)$ можем легко определить характер ее поведения по матрице формы.

Пример

$$f(x) = x_1 + 2x_3 + x_2x_3 - x_1^2 - x_2^2 - x_3^2 \rightarrow \min.$$

Является ли экстремальной стационарная точка $\mathbf{x}^{(*)} = (1/2, 2/3, 4/3)^T$ и если да, то точкой \max или \min ?

Пример

Является ли функция

$$f(x) = 2x_1^2 + x_2^2 + 3x_3^2 - x_1x_2 + 2x_1x_3 - x_2x_3 \text{ выпуклой?}$$

Решение

$$\mathbf{A} = \begin{vmatrix} 2 & -1/2 & 1 \\ -1/2 & 1 & -1/2 \\ 1 & -1/2 & 3 \end{vmatrix}.$$

Используем метод исключения Гаусса

$$\rightarrow \begin{vmatrix} 1 & -1/4 & 1/2 \\ 0 & 7/8 & -1/4 \\ 0 & -1/4 & 5/2 \end{vmatrix} \rightarrow \begin{vmatrix} 1 & -1/4 & 1/2 \\ 0 & 1 & 2/7 \\ 0 & 0 & 17/7 \end{vmatrix} \rightarrow \begin{vmatrix} 1 & -1/4 & 1/2 \\ 0 & 1 & 2/7 \\ 0 & 0 & 1 \end{vmatrix} = \mathbf{U};$$

$$\mathbf{D} = \begin{vmatrix} 2 & 0 & 0 \\ 0 & 7/8 & 0 \\ 0 & 0 & 17/8 \end{vmatrix}.$$

2. МЕТОДЫ БЕЗУСЛОВНОЙ ОПТИМИЗАЦИИ

Хотя большинство практических задач содержит ограничения, изучение методов безусловной оптимизации важно с нескольких точек зрения. Так, например, многие алгоритмы решения задачи с ограничениями предполагают сведение к последовательности задач безусловной оптимизации с помощью множителей Лагранжа или с помощью штрафных и барьерных функций.

2.1. Линейный поиск без использования производных

Одномерный поиск – основа многих алгоритмов для решения задач нелинейного программирования.

Обычно алгоритмы нелинейного программирования представляют собой следующую процедуру. На k -м шаге задается точка \mathbf{x}_k , определяется вектор направления \mathbf{d}_k и находится расстояние λ_k в направлении \mathbf{d}_k , затем вычисляется новая точка $\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k$ и процесс повторяется. Определение λ_k достигается решением задачи $\min f(\mathbf{x}_k + \lambda_k \mathbf{d}_k)$, зависящей от λ_k . Эта задача одномерного поиска.

Один из путей минимизации $\theta(\lambda) = f(x + \lambda d)$ – решить $\theta' = 0$ и отсюда найти λ . Однако если f недифференцируема, то и θ недифференцируема. В противном случае, $\theta'(\lambda) = \mathbf{d}^T \nabla f(x + \lambda d)$, т. е. для определения λ надо решать уравнение $\mathbf{d}^T \nabla f(x + \lambda d) = 0$, которое обычно нелинейно по λ . Более того, $\theta'(\lambda) = 0$ не обязательно доставляет минимум функции $\theta(\lambda)$. Это может быть локальный минимум или максимум. Поэтому, как правило, применяются численные процедуры минимизации θ .

1. Пусть функция $f(x)$ унимодальна на интервале $a \leq x \leq b$, а ее минимум достигается в точке x (рис. 6, а). Если заданы две точки x_1 и x_2 и $f(x_1) > f(x_2)$, то точка минимума лежит в $[a, x_1]$, т. е. этот интервал можно исключить. Это позволяет исключить полный перебор всех допустимых точек.

2. Выделим два этапа:

– установление границ (поиск границ достаточно широкого интервала);

– этап уменьшения интервала (уменьшить его длину до заранее установленной величины).

Установление граници

Эвристический способ выглядит следующим образом. Выбираем исходную точку x_0 (предполагается, что заданы технологические интервальные ограничения на переменные x , т. е. интервал $[a, b]$ и $h = h_{\min}$). Если $f(x_0 - |h|) \geq f(x_0) \leq f(x_0 + |h|)$, то выход (интервал не найден). Определяем направление движения. Знак h определяется путем сравнения значений:

$$f(x_0), f(x_0 + |h|) \text{ и } f(x_0 - |h|).$$

Если $f(x_0 - |h|) \geq f(x_0) \geq f(x_0 + |h|)$, то точка минимума правее x_0 и h – положительно (рис. 6, б). Знак h выясняется только в окрестности x_0 .

Если изменить знаки неравенств на противоположные, то h – отрицательно. Итерации выполняются по формуле

$$x_{k+1} = x_k + h, k = 0, 1, 2, \dots,$$

пока не получим $f(x_{k+1} - |h|) \geq f(x_{k+1}) \leq f(x_{k+1} + |h|)$, т. е. точка минимума лежит между $x_{k+1} - |h|$ и $x_{k+1} + |h|$ и поиск граничных точек завершен.

Если $f(x_{k+1} - |h|) \leq f(x_{k+1}) \geq f(x_{k+1} + |h|)$, то функция неунимодальна.

Возможен также случай полного отсутствия какого-либо минимума на $[a, b]$, если ни одно условие не выполнено, а вышли за границу интервала.

Уменьшение интервала

Равномерный поиск: разбить интервал на N частей и выбрать минимум.

Метод дихотомии

Деление пополам позволяет исключать половину интервала на каждой итерации.

Алгоритм

1. $x_2 = \frac{a+b}{2}$; $L = b - a$; найти $f(x_m)$.

2. $x_1 = a + \frac{L}{4}$ $x_2 = b + \frac{L}{4}$;

$$f(x_1), f(x_2).$$

3. Сравнить $f(x_1)$ и $f(x_m)$.

Если $f(x_1) < f(x_2)$, то исключить (x_m, b) , положив $b = x_m$.

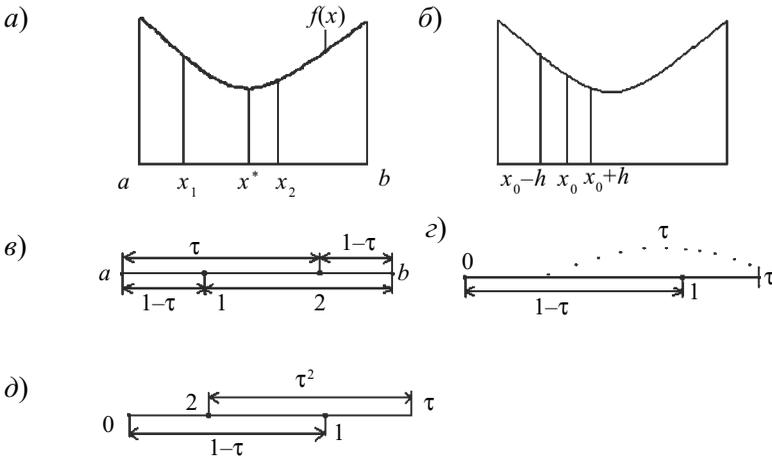


Рис. 6

Середина интервала переносится в точку x_1 .

$x_m = x_1$, переход к п. 5;

$f(x_1) \geq f(x_m)$, переход к п. 4.

4. $f(x_2) < f(x_m)$, исключаем (a, x_m) , положив $a = x_m$.

$x_m = x_2$, переход к п. 5;

$f(x_2) \geq f(x_m)$, исключаем (a, x_1) и (x_2, b) ;

$a = x_1$ $b = x_2$, переход к п. 5.

5. $L = b - a$.

Если $|L| < \varepsilon$ – закончить, иначе переход к п. 2.

Недостаток: забываем информацию в точке на сохраняемом интервале.

Метод золотого сечения

Рассмотрим симметричное расположение пробных точек на исходном интервале единичной длины. Пробные точки отстоят от границ на расстоянии τ -й части от длины интервала, т. е. в начале это расстояние равно $\tau = \tau \cdot 1$ (рис. 6, в). При таком расположении длина остающегося после исключения интервала всегда равна $\tau \cdot 1$ (длина интервала), независимо от того, какое из значений функции в пробных точках оказалось меньше.

Пусть исследуется правый интервал, тогда оставшийся содержит одну пробную точку, полученную ранее (мы ее используем) (рис. 6, г). Для того чтобы симметрия сохранилась, расстояние $(1 - \tau)$ должно состав-

лять τ часть длины интервала, равной τ , т. е. τ^2 . Таким образом, следующая пробная точка разместится на расстоянии τ^2 от правой границы уменьшенного интервала (рис. 6, δ). В силу симметрии имеем уравнение $1 - \tau = \tau^2$, откуда $\tau = (-1 \pm \sqrt{5})/2 = 0,618$ (при таком τ это выполняется).

Термин “золотое сечение” произошел от названия отношения последующих интервалов

$$\frac{L_{j-1}}{L_j} = \frac{L_j}{L_{j+1}} = \frac{L_{j+1}}{L_{j+2}} = \dots = \tau,$$

которое в этом методе постоянная величина. L_{j-1} делится на 2 части так, что отношение целого к большей части равно отношению большей части к меньшей, т. е. равно “золотому отношению”

$$\frac{\text{Ц}}{\text{Б}} = \frac{\text{Б}}{\text{М}}, \text{ в нашем случае } \frac{1}{\tau} = \frac{\tau}{1 - \tau} = \dots = \tau.$$

Метод Фибоначчи

Стратегия поиска связана с использованием чисел Фибоначчи, которые можно получить из рекуррентного уравнения

$$u_k = u_{k-1} + u_{k-2}, \quad k = 2, 3, \dots$$

$$u_0 = u_1 = 1.$$

Первые числа этого ряда

$$k \quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5$$

$$u_k \quad 1 \quad 1 \quad 2 \quad 3 \quad 5 \quad 8$$

В отличие от метода золотого сечения, здесь сокращение интервала неопределенности меняется от итерации к итерации.

При $n = 14$ (число шагов) апостериорный интервал неопределенности приблизительно в 5 раз меньше, чем полученный по методу деления пополам.

Недостаток: необходимо задать число шагов n , так как уже для определения первой точки надо знать отношение чисел Фибоначчи u_{n-1}/u_n (n можно оценить из неравенства $u_n > L_1 / L_n$, зная сами числа Фибоначчи). Отметим, что метод “золотого сечения” позволяет получить интервал только на 17 % больше, чем метод Фибоначчи и при больших n методы практически идентичны.

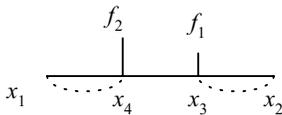
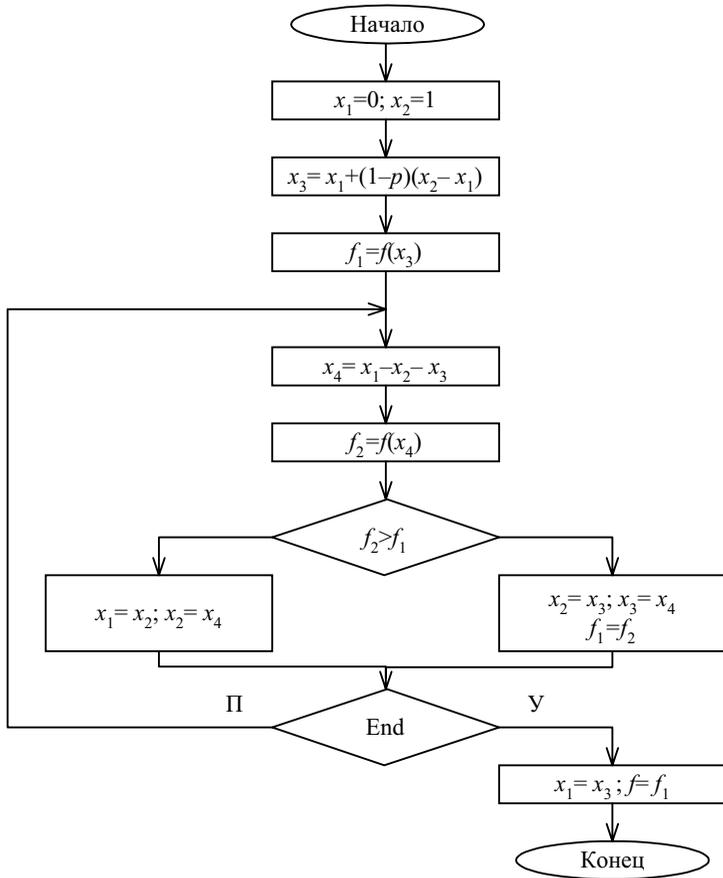


Рис. 7

Это позволяет объединить их в одной вычислительной процедуре (рис. 7)

$$p = \begin{cases} u_{n-1}/u_{n+1}, & \text{метод Фибоначчи,} \\ \tau^2, & \text{метод золотого сечения.} \end{cases}$$

В зависимости от знака разности $f(x_3) - f(x_4)$ отношение интервалов на k -м и $k-1$ -м шагах

$$t_k = \begin{cases} \frac{x_3 - x_1}{x_2 - x_1}; & f(x_3) > f(x_4); \\ \frac{x_2 - x_4}{x_2 - x_1}; & \text{иначе,} \end{cases}$$

$$t_k = 1 - p,$$

так как $x_3 - x_1 = x_2 - x_4$. Методы устойчивы к ошибкам округления, если $1/2 < t_k < 2/3$.

Полиномиальная аппроксимация

Идея связана с возможностью аппроксимации гладкой функции полиномом и использованием этого полинома для оценки координаты минимума. По-прежнему, предполагается унимодальность функции. Качество оценки оптимума можно повысить, повышая порядок полинома и уменьшая интервал аппроксимации (что более предпочтительно, так как выше 3-го порядка – расточительно с вычислительной точки зрения).

Квадратичная аппроксимация

Это простейший вид аппроксимации, в котором предполагается, что функцию можно приблизить квадратичным полиномом.

Если известны 3 точки: x_1, x_2, x_3 и соответствующие им значения функции: f_1, f_2, f_3 , то можно определить постоянные величины: a_0, a_1, a_2 так, что значения аппроксимирующей функции $g(x) = a_0 + a_1(x - x_1) + a_2(x - x_1)(x - x_2)$ совпадут со значением $f(x)$ в трех заданных точках. Наложим эти условия, учитывая, что $f_1 = f(x_1) = g(x_1) = a_0$, т. е. $a_0 = f_1$. Учитывая, что $f_2 = f(x_2) = g(x_2) = f_1 + a_1(x_2 - x_1)$, имеем $a_1 = (f_2 - f_1)/(x_2 - x_1)$.

Так как $f_3 = f(x_3) = g(x_3) = f_1 + (f_2 - f_1)/(x_2 - x_1)(x_3 - x_1) + a_2(x_3 - x_1)(x_3 - x_1)$, имеем

$$a_2 = \frac{1}{x_3 - x_2} \left(\frac{f_3 - f_1}{x_3 - x_1} - \frac{f_2 - f_1}{x_2 - x_1} \right).$$

Точку минимума квадратичной функции находим из условия

$$\frac{dg}{dx} = 0 = a_1 + a_2(x - x_2) + a_2(x - x_1),$$

откуда

$$\bar{x} = (x_2 + x_1)/2 - (a_1 / (2 a_2)).$$

Алгоритм метода Пауэлла

1. $x_2 = x_1 + \Delta$, x_1 – исходная точка, Δ – шаг (рис. 8, а).
2. $f(x_1), f(x_2)$.
3. Если $f(x_1) > f(x_2)$, то $x_3 = x_1 + 2\Delta$, иначе $x_3 = x_1 - \Delta$.
4. $f(x_3)$;

определяя $f_{\min} = \min\{f_1, f_2, f_3\}$, находим номер i -й точки и полагаем $x_{\min} = x_i$.

5. По x_1, x_2, x_3 вычислим \bar{x} (на основании вышеописанных формул).

6. Проверка на завершение:

$|f_{\min} - f(x)| < \varepsilon_1 \wedge |x_{\min} - x| < \varepsilon_2$, то вычисления завершить. В противном случае перейти к п. 7.

7. Выбор точек.

Выбираем лучшую точку (ту, у которой меньше значение функции) из точек (x_{\min} или x) и две точки по обе стороны от нее, нумеруем их в естественной последовательности x_1, x_2, x_3 и переходим к п. 4.

Сравнение методов линейного поиска без вычисления производных

Среди методов исключения интервалов с точки зрения числа вычисления функции наиболее эффективен метод Фибоначчи, далее метод золотого сечения, дихотомии, равномерного поиска.

Для достаточно больших n методы Фибоначчи и золотого сечения почти совпадают по эффективности. Для гладких функций быстрее оказывается метод Пауэлла, но для быстро изменяющихся функций он медленнее методов исключения интервалов. Названные методы предполагают унимодальность функции. В общем случае (если это не выполняется или не может быть легко проверено) интервал делится на части. Наиболее надежным считается метод золотого сечения.

2.2. Линейный поиск с использованием производной минимизируемой функции

Если функция дифференцируема, можно повысить эффективность поиска.

Метод Ньютона

Этот метод основан на квадратичной аппроксимации функции f в точке x . Это будет квадратичная функция q , а именно

$$q(x) = f(x_k) + f'(x_k)(x - x_k) + 1/2 f''(x_k)(x - x_k)^2,$$

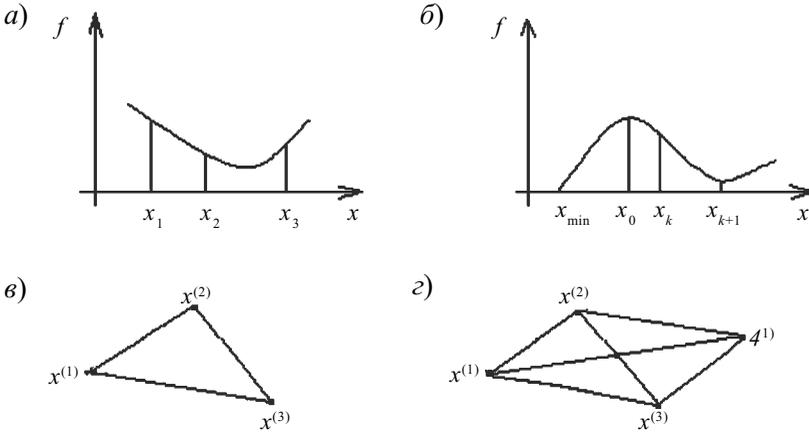


Рис. 8

экстремум определяется из условия, $\frac{\partial q}{\partial x} = 0$, т. е.

$$f'(x_k) + f''(x_k)(x - x_k) = 0,$$

откуда

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)},$$

причем x_{k+1} берем в точке, где $\frac{\partial q}{\partial x} = 0$.

Недостаток: метод расходится, если начальная точка правее x_0 (рис. 8, б).

Метод деления интервала пополам

Здесь на каждой итерации требуется только одно вычисление производной (в методе без использования производной требуется два вычисления функции).

Указание

1. $x_k = 1/2(a_k + b_k)$.

2. $f'(x_k) > 0$, т. е. функция возрастает и точка минимума не может быть правее x_k .

Метод кубической аппроксимации

Функция аппроксимируется полиномом 3-го порядка, но используются только две точки (а не 3), так как в каждой точке можно вычислить значения функции и ее производной. Этот метод более эффективен

вен (по сравнению с предыдущими). В случае, если производные найдены аналитически, но если они получаются на основе разностных формул, то предпочтение надо отдать методу Пауэлла.

2.3. Многомерный поиск без использования производных

Часто затруднительно или вообще невозможно найти аналитическое выражение для производных целевых функций. Для минимизации негладких функций (с разрывами) поиск осуществляется на основе сопоставления значений функций в пробных точках. Прибегая к этим методам (их можно применить всегда), надо быть уверенным в том, что метод другого типа применить нельзя, иначе можно дорого заплатить потерями машинного времени. Кроме того, общим их недостатком являются сомнительные гарантии сходимости к решению. Эти методы носят название – методов прямого поиска.

Среди них следует выделить в первую очередь:

- поиск по симплексу (по многограннику) Нелдера–Мида;
- метод Хука–Дживса;
- метод сопряженных направлений Пауэлла.

Первые два метода основаны на эвристических соображениях, последний – на теоретических результатах.

Метод поиска по симплексу

Симплекс – простейший выпуклый многогранник при данном числе измерений ($n = 2$ – треугольник, $n = 3$ – тетраэдр). Заметим, что данный метод не надо путать с симплексным методом в линейном программировании.

Идея в выборе базовой точки и оценки значений функции в окружающих точках, например в задаче с двумя переменными из пяти точек (базовая в центре квадрата), в качестве следующей точки выбирается наилучшая, вокруг которой строится новый образец.

Наименьшее число точек содержит регулярный симплекс (равноотстоящие вершины). В случае двух переменных (плоскость) – это равносторонний треугольник, в трехмерном пространстве – тетраэдр (рис. 8, в, г).

Алгоритм начинается с построения регулярного симплекса в пространстве n переменных (метод наиболее эффективен при $n \leq 6$). При этом определяется вершина, которой соответствует наибольшее значение функции. Эта вершина проецируется через центр тяжести осталь-

ных вершин в новую точку, которая будет вершиной нового симплекса итерации, и выполняются до тех пор, пока не будет накрыта точка минимума, либо не начнется циклическое движение (движение по двум или более симплексам). В этих случаях используются три следующих правила.

1. “Накрытие” точки минимума.

Если вершина, которой соответствует наибольшее значение функции уже построена на предыдущей итерации, то вместо нее используется вершина, которой соответствует следующее по величине значение функции.

2. Циклическое движение.

Если некоторая вершина не исключается на протяжении более $M = 1,65n + 0,05n^2$ итераций, то необходимо уменьшить размеры симплекса и построить новый симплекс, выбрав в качестве базовой точку с минимальным значением функции.

3. Окончание поиска.

Завершение поиска, если размеры симплекса или разности между значением функции в вершинах становятся достаточно малы.

Реализация метода основана на двух типах вычислений:

– построение симплекса при заданной базовой точке $x^{(0)}$

$$x^{(i)} = \begin{cases} x_j^{(0)} + \delta_1, & j \neq i, \\ x_j^{(0)} + \delta_2, & j = i, \end{cases} \quad i, j = \overline{1, n},$$

(вектор координат остальных n вершин симплекса),

$$\text{где } \delta_1 = \varphi_1(n, a) = \frac{\sqrt{(n+1)} + n - 1}{n\sqrt{2}} a; \quad \delta_2 = \varphi_2(n, a) = \frac{\sqrt{n+1} - 1}{n\sqrt{2}} a;$$

a – масштабный множитель (при $a = 1$ ребра единичной длины);

Расчет координат отраженной точки.

Пусть $x^{(j)}$ точка, подлежащая отражению. Центр тяжести остальных n

точек расположен в точке $x_c = \sum_{i=0}^n x^{(i)} / n, i \neq j$.

Точки прямой, проходящей через $x^{(j)}$ и x_c , задаются формулой $x = x^{(j)} + \theta(x_c - x^{(j)})$.

При $\theta = 0$ получим исходную точку $x^{(j)}$, при $\theta = 1$ получаем точку x_c , при $\theta = 2$ получаем новую вершину.

Таким образом, $x_{\text{нов}}^{(j)} = 2x_c - x_{\text{пред}}^{(j)}$.

Модификация Нелдера и Мида

При отражении симплекса выполнить растяжение или сжатие по прямой $\mathbf{x} = \mathbf{x}^{(h)} + (1 + q)(\mathbf{x}_c - \mathbf{x}^{(h)})$, где $\mathbf{x}^{(h)}$ – координата вершины с максимальным значением функции $f(\mathbf{x})$.

Введем 3 характерных вершины: $\mathbf{x}^{(h)}$ – соответствует максимальному значению функции; $\mathbf{x}^{(g)}$ – соответствует следующему по величине значению функции; $\mathbf{x}^{(l)}$ – соответствует наименьшему значению функции;

Затем три параметра: α, β, γ . Обозначим $\mathbf{x}^{(r)}$ – отраженную вершину (новую); $f^{(r)}$ – соответствующее значение функции.

Растяжение и сжатие выполняется по следующим правилам:

а) нормальное отражение: $f^{(l)} < f^{(r)} < f^{(g)}$, $\theta = \alpha = 1$ (рис. 9, а);

б) сжатие (если минимум лежит внутри многогранника, тогда надо сжать)

$$f^{(r)} > f^{(g)}, f^{(r)} f^{(h)}, \theta = \beta = -0,5;$$

$$f^{(g)} < f^{(r)} < f^{(h)}, \theta = \beta = 0,5, \text{ (рис. 9, б, в);}$$

в) растяжение

$$f^{(r)} < f^{(l)}, \theta = \gamma = 2, \text{ (рис. 9, г).}$$

Алгоритм

Шаг 0: задание α, β, γ , количество итераций и начальную точку.

Шаг 1: формирование координат вершин исходного симплекса.

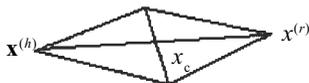
Шаг 2: вычисление значений функции во всех вершинах.

Шаг 3: ранжирование всех вершин по значениям функции и выбор трех вершин: $\mathbf{x}^{(h)}, \mathbf{x}^{(g)}, \mathbf{x}^{(l)}$.

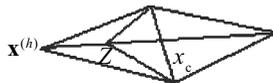
Шаг 4: вычисление координат центраида.

Шаг 5: выполнение операции отражения (с учетом тестов (а), (б), (в)).

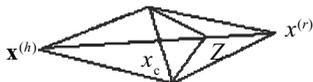
а)



б)



в)



г)

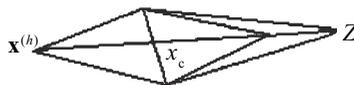


Рис. 9

$\mathbf{x}^{(3)}$

Шаг 6: замена вершины $\mathbf{x}^{(h)}$ на лучшую с вычислением значения функции в этой точке, иначе сжатие всего симплекса относительно наилучшей вершины $\mathbf{x}^{(l)}$, т. е. $\mathbf{x}^{(i)} = \frac{1}{2}(\mathbf{x}^{(i)} + \mathbf{x}^{(l)})$, $i \neq l$.

Шаг 7: проверка критерия окончания.

Если надо продолжить и была выполнена операция сжатия, то переход на шаг 2. Если была заменена наихудшая вершина, то на шаг 3.

Пример: $\min f(x) = (1-x_1)^2 + (2-x_2)^2$;

$\mathbf{x}^{(0)} = [0, 0]^T$ – базовая;

$$\delta_1 = \frac{\sqrt{3}+1}{2\sqrt{2}} \quad a = 1,93; \quad \delta_2 = \frac{\sqrt{3}-1}{2\sqrt{2}} \quad a = 0,517;$$

$\mathbf{x}^{(1)} = [0,517, 1,93]^T$; $\mathbf{x}^{(2)} = [1,93, 0,517]^T$; $f(x^{(1)}) = 0,237$; $f(x^{(2)}) = 3,065$.

Так как $f(x^{(0)}) = 5$, то надо отразить точку $x^{(0)}$.

$$\mathbf{x}_c = \frac{1}{2}(\mathbf{x}^{(1)} + \mathbf{x}^{(2)}).$$

Используя $\mathbf{x}_{\text{нов}}^{(j)} = 2\mathbf{x}_c - \mathbf{x}_{\text{пред}}^{(g)}$ (соответствует нормальному отражению), имеем: $\mathbf{x}^{(3)} = \mathbf{x}^{(1)} + \mathbf{x}^{(2)} - \mathbf{x}^{(0)} = [2,449, 2,449]^T$; $f(\mathbf{x}^{(3)}) = 2,3$, т. е. функция убывает.

Метод поиска Хука–Дживса

Множество отраженных точек описывается вектором, определяющим некоторое направление поиска, остальное – это выбор наилучшего шага λ .

Можно построить стратегию поиска, по которой одно или несколько направлений уточняется на каждой итерации вдоль каждого координатного направления в пространстве управляемых переменных. Чтобы гарантировать возможность проведения поиска по всей области изменения функции, используют n независимых направлений (n – размерность вектора \mathbf{x}). Происходит циклическое изменение переменных, причем каждый раз меняется только одна переменная. Вдоль каждого направления осуществляется поиск минимума в результате минимизации функции одной переменной.

В случае несферических линий уровня метод оказался неэффективен. Хуком и Дживсом была предложена модификация, связанная с учетом информации, полученной на предыдущих итерациях, – поиск проводить также в направлении $\mathbf{d}^{(i)} = \mathbf{x}^{(i)} - \mathbf{x}^{(i-1)}$ (рис. 10, а).

Процедура представляет собой комбинацию исследующего поиска (выявление характера локального поведения функции и направлений

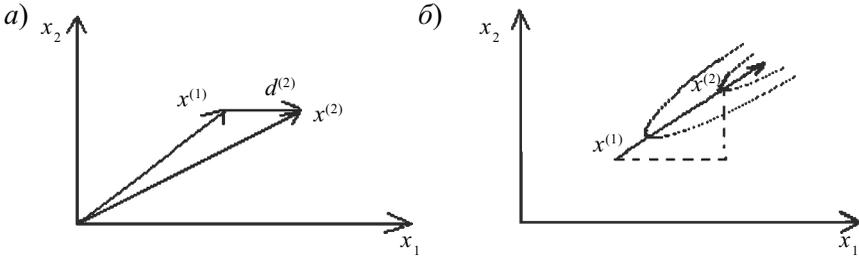


Рис. 10

вдоль “оврага”) с циклическим изменением переменных и ускоряющегося поиска по образцу (движение по “оврагам”) (рис. 10, б).

1. Исследующий поиск.

Требует задания величины шага, возможно различной для разных направлений, и изменяющийся в процессе поиска. Поиск начинается в некоторой исходной точке. Если значение целевой функции в новой точке не превышает значения в исходной точке, то шаг рассматривается как успешный, иначе надо вернуться в исходную точку и сделать шаг в противоположном направлении. При переборе всех n координат исследующий поиск завершается. Полученную в результате точку называют базовой.

2. Поиск по образцу (найденному направлению).

Заключается в реализации одного шага из базовой точки прямой, соединяющей эту точку с предыдущей базовой точкой $\mathbf{x}_p^{(k+1)} = \mathbf{x}^{(k)} + (\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)})$ (точка, построенная при движении по образцу).

Если движение по образцу не приводит к уменьшению функции, то $\mathbf{x}_p^{(k+1)}$ фиксируется в качестве временной базовой точки и вновь проводится исследующий поиск. Если в результате поиска получается точка с меньшим значением функции, чем в точке $\mathbf{x}^{(k)}$, то она рассматривается как новая базовая точка $\mathbf{x}^{(k)}$. Если исследующий поиск неудачен, необходимо вернуться в точку $\mathbf{x}^{(k)}$ и провести исследующий поиск с целью выявления нового направления минимизации. Если такой поиск не приводит к успеху, требуется уменьшить величину шага и возобновить исследующий поиск. Поиск завершается, когда длина шага становится достаточно малой.

Алгоритм Хука–Дживса

Обозначим: \mathbf{x}^B – точка начала; \mathbf{x}^C – после изменения всех координат (базовая); \mathbf{x}^D – точка роста. (рис. 11, а).

Шаг 0: задание начальной точки $\mathbf{x}^B = \mathbf{x}^0$ и определение $f(\mathbf{x}^B)$.

Шаг 1 : зондирующий поиск \mathbf{x}^C , в котором, если $f(\mathbf{x}^C) < f(\mathbf{x}^B)$ – удача.

Шаг 2 : если поиск не удачен, то уменьшение шага. Если шаг меньше предельно допустимого, то остановка, иначе переход к шагу 1.

Шаг 3 : (поиск по образцу) если поиск удачен, то определим точку роста $\mathbf{x}^D = \mathbf{x}^C + (\mathbf{x}^C - \mathbf{x}^B) = 2\mathbf{x}^C - \mathbf{x}^B$. Из точки \mathbf{x}^D осуществляется зондирующий поиск \mathbf{x}^E . Если $f(\mathbf{x}^E) < f(\mathbf{x}^C)$ – удача.

Шаг 4 :если поиск удачен, то $\mathbf{x}^B = \mathbf{x}^C$; $\mathbf{x}^C = \mathbf{x}^E$, переход на шаг 3.

Шаг 5 :если поиск неудачен, то берем за \mathbf{x}^B последнюю удачную точку, уменьшаем шаг и идем на шаг 1.

Задание. Найти точку минимума функции $f(x) = 8x_1^2 + 4x_1x_2 + 5x_2^2$, учитывая, что $\mathbf{x}^0 = [-4; 4]^T$; $\Delta x = [1; 1]^T$ шаг; $\alpha = 2$ – коэффициент уменьшения шага; $\varepsilon = 10^{-4}$.

Достоинства метода:

- позволяет двигаться вдоль узких, глубоких оврагов;
- время поиска линейно растёт от числа переменных (для градиентных методов $t \cong O(n)$).

Недостатки:

- исследующий поиск не позволяет найти направление убывания функции и метод заклинивает (рис. 11, б). При этом целесообразно повторить поиск из другой начальной точки;
- вблизи минимума скорость движения замедляется.

Метод Пауэлла

Метод сопряженных направлений Пауэлла – это наиболее эффективный метод прямого поиска. Информация, полученная на предыдущих итерациях, используется для построения векторов направлений поиска. Метод ориентирован на решение задач с квадратичными целевыми функциями. Этот факт не умаляет общности, так как любая функция в окрестности оптимума может быть аппроксимирована квадратичной функцией.

Идея алгоритма заключается в том, что если квадратичная функция n переменных приведена к виду суммы полных квадратов, то оптимум может быть найден в результате n одномерных поисков по преобразованным координатным направлениям. Процедура преобразования квадратичной функции $q(x) = a + b^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \mathbf{C} \mathbf{x}$ к виду суммы полных квадратов эквивалентна нахождению такой матрицы преобразования \mathbf{T} ,

которая приводит матрицу квадратичной формы к диагональному виду. Таким образом, квадратичная форма: $Q(x) = \mathbf{x}^T \mathbf{C} \mathbf{x}$ преобразованием $\mathbf{x} = \mathbf{T} \mathbf{z}$ приводится к виду $Q(z) = \mathbf{z}^T \mathbf{T}^T \mathbf{C} \mathbf{T} \mathbf{z} = \mathbf{z}^T \mathbf{D} \mathbf{z}$. Вместо координат \mathbf{x} в стандартной системе $e^{(i)}$ используются координаты \mathbf{z} в системе, заданной векторами t_j (столбцы \mathbf{T}). Кроме того, система векторов t_j соответствует главным осям квадратичной формы. Одномерный поиск точек оптимума пространства \mathbf{z} , эквивалентен поиску вдоль каждой из главных осей квадратичной функции, т. е. вдоль направлений, заданных векторами t_j .

Пример: $f(x) = 4x_1^2 + 3x_2^2 - 4x_1x_2 + x_1 \rightarrow \min x_1 = z_1 + \frac{1}{2}z_2, x_2 = z_2$ или

$$\mathbf{x} = \begin{vmatrix} 1 & \frac{1}{2} \\ 0 & 1 \end{vmatrix} \mathbf{z};$$

$$f(z) = 4z_1^2 + 2z_2^2 + z_1 + \frac{1}{2}z_2.$$

Пусть $\mathbf{x}_0 = [0, 0]^T$; $\mathbf{t}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$; $\mathbf{t}_2 = \begin{pmatrix} 1/2 \\ 1 \end{pmatrix}$.

1. Поиск в направлении t_1

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \lambda \mathbf{t}_1, \lambda = -\frac{1}{8} \text{ и } \mathbf{x}^{(1)} = \left(-\frac{1}{8}, 0\right)^T.$$

Теперь из точки $\mathbf{x}^{(1)}$ проводится поиск в направлении \mathbf{t}_2 .

$$\text{Находим } \lambda = -\frac{1}{8} \text{ и } \mathbf{x}^{(2)} = \left[-\frac{3}{16}, -\frac{1}{8}\right]^T.$$

Выполнили n (здесь $n = 2$) одномерных поисков вдоль каждого из n сопряженных направлений \mathbf{t}_j .

Зная \mathbf{C} , можно найти \mathbf{T} (т. е. матрицу собственных векторов для \mathbf{C}), но мы хотим использовать только значения функции $f(x)$, так как матрица \mathbf{C} неизвестна.

Систему направлений \mathbf{t}_j можно найти из следующего свойства квадратичных функций.

Если заданы $Q(x)$, две точки $\mathbf{x}^{(1)}$, $\mathbf{x}^{(2)}$, направление \mathbf{d} (рис. 11, в), тогда точка $\mathbf{y}^{(1)}$ минимизирует $Q(\mathbf{x}^{(1)} + \lambda \mathbf{d})$, а $\mathbf{y}^{(2)}$ минимизирует $Q(\mathbf{x}^{(2)} + \lambda \mathbf{d})$ и направление $(\mathbf{y}^{(2)} - \mathbf{y}^{(1)})$ сопряжено с \mathbf{d} . Поиск вдоль направления $\mathbf{y}^{(2)} - \mathbf{y}^{(1)}$ обеспечивает нахождение точки оптимума.

На практике для получения сопряженного направления не задают точки и некоторое направление, а используют одну начальную точку и единичные векторы $\mathbf{e}^{(1)}$.

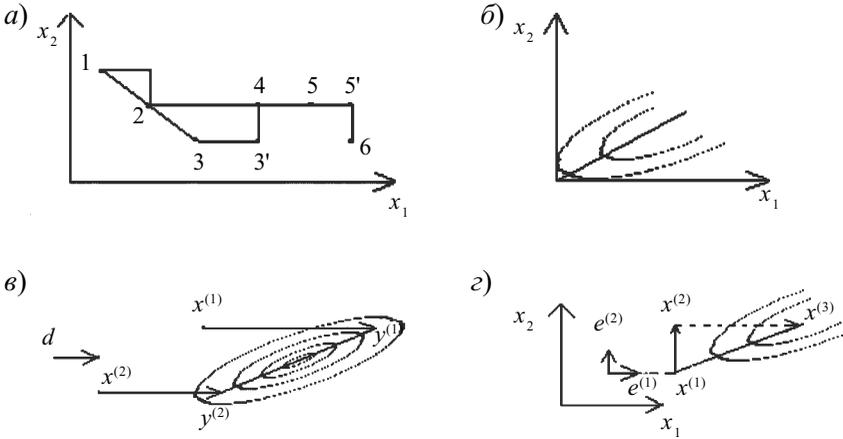


Рис. 11

Пусть $\mathbf{e}^{(1)} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $\mathbf{e}^{(2)} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$. Найдем, $\lambda^{(0)}$ соответствующее минимуму $f(\mathbf{x}^{(0)} + \lambda^{(0)} \mathbf{e}^{(1)})$, и положим $\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \lambda^{(0)} \mathbf{e}^{(1)}$.

Затем найдем $\lambda^{(1)}$, соответствующее минимуму $f(\mathbf{x}^{(1)} + \lambda^{(1)} \mathbf{e}^{(2)})$, и положим $\mathbf{x}^{(2)} = \mathbf{x}^{(1)} + \lambda^{(1)} \mathbf{e}^{(2)}$ (рис. 11, з). Теперь вычислим $\lambda^{(2)}$, минимизируя $f(\mathbf{x}^{(2)} + \lambda^{(2)} \mathbf{e}^{(1)})$ и положим $\mathbf{x}^{(3)} = \mathbf{x}^{(2)} + \lambda^{(2)} \mathbf{e}^{(1)}$. Тогда $(\mathbf{x}^{(3)} - \mathbf{x}^{(1)})$ и $\mathbf{e}^{(1)}$ окажутся сопряженными (свойство квадратичных функций). Теперь, если выполнить поиск в направлении $(\mathbf{x}^{(3)} - \mathbf{x}^{(1)})$, найдем минимум.

Алгоритм

Шаг 1: задать $\mathbf{x}^{(0)}$ и систему n независимых направлений $\mathbf{s}^{(i)} = \mathbf{e}^{(i)}$, $i = \overline{1, n}$.

Шаг 2: минимизировать $f(x)$ при последовательном движении вдоль $(n+1)$ направления. При этом полученная ранее точка минимума берется в качестве исходной, а направление $\mathbf{s}^{(n)}$ используются как при первом, так и при последнем поиске.

Шаг 3: определить новое сопряженное направление.

Шаг 4: заменить $\mathbf{s}^{(i)}$ на $\mathbf{s}^{(2)}$ и т. д., $\mathbf{s}^{(n)}$ заменить сопряженным направлением, переход к шагу 2.

Пример: $f(x) = 2x_1^3 + 4x_1x_2^3 - 10x_1x_2 + x_2^2$, $\mathbf{x}^{(0)} = (5, 2)^T$, $f(\mathbf{x}^{(0)}) = 3,14$

Шаг 1: $\mathbf{s}^{(1)} = (1, 0)^T$, $\mathbf{s}^{(2)} = (0, 1)^T$.

Шаг 2: а) $f(\mathbf{x}^{(0)} + \lambda \mathbf{s}^{(2)}) \rightarrow \min_{\lambda}$, откуда $\lambda^* = -0,81$, $\mathbf{x}^{(1)} = (5, 2)^T - 0,81(0, 1)^T$, $f(\mathbf{x}^{(1)}) = 250$;

$$\text{б) } f(x^{(1)}) + \lambda s^{(1)} \rightarrow \min_{\lambda}, \text{ откуда } \lambda^* = -3,26, \mathbf{x}^{(2)} = (1,74, 1,19)^T, f(x^{(2)}) = 1,1;$$

$$\text{в) } f(x^{(2)}) + \lambda s^{(2)} \rightarrow \min_{\lambda}, \text{ откуда } \lambda^* = -0,098, \mathbf{x}^{(3)} = (1,74, 1,092)^T, f(x^{(3)}) = 0,72.$$

$$\text{Шаг 3: } \mathbf{s}^{(3)} = \mathbf{x}^{(3)} - \mathbf{x}^{(1)} = (-3,26, -0,098)^T, \mathbf{s}^{(3)} = \frac{\mathbf{s}^{(3)}}{\|\mathbf{s}^{(3)}\|} (-0,9995, -0,03)^T, \\ \mathbf{s}^{(1)} = \mathbf{s}^{(2)}, \mathbf{s}^{(2)} = \mathbf{s}^{(3)}.$$

Шаг 4: найти λ (вдоль сопряженного направления) $f(x^{(3)} + \lambda s^{(3)}) \rightarrow \min_{\lambda}$, откуда $\lambda^* = 0,734$, $\mathbf{x}^{(4)} = [1,006, 1,070]^T$, $f(x^{(4)}) = -2,86$; $\mathbf{x}^{(0)} = \mathbf{x}^{(4)}$, перейти на шаг 2 (так как f – не квадратичная, требуются итерации).

2.4. Градиентные методы

При использовании даже самых эффективных прямых методов иногда требуется весьма большое количество вычислений значений функции. Естественно реализовать возможность нахождения стационарных точек в $(\nabla f(\bar{x}) = 0)$, т. е. применить методы, использующие значение градиента функции (что возможно, в случае непрерывности $f(x)$, $\nabla f(x)$, $\nabla^2 f(x)$).

В основе лежит итерационная схема (так как $\nabla f(x)$ – нелинейная функция x)

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{s}(\mathbf{x}^{(k)}),$$

где $\alpha^{(k)}$ – шаг определенный в процессе линейного поиска; $\mathbf{s}(x)$ – направление поиска в n -мерном пространстве переменных \mathbf{x} .

Метод Коши

Метод использует направление наискорейшего спуска, т. е. наибольшего локального минимума функции. Схема метода имеет вид $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \nabla f(\mathbf{x}^{(k)})$, так как функция быстрее всего убывает в направлении антиградиента.

Упражнение

Решить задачу

$$f(x) = (x_1 - 2)^4 + (x_1 + 2x_2)^2 \rightarrow \min, \mathbf{x}^{(0)} = (0, 3)^T \text{ методом Коши.}$$

$$\text{Ответ: } \mathbf{x} = (2,00, 1,00)^T.$$

Недостатком метода является низкая скорость сходимости, так как изменение переменных зависит от величины градиента, который приближенно равен 0 в окрестности точки минимума.

Достоинства метода

1. Устойчивость, так как здесь $f(\mathbf{x}^{(k+1)}) \leq f(\mathbf{x}^{(k)})$ всегда при достаточно малых α .

2. На больших расстояниях от точки минимума позволяет уменьшить $f(x)$. Поэтому метод используется для получения хороших начальных точек.

Задание

1. Нарисовать блок-схему алгоритма.

2. Решить $f(x) = 8x_1^2 + 4x_1x_2 + 5x_2^2$, учитывая что $\mathbf{x}^{(0)} = [10, 10]$, $\mathbf{x}^* = [0, 0]$,

$$\|\nabla f(\mathbf{x}^{(k)})\| \leq 0 - \text{критерий останова.}$$

Метод Ньютона

Если функция квадратична, то решение методом Ньютона достигается за один шаг! Метод Коши наиболее эффективен, когда линии уровня – окружности (тогда антиградиент приводит к точке минимума). В общем случае нужно привлечь информацию о вторых производных. Рассмотрим квадратичную аппроксимирующую функцию $f(x)$ в точке $\mathbf{x}^{(k)}$

$$\bar{f}(x, \mathbf{x}^{(k)}) = f(\mathbf{x}^{(k)}) + \nabla f(\mathbf{x}^{(k)})^T \Delta \mathbf{x} + \frac{1}{2} \Delta \mathbf{x}^T \nabla^2 f(\mathbf{x}^{(k)}) \Delta \mathbf{x}.$$

Возьмем градиент от обеих частей равенства. Пусть во вновь получаемой точке $\mathbf{x}^{(k+1)}$ градиент аппроксимирующей функции обращается в нуль, т. е.

$$\nabla \bar{f}(\mathbf{x}, \mathbf{x}^{(k)}) = \nabla f(\mathbf{x}^{(k)}) + \nabla^2 f(\mathbf{x}^{(k)}) \Delta \mathbf{x} = 0.$$

Полученное уравнение решим относительно $\Delta \mathbf{x}$

$$\Delta \mathbf{x} = -\nabla^2 f(\mathbf{x}^{(k)})^{-1} \nabla f(\mathbf{x}^{(k)}).$$

Учитывая, что $\Delta \mathbf{x} = \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}$, получим итерационную формулу для метода Ньютона

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \nabla^2 f(\mathbf{x}^{(k)})^{-1} \nabla f(\mathbf{x}^{(k)}).$$

Пример: $f(\mathbf{x}) = 8 + 4x_1x_2 + 5 \rightarrow \min$;

$$\nabla f(x) = \begin{pmatrix} 16x_1 & 4x_2 \\ 10x_2 & 4x_1 \end{pmatrix}; \quad \nabla^2 f(x) = \begin{pmatrix} 16 & 4 \\ 4 & 10 \end{pmatrix};$$

$$\mathbf{x}^{(0)} = (10, 10)^T$$

$$\mathbf{x}^{(1)} = (10, 10)^T - \frac{1}{144} \begin{pmatrix} 10 & -4 \\ -4 & 16 \end{pmatrix} \begin{pmatrix} 200 \\ 140 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} - \text{оптимальная точка.}$$

Из итерационной формулы следует, что функция убывает от итерации к итерации, если $-\nabla f(\bar{x})^T \nabla^2 f(\bar{x})^{-1} \nabla f(\bar{x}) < 0$, что выполняется, если матрица Гессе – положительно-определена, тогда и $\nabla^2 f^1$ – то же положительно определена, т. е. метод работает в случае, если $f(x)$ выпукла (вниз).

Поскольку используется квадратичное приближение, то если оно неверно описывает поведение $f(x)$ вне малых окрестностей опорных точек $\mathbf{x}^{(k)}$, то метод работает плохо.

Обобщенный градиентный алгоритм

Шаг 1. Задать $\mathbf{x}(0)$, M – максимальное количество итераций, N – количество переменных; ε_1 – параметр сходимости алгоритма; ε_2 – параметр сходимости для поиска вдоль прямой.

Шаг 2. Положить $k = 0$.

Шаг 3. Определить $\nabla f(x(k))$.

Шаг 4. Выполняется ли $\|\nabla f(x(k))\| \leq \varepsilon$?

Да – печать “сходимость : градиент”; перейти к шагу 13.

Нет – перейти к следующему шагу.

Шаг 5. Выполняется ли неравенство $k \geq M$?

Да – печать “окончание поиска”, перейти к шагу 13.

Нет – переход к следующему шагу.

Шаг 6. Вычислить $\mathbf{s}(x(k))$.

Шаг 7. Выполняется ли $\nabla f(x^T(k))\mathbf{s}(x(k)) < 0$?

Да – перейти к шагу 9.

Нет – положить $\mathbf{s}(x(k)) = -\nabla f(x(k))$, печать “возврат :неудачное направление “перейти к шагу 9.

Шаг 8. Найти такое $\alpha(k)$, при котором $f(x(k)) + \alpha(k)\mathbf{s}(x(k)) \rightarrow \min$ (используя параметр ε_2).

Шаг 9. Положить $x(k+1) = x(k) + \alpha(k)\mathbf{s}(x(k))$.

Шаг 10. Выполняется ли $f(x(k+1)) < f(x(k))$

Да – переход к шагу 11.

Нет – печать “окончание поиска:нет уменьшения функции”, перейти к шагу 13.

Шаг 11. Выполняется ли $\|\Delta x\|/\|x(k)\| \leq \varepsilon_1$?

Да – печать “окончание поиска: нет продвижения к решению”, переход к шагу 13.

Нет – переход к шагу 12.

Шаг 12. Положить $k = k+1$. Перейти к шагу 3.

Шаг 13. Останов.

Модифицированный метод Ньютона

Модификация метода Ньютона состоит в следующем.

1. Используется формула : $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha^{(k)} \nabla f(\mathbf{x}^{(k)})$, где $\alpha^{(k)}$ выбираем из условия $f(\mathbf{x}^{(k+1)}) \rightarrow \min_{\alpha^{(k)}}$, тогда гарантируется, что $f(\mathbf{x}^{(k+1)}) \leq f(\mathbf{x}^{(k)})$,

т. е. повышается надежность в случае неквадратичных функций.

2. Если $f(x)$ не выпукла ($\mathbf{x}^{(k)}$ – седловая точка), что определяется в процессе итераций по знаку собственных значений матрицы Гессе в качестве направления спуска $\mathbf{d}^{(k)}$ берут вектор, удовлетворяющий неравенству $\mathbf{d}^{(k)T} \nabla^2 f(\mathbf{x}^{(k)}) \mathbf{d}^{(k)} < 0$, где $\mathbf{d}^{(k)}$ – направление отрицательной кривизны.

Пример

Пусть $\mathbf{G}'(\mathbf{x}^{(k)}) = \nabla^2 f(\mathbf{x}^{(k)}) = \begin{pmatrix} 1 & 0 \\ 0 & -5 \end{pmatrix}$; $\nabla f(\mathbf{x}^{(k)}) = \begin{pmatrix} 2 \\ 5 \end{pmatrix}$, тогда ньютоновское направление $\mathbf{d}^{(k)} = \begin{pmatrix} -2 \\ 5 \end{pmatrix}$ приводит в седловую точку функции $f^{(k)} + \nabla f^{(k)} \Delta \mathbf{x} + \frac{1}{2} \Delta \mathbf{x}^T \nabla^2 f \Delta \mathbf{x}$, так как собственные значения $\nabla^2 f(x)$ отличаются по знаку ($\lambda_1 < 0$, $\lambda_2 > 0$). Оно составляет острый угол с градиентом ∇f , т. е. указывает в сторону возрастания функции.

Можно заменить собственные значения матрицы $\nabla^2 f(x)$ их модулями, тогда направлением $\overline{\mathbf{d}^{(k)}}$ будет вектор, равный по длине первоначальному, но направленный в противоположном направлении, т. е. в сторону минимума функции. При этом способе модификации получается следующая матрица вторых производных:

$$\mathbf{G}^{(k)} = \overline{\nabla^2 f(x^{(k)})} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^T = \begin{pmatrix} 1 & 0 \\ 0 & 5 \end{pmatrix},$$

где $\mathbf{\Lambda}$ – диагональная матрица с измененными собственными значениями; \mathbf{U} – матрица из собственных векторов $\mathbf{G}^{(k)}$. Заметим, что $\mathbf{d}^{(k)}$ составляет острый угол с $\overline{\nabla f(\mathbf{x}^{(k)})}$, т. е. указывает в направлении возрастания функции, а вектор $\mathbf{d}^{(k)}$ – в сторону убывания.

Метод Марквардта

Метод удачно сочетает положительные свойства обоих методов (метод Коши и Ньютона): первый работает вдали, а второй вблизи точки оптимума.

Направление поиска определяется вектором $\mathbf{s}(x^{(k)}) = -[\mathbf{G}^{(k)} + \lambda^{(k)}\mathbf{I}]^{-1} \nabla f(x^{(k)})$.

Параметру $\lambda^{(0)}$ присваивают большое значение, например 10^4 , так,

что $[\mathbf{G}^{(0)} + \lambda^{(0)} \mathbf{I}]^{-1} = [\lambda^{(0)} \mathbf{I}]^{-1} = \frac{1}{\lambda^{(0)}} \mathbf{I}$, поэтому большим значениям соответствует направление наискорейшего спуска ($-\nabla f(x^{(k)})$). При уменьшении λ до 0 направление $\mathbf{s}(x)$ изменяется до направления Ньютона.

Если после первого шага $f(x^{(1)}) < f(x^{(0)})$, то следует выбрать $\lambda^{(1)} < \lambda^{(0)}$ и сделать следующий шаг, где $\beta > 1$, и вновь реализовать предыдущий шаг.

Достоинство метода: высокая скорость сходимости и отсутствие необходимости поиска вдоль прямой.

Недостаток метода: необходимо вычислять матрицу $\mathbf{G}^{(k)}$ и решать систему линейных уравнений. Данный метод наиболее часто используется в задачах вида

$$f(x) = \sum_{i=1}^m (\varphi(x, t_i) - \tilde{\varphi}_i)^2 \rightarrow \min_x,$$

где $\tilde{\varphi}_i$ – экспериментальные данные.

Тогда матрица Гессе может быть получена при использовании только первых производных, а именно используют особую структуру $\nabla f(x)$ и $\nabla^2 f(x)$.

Пусть $\mathbf{I}(x) = \left[\frac{df_i}{dx_j} \right]$ – матрица Якоби для $f_1(x)$, а $\mathbf{G}_i(x)$ – матрица Гессе для $f_1(x)$, тогда выражение для градиента функции f и ее матрицы Гессе

$$\mathbf{g}(x) = \mathbf{I}^T(x) \mathbf{f}(x), \quad \mathbf{G}(x) = \mathbf{I}^T(x) \mathbf{I}(x) + \mathbf{Q}(x),$$

где $\mathbf{Q}(x) = \sum_{i=1}^m f_i(x) \mathbf{G}_i(x)$ в процессе итераций рано или поздно становится доминирующим.

Поскольку при $\mathbf{Q}_k \rightarrow 0$ $\|f_k\| \rightarrow 0$ в силу все более точной аппроксимации функции, $\varphi_0(x, t)$ будет функцией $\tilde{\varphi}$.

В этом случае ньютоновская схема имеет вид

$$(\mathbf{I}_k^T \mathbf{I}_k + \mathbf{Q}_k) \mathbf{d}_k = -\mathbf{I}_k^T f_k,$$

где \mathbf{d}_k – направление Ньютона.

Пренебрегая \mathbf{Q} , получаем следующее направление поиска $\overline{\mathbf{d}}_k = -(\mathbf{I}_k^T \mathbf{I}_k)^{-1}$ – направление Гаусса–Ньютона.

Метод может достигать квадратичной скоростной сходимости, несмотря на то, что при вычислениях используются только первые производные. В качестве недостатка метода можно отметить, что число обусловленности матрицы $\mathbf{I}_k^T \mathbf{I}_k$ равно квадрату числа обусловленности матрицы \mathbf{I}_k (т. е. возможны трудности при обращении матрицы).

2.5. Методы сопряженных градиентов

Эти методы относятся к классу алгоритмов, в основе которых лежит построение сопряженных направлений. Как уже отмечалось, они позволяют получить решение задач с квадратичными целевыми функциями примерно за n шагов (квадратично сходящиеся).

Рассматриваемый метод для получения сопряженных направлений использует квадратичную аппроксимацию $f(x)$ и значения компонент градиента, причем обеспечивается убывание целевой функции от итерации к итерации.

Итак, предполагается, что

$$f(x) = q(x) = a + b^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \mathbf{C} \mathbf{x};$$

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)} \mathbf{s}(\mathbf{x}^{(k)}).$$

Направление поиска на каждой итерации определяется из

$$\mathbf{s}^{(k)} = -\mathbf{g}^{(k)} + \sum_{i=0}^{k-1} \gamma^{(i)} \mathbf{s}^{(i)}, \quad \mathbf{s}^{(0)} = -\mathbf{g}^{(0)},$$

где $\mathbf{g}^{(k)} = \nabla f(\mathbf{x}^{(k)})$.

Видно, что здесь направление наискорейшего спуска отклоняется путем добавления к нему с положительными коэффициентами направлений, используемых на предыдущих шагах.

Значения $\gamma^{(i)}$, $i = 1, k-1$ выбираются так, чтобы оно было \mathbf{C} и сопряжено со всеми построенными ранее направлениями поиска. Из предыдущего следует, что

$$\mathbf{s}^{(1)} = -\mathbf{g}^{(1)} + \gamma^{(0)} \mathbf{s}^{(0)} = -\mathbf{g}^{(1)} - \gamma^{(0)} \mathbf{g}^{(0)}.$$

Используя условие сопряженности $\mathbf{s}^{(1)}$ с $\mathbf{s}^{(0)}$, т. е. $\mathbf{s}^{(1)T} \mathbf{C} \mathbf{s}^{(0)} = 0$, получим $[\mathbf{g}^{(1)} + \gamma^{(0)} \mathbf{g}^{(0)}] \mathbf{C} \mathbf{s}^{(0)} = 0$.

Учитывая, что $\mathbf{s}^{(0)} = \frac{\Delta x}{\alpha^{(0)}}$ и свойство квадратичных функций

$$\Delta \mathbf{g} = \mathbf{g}(\mathbf{x}^{(1)}) - \mathbf{g}(\mathbf{x}^{(0)}) = \mathbf{C} \Delta \mathbf{x}$$

(так как $\mathbf{g}(\mathbf{x}^{(0)}) = \mathbf{C} \mathbf{x}^{(0)} + \mathbf{b}$; $\mathbf{g}(\mathbf{x}^{(1)}) = \mathbf{C} \mathbf{x}^{(1)} + \mathbf{b}$), имеем $[\mathbf{g}^{(1)T} + \gamma^{(0)} \mathbf{g}^{(0)T}]^T \Delta \mathbf{g} = 0$, откуда $\mathbf{g}^{(1)T} \mathbf{g}^{(1)} + \gamma^{(0)} \mathbf{g}^{(0)T} \mathbf{g}^{(1)} - \mathbf{g}^{(1)T} \mathbf{g}^{(0)} - \gamma^{(0)} \mathbf{g}^{(0)T} \mathbf{g}^{(0)} = 0$.

Известно, что в качестве критерия окончания одномерного поиска вдоль каждого из сопряженных направлений, используется условие $\mathbf{g}^{(k+1)T} \mathbf{g}^{(k)} = 0$ (направление градиента в точке $k+1$ перпендикулярно направлению градиента на предыдущем шаге, поскольку

$\frac{d}{d\lambda} f(x^{(k)} + \lambda \mathbf{s}^{(k)}) = 0$ в конце линейного поиска, т. е. $\mathbf{s}^{(k)T} \mathbf{g}^{(k+1)} = 0$). В силу сказанного, подчеркнутые слагаемые уходят и, следовательно:

$$\gamma^{(0)} = \frac{\|\mathbf{g}^{(1)}\|^2}{\|\mathbf{g}^{(0)}\|^2}.$$

Затем определяются следующее направление $\mathbf{s}^{(2)}$ и т. д. В общем случае, сопряженные направления, предлагаемые методом Флетчера–Ривса, имеют вид

$$\mathbf{s}^{(k)} = -\mathbf{g}^{(k)} + \left[\frac{\|\mathbf{g}^{(k)}\|^2}{\|\mathbf{g}^{(k-1)}\|^2} \right] \mathbf{s}^{(k-1)}, \quad \gamma^{(i)} = 0, \quad i = \overline{0, k-2}.$$

Второе слагаемое $\gamma^{(k-1)} \mathbf{s}^{(k-1)}$ – добавка к направлению наискорейшего спуска направления только предыдущего шага.

Отметим, что метод эффективен (небольшие затраты по памяти) позволяют использовать его для решения задач большой размерности.

Пример

$$f(x) = 4x_1^2 + 3x_2^2 - 4x_1x_2 + x_1 \rightarrow \min, \quad x^{(0)} = [0, 0]^T.$$

$$\text{Шаг 1: } \nabla f(x) = [8x_1 - 4x_2 + 1, 6x_2 - 4x_1]^T;$$

$$\mathbf{s}^{(0)} = -\nabla f(x^{(0)}) = -[1, 0]^T.$$

Шаг 2 : линейный поиск

$$x^{(1)} = x^{(0)} - \alpha^{(0)} \nabla f(x^{(0)}) \rightarrow \alpha^{(0)} = \frac{1}{8};$$

$$x^{(1)} = (0, 0)^T - \frac{1}{8}(1, 0)^T = \left(-\frac{1}{8}, 0\right)^T.$$

Шаг 3 : $k=l$

$$s^{(1)} = -\left(0, \frac{1}{2}\right)^T - \frac{1}{4}(1, 0)^T = \left(-\frac{1}{4}, -\frac{1}{2}\right), \quad \gamma^{(0)} = \frac{\|g^{(1)}\|^2}{\|g^{(0)}\|^2} = \frac{1}{4}.$$

Шаг 4 : линейный поиск

$$x^{(2)} = x^{(1)} + \alpha^{(1)}S^{(1)} \rightarrow \alpha^{(1)} = \frac{1}{4};$$

$$x^{(2)} = \left(-\frac{1}{8}, 0\right)^T - \frac{1}{4}\left(\frac{1}{4}, \frac{1}{2}\right)^T = \left(-\frac{3}{16}, -\frac{1}{8}\right)^T.$$

$\nabla f(x^{(2)}) = (0, 0)^T$, таким образом, $x^{(k)} = x^*$.

Другой выбор параметра γ

$$\gamma^{(k)} = \frac{\Delta g(x^{(k)})^T g(x^{(k)})}{\|g(x^{(k-1)})\|}$$

(как и раньше, $\Delta g(x^{(k)}) = g(x^{(k)}) - g(x^{(k-1)})$).

В методе Полака–Рибьера (метод работает с целевыми функциями общего вида и менее чувствителен к ошибкам округления при линейном поиске).

2.6. Квазиньютоновские методы

Эти методы также основаны на свойствах квадратичных функций, хотя могут применяться к функциям общего вида. Поиск осуществляется по системе сопряженных направлений. Эти методы обладают положительными чертами метода Ньютона, но используют только первые производные. Итерация выполняется по формуле

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)}\mathbf{s}(x^{(k)}),$$

где $\mathbf{s}(x^{(k)}) = -\mathbf{A}^{(k)}\nabla f(x^{(k)})$ (сравните с методом Ньютона, где $\mathbf{s}(x^{(k)}) = -\mathbf{A}^{-1}(x^{(k)})\nabla f(x^{(k)})$). Важно отметить, что здесь нет необходимости на каждой итерации вычислять обратный гессиан. Он аппроксимируется последовательностью матриц $\mathbf{A}^{(k)}$ по формуле $\mathbf{A}^{(k+1)} = \mathbf{A}^{(k)} + \Delta\mathbf{A}^{(k)}$.

Матрица $\mathbf{A}^{(k)}$ [$n \times n$] носит название метрики, поскольку матрица не рассчитывается на каждой итерации, методы с такой формулой называются методами переменной метрики. Учитывая свойство квадратичных функций, можно записать $\Delta \mathbf{x} = \mathbf{C}^{-1} \Delta \mathbf{g}$. В данном случае хотелось бы, чтобы $\mathbf{A}^{(k+1)} = \mathbf{A}^{(k)} \Delta \mathbf{g}^{(k)}$, где $\Delta \mathbf{x}^{(k)} = \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}$, $\Delta \mathbf{g}^{(k)} = \mathbf{g}(\mathbf{x}^{(k+1)}) - \mathbf{g}(\mathbf{x}^{(k)})$. Однако построить такую аппроксимацию нельзя, так как, чтобы найти $\Delta \mathbf{g}^{(k)}$ надо знать $\mathbf{A}^{(k)}$ ($\Delta \mathbf{g}^{(k)}$ получено на основе $\Delta \mathbf{x}$, а оно по $\mathbf{A}^{(k)}$). Можно потребовать, чтобы новое приближение матрицы \mathbf{A} удовлетворяло этому соотношению, т. е.

$$\Delta \mathbf{x}^{(k)} = \beta \mathbf{A}^{(k+1)} \Delta \mathbf{g}^{(k)},$$

где β – скаляр. Раскрывая $\mathbf{A}^{(k+1)}$, получим $\Delta \mathbf{A}^{(k)} \Delta \mathbf{g}^{(k)} = \frac{1}{\beta} \Delta \mathbf{x}^{(k)} - \mathbf{A}^{(k)} \Delta \mathbf{g}^{(k)}$. Подстановкой можно убедиться, что $\Delta \mathbf{A}^{(k)} = \frac{1}{\beta} \times$

$$\times \left(\frac{\Delta \mathbf{x}^{(k)} \mathbf{y}^T}{\mathbf{y}^T \Delta \mathbf{g}^{(k)}} \right) - \frac{\mathbf{A}^{(k)} \Delta \mathbf{g}^{(k)} \mathbf{z}^T}{\mathbf{z}^T \Delta \mathbf{g}^{(k)}}$$

Здесь \mathbf{y} и \mathbf{z} – произвольные векторы, т. е. это семейство решений.

В известном методе Дэвидсона–Флетчера–Пауэлла

$$\mathbf{y} = \Delta \mathbf{x}^{(k)}, \quad \mathbf{z} = \mathbf{A}^{(k)} \Delta \mathbf{g}^{(k)}.$$

Таким образом, имеем

$$\mathbf{A}^{(k)} = \mathbf{A}^{(k+1)} + \frac{\Delta \mathbf{x}^{(k-1)T} \Delta \mathbf{x}^{(k-1)T}}{\Delta \mathbf{x}^{(k-1)T} \Delta \mathbf{g}^{(k-1)}} - \frac{\mathbf{A}^{(k-1)} \Delta \mathbf{g}^{(k-1)} \Delta \mathbf{g}^{T(k-1)} \mathbf{A}^{(k-1)}}{\Delta \mathbf{g}^{T(k-1)} \mathbf{A}^{(k-1)} \Delta \mathbf{g}^{(k-1)}},$$

такой выбор \mathbf{y} , \mathbf{z} связан с требованием сохранения симметричности и положительной определенности матрицы $\mathbf{A}^{(k)}$ (что связано с таким же требованием на все слагаемые рекуррентной формулы). В качестве $\mathbf{A}^{(0)}$ может быть выбрана единичная матрица.

Это требование является условием сходимости алгоритма к стационарной точке \mathbf{x}^* . Может быть показано, что

$$\Delta f(\mathbf{x}) = \nabla f(\mathbf{x}^{(k)})^T \Delta \mathbf{x} = -\nabla f(\mathbf{x}^{(k)})^T \alpha^{(k)} \mathbf{A}^{(k)} \nabla f(\mathbf{x}^{(k)}).$$

Отсюда следует, что если $\alpha^{(k)} > 0$ и матрица $\mathbf{A}^{(k)}$ – положительно определена (требование сходимости метода), то $\Delta f = f(\mathbf{x}^{(k+1)}) - f(\mathbf{x}^{(k)}) < 0$, $f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}^{(k)})$, т. е. функция убывает от итерации к итерации.

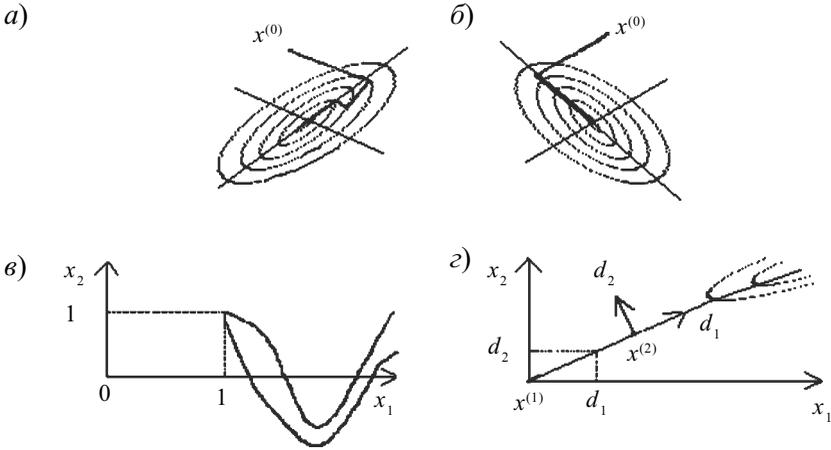


Рис. 12

Замечание. Симметричной положительно-определенной матрицей $\mathbf{A}^{(k)}$ аппроксимируют обратный гессиан, который не обязан быть симметричной положительно-определенной матрицей (хотя сам гессиан симметричная положительно-определенная матрица по определению), т. е. такая аппроксимация понимается лишь с точки зрения генерации одинаковой последовательности точек $\mathbf{x}^{(k)}$ (а элементы этих матриц могут сильно отличаться).

Для сохранения на практике свойств положительной определенности $\mathbf{A}^{(k)}$ (из-за ошибок счета) предпринимаются специальные меры (разного рода факторизации).

Отметим, что в случае квадратичной функции направления $\mathbf{s}^{(k)} = -\mathbf{A}^{(k)}\nabla f(x^{(k)})$ являются \mathbf{C} -сопряженными, а $\mathbf{A}^{k+1} = \mathbf{C}^{-1}$, совпадает с обратным гессианом в конце единственной генерации, приводящей в оптимальную точку.

Следует отметить также высокую эффективность метода. Он позволяет обойти вычисления и обращение матрицы Гессе.

Недостаток: необходимо хранить матрицу $\mathbf{A}^{(k)}$.

Упражнение

$$f(x) = 4x_1^2 + 3x_2^2 - 4x_1x_2 - x_1 \rightarrow \min_x, \quad x^{(0)} = (0,0)^T.$$

2.7. Методы минимизации овражных функционалов

Овражный функционал обладает следующими свойствами: почти вдоль всех направлений функционал резко увеличивается (склон оврага) и только в узком конусе направлений он слабо уменьшается (направление дна оврага). Локальный поиск носит рыскающий характер (рис. 12, в).

Метод Розенброка

Метод Розенброка используется при минимизации овражных функционалов, если овраг одномерный (рис. 12, з). Размерность дна оврага определяется числом малых собственных значений матрицы Гессе.

Идея метода Розенброка состоит в организации спуска не вдоль фиксированных координатных ортов, а вдоль осей специальным образом выбираемой системы координат. Одна из осей должна составлять достаточно малый угол с образующей дна одномерного оврага. Тогда смещения по этой оси совпадают с продвижением вдоль дна к точке минимума.

Пусть \mathbf{x}^{m-1} и \mathbf{x}^m – две соседние точки в последовательности $\{\mathbf{x}^k\}$. Тогда переход к \mathbf{x}^{m+1} точке осуществляется следующим образом.

1. Выбрать новую систему координат, первая ось которой направлена вдоль вектора $(\mathbf{x}^m - \mathbf{x}^{m-1})$, а остальные дополняют ее до ортонормированного базиса (“поворот осей”).

2. В новой системе координат для поиска точки \mathbf{x}^{m+1} осуществить покоординатный спуск до выполнения условия поворота осей (пока по всем направлениям не получим неудачу, т. е. возрастание целевой функции).

3. Возвратиться к старой системе и перейти к шагу 1.

Рассмотрим более подробно.

1. Пусть d_i – линейно независимые векторы (т. е. $\sum_{j=1}^n \lambda_j d_j = 0$, где λ – длина шага по d_j ; при всех $\lambda_j=0$), причем $\|d_j\| = 1$.

2. Кроме того, d_j взаимно-ортогональны, т. е. $d_i^T d_j = 0$, $i \neq j$, x_1, x_2 – две соседние точки. Вектор $(x_2 - x_1)$ – как и в методе Хука–Дживиса –

вектор, направленный вдоль дна оврага. Новая $\mathbf{x}^{m+1} = \mathbf{x}^m + \sum_{j=1}^m \lambda_j d_j$ (λ_j

– длина шага в направлении d_j). Шаги λ_j определяются линейным поиском или специальными алгоритмами. Новый набор направлений d_j ,

строится с помощью процедуры Грамма–Шмидта (построение ортонормированного базиса).

Следует отметить, что при наличии многомерных оврагов эффективность метода теряется, поскольку целенаправленно изменяется ориентация одной оси в процессе движения по дну оврага.

Недостатком метода является невозможность продолжения оптимизации, если в качестве начальной точки выбрана “точка заклинивания” (рис. 13, а). Если процесс попадает в точку A , тогда ни в одной из ближайших точек B, B, C, C не происходит уменьшение функции.

Метод обобщенного покоординатного спуска

Прежде всего отметим, что овражные функционалы отличает наличие нескольких изолированных групп собственных значений матрицы Гессе $\mathbf{I}''(x)$. Для овражного функционала $\mathbf{I}(x)$ выполняется следующее правило: для любого x из множества D для собственных чисел $\mathbf{I}''(x)$ имеем, $\lambda_1 \geq \dots \geq \lambda_{n-r} \gg |\lambda_{n-r+1}| \geq \dots \geq \lambda_n$, где r – число малых собственных значений (определяет размерность дна оврага).

Основной процедурой является приведение $\mathbf{I}''(x)$ к главным осям квадратичной формы, т. е. диагонализация \mathbf{I}'' с последующим покоординатным спуском вдоль собственных векторов матрицы \mathbf{I}'' . Направления, определяемые этими векторами совпадают с осями наилучшей системы координат при минимизации квадратичных функционалов (независимо от их выпуклости).

В процессе работы алгоритма новые оси (соответствующие собственным векторам \mathbf{I}'') вычисляются по каждой группе изолированных собственных значений. Переход к новым осям осуществляется, когда старые оси “исчерпали себя”, т. е. вслед за успешным продвижением последовала неудача – возрастание функции $\mathbf{I}(x)$. Обновление осей происходит после того, как по каждому координатному направлению последовала неудача.

Приведем алгоритм описанного метода

Шаг 1: ввод исходных данных: \mathbf{x}_0, S_0 (начальная точка и шаг), N_{\max} (максимальное число итераций), $k = 0$ – счетчик числа шагов

$$\mathbf{x}^{(k)} = \mathbf{x}_0, (x_i, i = \overline{1, n});$$

$S^{(k)} = S_0$ – начальный шаг для вычисления производных;

$h_0 = S_0$ – начальный шаг для координатного спуска.

Шаг 2: $\mathbf{U} = \mathbf{E}$, где \mathbf{E} – единичная матрица.

В качестве координатных векторов взять столбцы $\{u_i\}$ матрицы \mathbf{U} .

Шаг 3: построить симметричную матрицу Гессе $\mathbf{B} = \{b_{ij}\}$,

$$b_{ij} = I(x^{(k)} + s^{(k)}u_i + s^{(k)}u_j) - I(x^{(k)} - s^{(k)}u_i + s^{(k)}u_j) - \\ - I(x^{(k)} + s^{(k)}u_i - s^{(k)}u_j) + I(x^{(k)} - s^{(k)}u_i - s^{(k)}u_j),$$

$$k=k-1.$$

Шаг 4: $\mathbf{U} = \mathbf{U}\mathbf{T}$ (\mathbf{T} – ортогональная матрица), приводящая \mathbf{B} к диагональному виду $\mathbf{T}^T \mathbf{B} \mathbf{T}$ (\mathbf{T} определяется, например, алгоритмом Якоби).

Шаг 5: в осях $\{u_i\}$ реализовать процесс покоординатного спуска из точки $x^{(k-1)}$ до выполнения условия поворота осей, а именно:

$$a) \quad P_j=2; h_j=h_0; j = \overline{1, n}; f = I(x^{(k-1)}); x_{\text{old}} = x^{(k-1)},$$

где P – массив признаков для хранения информации о том, по всем ли осям получили возрастание функции;

б) $i = 1$;

в) $x^{(k-1)} = x^{(k-1)} + h_i u_i$; $f_1 = I(x^{(k-1)})$, если $f_1 \leq f$, то $\{h_i = 3h_i; f = f_1$;
если $p_i = 2$, то $p_i = 1\}$,

иначе

$$\{x^{(k-1)} = x^{(k-1)} - h_i u_i; h_i = -0,5h_i; \text{if } (p_i \neq 2), p_i = 0\};$$

если $p_i \neq 0$, $j = \overline{1, n}$, то $\{\text{если } i=n \text{ идти на шаг 5, б), иначе } \{i=i+1; \text{ на шаг 5, в)}\}$;

$$г) s^{(k)}=0,1 ||x^{(k-1)} - x_{\text{old}}||.$$

Если $k \leq N_{\text{max}}$ на шаг 3 (поворот осей), иначе на конец (таким образом, поиск осуществляется до тех пор, пока не превысим заданное число шагов).

Пояснение к шагу 4. Диагональное преобразование матрицы Гессе эквивалентно выбору в качестве новых направлений осей столбцов \mathbf{U}^T , т. е. выбор в качестве координатных осей u_i эквивалентен замене переменных

$$\mathbf{x} = \mathbf{U}\mathbf{y} \quad I(\mathbf{x}) = I(\mathbf{U}\mathbf{y}) = \bar{I}(\mathbf{y}).$$

Можно показать, что $\bar{\mathbf{I}}''(\mathbf{y}) = \mathbf{U}^T \mathbf{I}''(\mathbf{x}) \mathbf{U}$, поэтому, если требуется изменить матрицу Гессе $\bar{\mathbf{I}}''(\mathbf{y})$ и привести ее к диагональному виду $\mathbf{T}^T \bar{\mathbf{I}}''(\mathbf{y}) \mathbf{T} = \mathbf{T}^T \mathbf{U}^T \mathbf{I}''(\mathbf{x}) \mathbf{U} \mathbf{T}$, т. е. в качестве новых направлений достаточно выбрать столбцы матрицы $\mathbf{U}_1 = \mathbf{U}\mathbf{T}$.

2.8. Практические вопросы

В данном разделе рассмотрим некоторые практические вопросы, важные с точки зрения численного исследования функции.

Анализ чувствительности

Как влияют вариации \mathbf{x}^k (или точность получения оптимального решения) на значение функции f^* ? Если можем вычислить матрицу вторых производных $\mathbf{A} = \nabla^2 f(\mathbf{x}^*)$ в точке оптимума, то решив полную проблему собственных значений для матрицы \mathbf{A} (т. е. определив собственные числа λ_i и собственные векторы \mathbf{u} (связаны уравнением $\lambda \mathbf{u} = \mathbf{A}\mathbf{u}$), можем ответить на этот вопрос. Существует алгоритм, который по множествам λ_m (малые собственные значения) и соответствующие им \mathbf{u}_m , позволяет найти элементы вектора \mathbf{x} функции $f(\mathbf{x})$, которые можно сохранить, и те, которые можно исключить без существенной потери точности.

Физическое объяснение этого заключается в том, что в овражной ($|\lambda_{\max}| / |\lambda_{\min}| \gg 1$) ситуации линии уровня функции имеют вытянутую форму (рис. 13, в), т. е. при одних и тех же значениях функции f изменения x_1 и x_2 различны (f менее чувствительна к x_1 , чем к x_2). Изменяя x_1 до 0, т. е. исключая этот параметр, можем не превысить порог $\delta f \leq \delta f^{\text{доп}}$ для допустимого изменения функции $\delta f \leq \delta f^{\text{доп}}$. Для сравнения, в ситуации (рис. 13, з) не очевидно, что можно исключить: x_1 или x_2 .

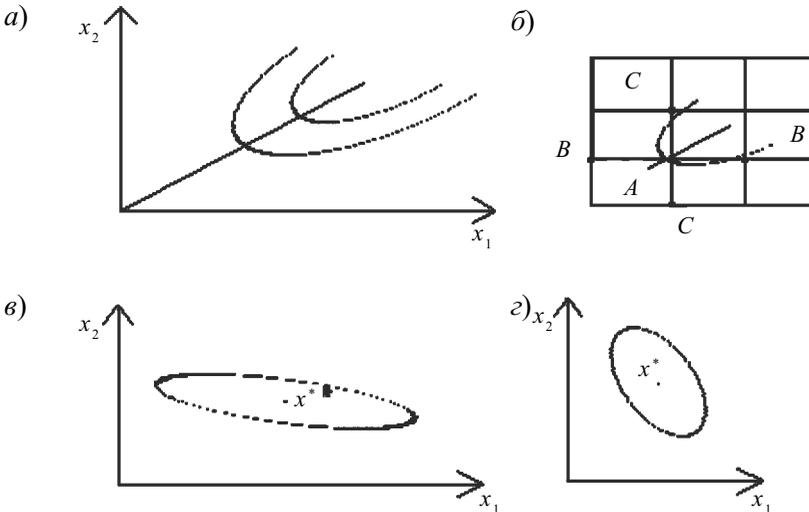


Рис. 13

Цель такого анализа может, например, состоять в том, чтобы упростить исходную математическую модель $f(x)$, упростив тем самым вычислительную задачу. Исключение избыточных параметров из критерия качества позволяет упростить избыточную структуру проектируемой системы.

Проверка решения на оптимальность

При данной проверке можно использовать следующие приемы:

- проверку необходимых и достаточных условий оптимальности в найденной точке;
- повторное решение с другими начальными данными;
- попытку уменьшить значение функции, используя альтернативные методы поиска.

Важнейшим фактором является выбор начального приближения. Для этой цели обычно используют методы глобального поиска (например, случайный поиск), методы, хорошо сходящиеся вдали от экстремума (метод Коши).

Критерии останова

Прежде всего, заметим, что остановка работы алгоритма может быть точкой перехода к более перспективной вычислительной процедуре.

Наиболее распространены следующие критерии останова итерационной процедуры:

$$\frac{\|x^{(k+1)} - x^{(k)}\|}{\|x^{(k)}\|} < \varepsilon -$$

относительное смещение в результате данной итерации стало $< \varepsilon$, где $(\|x\| = \sqrt{\sum x_i^2})$ (суммарное перемещение после N последовательных итераций $< \varepsilon$);

$\frac{f(x^{(k)}) - f(x^{(k+1)})}{|f(x^{(k)})|} < \varepsilon$, т. е. относительное убывание значения функции на данной итерации $< \varepsilon$.

Из примера на рис. 14, а хорошо видно, почему важно удовлетворение условий на функцию и на аргумент. В области Ω_1 после N_{\max} итераций выполняется условие на близость значений функции $|f_k - f_{k-1}|$, а в области Ω_2 – на близость значений аргументов $|f_k - f_{k-1}|$. В точке x^* выполняются оба условия.

При минимизации овражных функционалов следует иметь в виду, что чем выше число обусловленности (cond), тем осторожнее надо от-

носиться к используемым критериям качества решения. Речь идет о числе обусловленности матрицы Гессе, которое может быть оценено следующим образом :

$$а) 1 \leq \eta(x) = \text{cond} \left[\nabla^2 f(x) \right] = \frac{\max_i \lambda_i(x)}{\min_i \lambda_i(x)},$$

если $\nabla^2 f(x) > 0$ (тогда $\lambda_i > 0$), где λ_i – собственное значение матрицы $\nabla^2 f(x)$, если $\eta > 10^2$, задача плохо обусловлена (овражный функционал);

$$б) \eta \approx \frac{2}{1-\mu}, \text{ где } \mu \text{ установившееся значение отношения}$$

$$\frac{\|\nabla f(x^{(k+1)})\|}{\|\nabla f(x^{(k)})\|}.$$

Рассмотрим пример, демонстрирующий опасность использования некоторых критериев останова в овражных ситуациях. Пусть

$$f(x_1, x_2) = (x_1 - 1)^2 + 10^{-6}(x_2 - 1)^2 + 1;$$

$x^* = [1, 1]^T$ – оптимальное решение $f^* = 1$.

Рассмотрим две точки:

$$1. \bar{x} = [1, 2]^T, \quad f(\bar{x}) = 1 + 10^{-6}, \quad \|\bar{x} - x^*\| = 1.$$

$$2. \hat{x} = [1 + 10^{-3}, 1]^T, \quad f(\hat{x}) = 1 + 10^{-6}, \quad \|\hat{x} - x^*\|_2 = 10^{-3}.$$

Таким образом, по критерию нормы разности (удаленность одной точки от другой) точка \hat{x} – лучшая, но $\|\mathbf{g}(x)\|_2 = 2 \cdot 10^{-6}$ (\mathbf{g} – градиент), а $\|\mathbf{g}(\hat{x})\|_2 = 2 \cdot 10^{-3}$, т. е. $\|\mathbf{g}(\bar{x})\|_2$ (ближе к нулю) оказывается лучше точка \mathbf{x} , что, конечно, неверно.

Надо учесть, что малость градиента характерна для дна оврага (движение вдоль дна происходит всегда с малым градиентом) и на градиент нельзя ориентироваться в плохо обусловленных задачах.

Численная аппроксимация градиентов

При отсутствии возможности получения аналитических выражений для производных можно использовать различного рода аппроксимации ∇f , $\nabla^2 f$ конечными разностями

$$\frac{df}{dx_i} \Big|_{x=\bar{x}} = \frac{f(x + h e^{(i)}) - f(\bar{x})}{h}$$

или точнее

$$\left. \frac{df}{dx_i} \right|_{x=x} = \frac{f(\bar{x} + he^{(i)}) - f(\bar{x} - he^{(i)})}{2h},$$

где h – шаг аппроксимируемой производной; $e^{(i)}$ – вектор с единицей в позиции i .

Во втором случае требуется дополнительное вычисление значения функции

$$\begin{aligned} \frac{\partial^2 f}{\partial x_i \partial x_j} = & \frac{f(\bar{x} + he^{(i)} + he^{(j)}) - f(\bar{x} - he^{(i)} + he^{(j)})}{4h} + \\ & + \frac{-f(\bar{x} + he^{(i)} - he^{(j)}) + f(\bar{x} - he^{(i)} - he^{(j)})}{4h} \end{aligned}$$

– для вычисления матрицы Гессе.

Точность повышается с уменьшением h , но знаменатель нельзя уменьшать бесконечно. Если h увеличивается, получаем “хорошие оценки” “с точки зрения расчетов на ЭВМ, но плохие оценки производных. Выбор h должен осуществляться в зависимости от вида $f(x)$, координат точки x и точности ЭВМ. Для определения шага h можно использовать величину продвижения в пространстве переменных, процент от состояния $h = 0,1 \|x^{(i+1)} - x^{(i)}\|$. Для оценки производной на первом шаге надо знать h_0 .

Скорость сходимости последовательностей

Последовательность $\{x_k\}$ называется сходящейся с порядком r , если r – максимальное число, для которого

$$0 \leq \lim_{k \rightarrow \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^r} = \gamma < \alpha,$$

где x^* – предел последовательности; $\gamma = \text{const}$. Если $r = 1$ – линейная скорость сходимости, то на практике это означает, что через каждые три итерации в x_k появляется дополнительная правильная цифра. Если $r = 2$ – квадратичная скорость сходимости, то с каждым шагом число правильных цифр у x_k удваивается.

Если $\gamma = 0$, то последовательность сходится “суперлинейно” (для линейно сходящихся последовательностей).

Многоэкстремальность

Следует отметить, что проблема многоэкстремальности возникает часто по причине непонимания реальной ситуации. Регистрируемые на практике многочисленные экстремумы в действительности оказываются точками останковки применяемых поисковых процедур. На самом деле наличие многих локальных минимумов в практических задачах встречается реже, чем об этом принято говорить. Тем не менее использование процедур глобального поиска в ряде случаев является необходимым.

В процессе глобального поиска должны решаться две противоречивые задачи: искать каждый конкретный минимум и одновременно уклоняться от него, чтобы найти наименьший. Причем нет уверенности, что найденный минимум является глобальным, поскольку время поиска ограничено.

Часто глобальный минимум находится на “дне оврага” минимизируемого функционала $f(x)$. Один из приемов основан на сглаживании траектории поиска (в этой ситуации она имеет рыскающий характер, следовательно, ее надо сгладить).

На сферическом (радиусе R) дне конуса с вершиной в $\mathbf{x}^{(k)}$ (вектор) и осью $\mathbf{V}^{(k)}$ делают m случайных проб $x_1^{(k)}, \dots, x_m^{(k)}$ (рис. 14, б). Следующее $(k+1)$ – состояние вектора определяется по наилучшей пробе $\mathbf{x}^{(k+1)} = \arg \min f(x_i^{(k)})$, а ось следующего конуса выбирается вдоль сделанного рабочего шага в соответствии с $\mathbf{V}^{(k+1)} = \frac{1}{R}(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)})$, где R – нормирован.

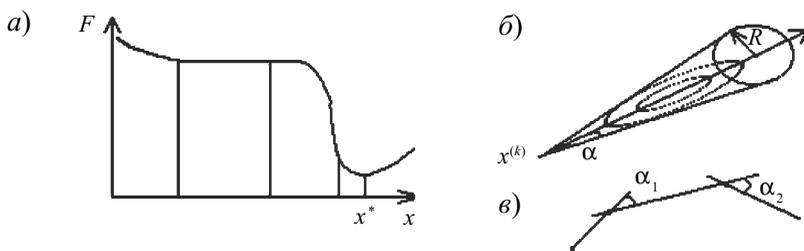


Рис. 14

Траектория такого поиска позволяет отслеживать направление оврага независимо от того, вверх или вниз идет этот овраг. Варьируя R , можно воздействовать на гладкость траектории поиска (адаптировать поиск) (рис. 14, в).

3. МЕТОДЫ УСЛОВНОЙ ОПТИМИЗАЦИИ

Большинство практических задач связано с оптимизацией при наличии некоторого количества ограничений на управляемые переменные. Существующие ограничения существенно уменьшают размеры области, в которой производится поиск оптимума.

3.1. Критерии оптимальности в задачах с ограничениями

Процесс оптимизации становится более сложным, так как при наличии ограничений нельзя использовать критерии оптимальности безусловной оптимизации. Может нарушаться даже основное условие – равенство нулю градиента в стационарной точке, как, например, в задаче

$f(x) = (x - 2)^2 \rightarrow \min_x$ имеем $x^* = 2$, а при введении ограничения $x \geq 4$ будет найден условный минимум $x^* = 4$. Заметим, что при этом $f'(4) = 4 \neq 0!$

Ограничения в виде равенств.

Рассмотрим задачу $f(x) \rightarrow \min$ при $h_k(x) = 0$; $k = \overline{1, K}$. Если ограничения можно разрешить относительно k независимых переменных (аналитически) и затем их исключить из функции f , то применимы методы безусловной оптимизации, рассмотренные выше.

Если это трудно или невозможно сделать, то используют метод множителей Лагранжа. При этом осуществляется преобразование в эквивалентную задачу безусловной оптимизации

$$f(x) \rightarrow \min ; \quad (3.1)$$

$$\text{при } h_1(x) = 0 \quad (3.2)$$

преобразуется в

$$L(x, v) = f(x) - v h_1(x) \rightarrow \min, \quad (3.3)$$

где L – функция Лагранжа; v – неизвестная постоянная (множитель Лагранжа), на знак которой не накладываются требования. Исходная задача будет решена, если для всех x удовлетворяется (т. е. при $h_1(x) = 0$)

$$\min_x L(x, V) = \min_x f(x).$$

Задача в том, чтобы подобрать значение v^0 и в точке безусловного минимума x^0 функции L , удовлетворялось ограничение (3.2). Тогда x^0 , v^0 – будет решением задачи.

Алгоритм

Найти безусловный минимум функции L по x , как функции переменной v и выбрать такое v , которое удовлетворяет ограничению.

Пример: $f(x) = x_1^2 + x_2^2$; $h_1(x) = 2x_1 + x_2 - 2 = 0$;

$$L(x, v) = x_1^2 + x_2^2 - v(2x_1 + x_2 - 2);$$

$$\frac{dL}{dx_1} = 2x_1 - 2v = 0 \rightarrow x_1^0 = v;$$

$$\frac{dL}{dx_2} = 2x_2 - v = 0 \rightarrow x_2^0 = \frac{v}{2}.$$

Матрица $\mathbf{H}_L = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$ положительно-определенная, следовательно, функция выпукла (это точка глобального минимума). Подставляем x^0 в $h_1(x)$, так как должны удовлетворить ограничениям

$$2v + \frac{v}{2} = 2 \rightarrow v^0 = \frac{4}{5}.$$

Таким образом, имеем результат

$$x_1^0 = \frac{4}{5}, x_2^0 = \frac{2}{5}, \min f(x) = \frac{4}{5}.$$

При наличии K штук ограничений имеем $L(x, v) = f(x) - \sum_{k=1}^K v_k h_k$.

Если не удастся найти решение системы $\frac{dL}{dx} = 0$, как функции вектора \mathbf{V} , то ее надо дополнять ограничениями, так как $\frac{dL}{dv} = L(x) = 0$.

Необходимые условия минимума функции $f(x)$

$$\left\{ \begin{array}{l} \frac{dL}{dx_j} = \frac{df}{dx_j} - \sum_{k=1}^K \delta_k \frac{dh_k}{dx_j} = 0, \quad j = \overline{1, n}, \quad k = \overline{1, K}; \\ \frac{dL}{dv_k} = h_k(x) = 0. \end{array} \right. \quad (3.5)$$

Для выполнения достаточных условий требуется положительная определенность $\frac{dL^2}{dx_i dx_j}$ в точке минимума.

Замечания:

а) уравнение (3.5) может не иметь решений ;

б) вектор (x^0, v^0) является седловой точкой функции L , поэтому нельзя решать задачу $\min L(x, v)$ стандартными методами безусловной минимизации.

Рассмотрим решение описанной выше задачи на примере.

Пусть требуется изготовить 180 изделий. Их можно изготовить двумя технологическими способами. Затраты связаны функциональной зависимостью

способ 1: $4x_1 + x_1^2,$

способ 2: $8x_2 + x_2^2,$

где x_1 – число изделий, изготовленных первым способом; x_2 – изготовленных вторым способом;

Определить сколько изделий может быть изготовлено каждым способом так, чтобы суммарные затраты были минимальны?

Найденное решение соответствует максимуму или минимуму

$$f(x) = 4x_1 + x_1^2 + 8x_2 + x_2^2 \rightarrow \min;$$

$$x_1 + x_2 = 180;$$

$$L(x, V) = x_1^2 + 4x_1 + 8x_2 + x_2^2 - v_1(x_1 + x_2 - 180);$$

$$\frac{dL}{dx_1} = 2x_1 + 4 - v = 0;$$

$$\frac{dL}{dx_2} = 2x_2 + 8 - v = 0;$$

$$\frac{dL}{dx_3} = x_1 + x_2 - 180 = 0; \quad \frac{d^2L}{dx_1^2} = 2; \quad \frac{d^2L}{dx_2^2} = 2;$$

$$H = \begin{vmatrix} 2 & 0 \\ 0 & 2 \end{vmatrix},$$

где $2x_1 = 182; x_2 - x_1 = 2; x_1 = 91; x_1 + x_2 = 180; x_2 = 89.$

Решить задачи.

$$1. \quad \begin{aligned} f(x) &= x_1^2 + x_2^2 + x_3^2 \rightarrow \min; \\ g_1(x) &= x_1 + x_2 + 3x_3 - 2 = 0; \\ g_2(x) &= 5x_1 + 2x_2 + x_3 - 5 = 0. \end{aligned}$$

$$2. \quad \begin{aligned} f(x) &= x_1 + x_2 \rightarrow \min; \\ x_1^2 + x_2^2 &= 1; \\ x_1^* &= x_2^* = \lambda_1^* = \frac{-1}{\sqrt{2}}. \end{aligned}$$

3.2. Экономическая интерпретация множителей Лагранжа

Множители v_k можно интерпретировать, как цены ресурсов, определяемых ограничениями. Оптимальные значения v_k^* — коэффициенты чувствительности значения целевой функции в точке оптимума к ресурсам. Покажем это.

Пусть $f(x_1, x_2) \rightarrow \min$; $h_1(x_1, x_2) = b_1$, где b_1 — ресурсы (постоянные в ограничениях равенствах).

Запишем функцию Лагранжа $L(x_1, v_1) = f(x) - v_1(h_1(x) - b_1)$. Нас интересует изменение f^* , связанное с изменением b_1 , т. е.

$$\frac{df^*}{db_1} = \frac{df^*}{dx_1^*} \frac{dx_1^*}{db_1} + \frac{df^*}{dx_2^*} \frac{dx_2^*}{db_1}. \quad (3.6)$$

Аналогично, интересуемся изменением ограничений. Дифференцируя $h_1(x) - b_1 = 0$, получим

$$\frac{\partial h_1}{\partial x_1} \cdot \frac{\partial x_1^*}{\partial b_1} + \frac{\partial h_1}{\partial x_2} \cdot \frac{\partial x_2^*}{\partial b_1} - 1 = 0. \quad (3.7)$$

Умножим обе части (3.7) на v_1^* и вычтем из (3.6)

$$\frac{\partial f^*}{\partial b_1} = v_1^* + \sum_{j=1}^2 \left(\frac{\partial f^*}{\partial x_j^*} - v_1^* \frac{\partial h_1}{\partial x_j} \right) \frac{\partial x_j^*}{\partial b_1},$$

$$\text{т. е. } \frac{\partial f^*}{\partial b_1} = v_1^*,$$

где v_1^* – скорость изменение оптимального значения f , вызываемого изменением b_1 . Фактически, v_1^* можно интерпретировать, как стоимость единицы ресурсов, т. е. как цену ресурса.

В зависимости от v_1^* при изменении b_1^* значение f^* увеличивается или уменьшается.

3.3. Условия оптимальности Куна–Таккера

Кун и Таккер обобщили описанный выше подход к решению задачи с ограниченными равенствами на случай общей задачи нелинейного программирования с ограничениями-равенствами и неравенствами. В общем случае задача имеет вид

$$f(x) \rightarrow \min, \quad g_j(x) \geq 0, \quad j = \overline{1, I}, \quad h_k(x) = 0. \quad (3.8)$$

Если область D выпуклая и имеет внутренние точки, то существует хотя бы одна точка $x \in D$, в которой все ограничения $g_j \geq 0$ могут быть разделены на два типа.

1. Активные (в которых эти неравенства выполняются как равенства) $g_j(x)=0, j \in I^-$.
2. Неактивные, где $g_j(x)>0, k \in I^+$.

В последнем случае точка x не лежит на поверхности ограничения. Поэтому из этой точки ненулевой шаг может быть выполнен в любом направлении. Напротив, если j -е ограничение активно в точке x , то движение возможно лишь в строго определенном направлении.

Если бы можно было заранее (до решения) обнаружить неактивные в точке оптимума ограничения, то их можно бы было исключить из модели и таким образом уменьшить ее размерность.

Пример

Точка оптимума лежит на кривой ограничений g_1 , (рис. 15, а, б), т. е. при подстановке x^*_{opt} в g_1 оно обратится в равенство

$$(g_1=0), \quad \text{а } g_2, g_3 - \text{ неактивны,} \\ g_1(x^*) = 0, \quad g_2(x^*) > 0, \quad g_3(x^*) > 0.$$

Условия оптимальности Куна–Таккера можно сформулировать в виде задачи нахождения решения некоторой системы нелинейных уравнений и неравенств.

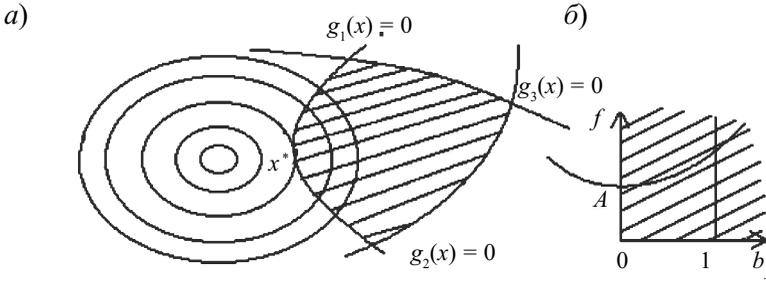


Рис. 15

Найти $x \in R^n$ и $u \in R^I$, $v \in R^K$, удовлетворяющие условиям

$$\nabla f(x) - \sum_{j=1}^I u_j \nabla g_j(x) - \sum_{k=1}^K v_k \nabla h_k(x) = 0;$$

$$g_j(x) \geq 0; h_k(x) = 0; u_j g_j(x) = 0; u_j \geq 0, j=1, I; k=1, K. \quad (3.9)$$

Сразу видно, что без неравенств эти условия совпадают с условиями оптимальности для задачи Лагранжа (они из них и получены). Учитывая неравенства, отметим, что если предположить j -е ограничение неактивным ($g_j(x) > 0$), то минимум, который удовлетворяет неактивному ограничению, можно обнулить, откуда $u_j g_j(x) = 0$.

В случае активного ограничения, т. е. когда $g_j(x) = 0$, u не обязательно равны нулю, но так как $g_j(x) = 0$, то $u_j g_j(x) = 0$ (эти условия называются условиями дополняющей нежесткости).

Рассмотрим вопрос о знаках u_j . Пусть $g_j(x) \geq 0$. Преобразуем неравенства в равенства введением ослабляющей переменной

$$a_j^2 > 0; g_j(x) - a_j^2 - b_j = 0; a_j^2 > 0 \quad (\text{здесь рассматривается точка } j \text{ } b_j = 0).$$

Учитывая предыдущие рассмотрения при ограничениях-равенствах, можем записать

$$\left. \frac{\partial f^*}{\partial b_j} \right|_{b=0} = u_j - \text{цена ресурса (в нашем случае } b_j = 0).$$

Теперь увеличивая b_j от 0 до 1, видим, что область сужается (рис. 15, б); A – точка минимума, а значит, f^* может только возрастать, так как минимум находится внутри дополнительной области, следовательно, $u_j \geq 0$, (так как если b_j возрастает, то и f^* возрастает).

Видим, что u_j не отрицательны, как и положено быть.

Теорема 1 (необходимые условия)

Рассмотрим задачу (3.8). Пусть $\mathbf{f}, \mathbf{g}, \mathbf{h}$ – дифференцируемые функции, а \mathbf{x}^* – допустимое решение задачи. Положим $I = \{j \mid g_j(\mathbf{x}^*) = 0\}$,

т. е. множество индексов активных ограничений.

Пусть $\nabla g_j(\mathbf{x}^*)$ при $j \in I$ и $\nabla h_k(\mathbf{x}^*)$ при $k = \overline{1, K}$ линейно-независимы. Если \mathbf{x}^* – оптимальное решение задачи (3.8), то существует пара векторов $\mathbf{u}^*, \mathbf{v}^*$ и она является решением задачи Куна–Таккера (3.9). Это точка Куна–Таккера.

Напомним, что векторы $\nabla \mathbf{g}_j$ линейно-независимы, если $\sum_{j=1}^I c_j \nabla \mathbf{g}_j = 0$ только при всех $x_i = 0$. Проверить это трудно, так как надо уже знать оптимальную точку. Эту теорему можно использовать для доказательства того, что заданная допустимая точка, удовлетворяющая условию линейной независимости, не является оптимальной, если не удовлетворяет условиям Куна–Таккера.

Отметим, что условия линейной независимости всегда удовлетворяются, если все ограничения в виде равенств и неравенств содержат только линейные функции.

Например, в случае $f(x) = 1 - x^2 \rightarrow \min$;

$$-1 \leq x \leq 3;$$

$$g_1(x) = x + 1 \geq 0;$$

$$g_2(x) = 3 - x \geq 0.$$

Допустим, что найдено $x^* = 3$.

Условия Куна–Таккера: $-2x - u_1 + u_2 = 0$;

$$\left. \begin{array}{l} -1 \leq x \leq 3, \\ u_1(x+1) = 0, \\ u_2(3-x) = 0, \\ u_1, u_2 \geq 0 \end{array} \right\} \rightarrow x^* = 3, u_1 = 0, u_2 = 6 > 0,$$

т. е. выполняется условие первой теоремы и точка $x = 3$ может быть точкой оптимума, однако необходимость условий не гарантирует, что это точка оптимума.

Теорема 2 (достаточные условия)

Рассмотрим задачу (3.8). Пусть функция $f(x)$ выпуклая, а все ограничения в виде неравенств содержат вогнутые функции $g_j(x), j=\overline{1, I}$, ограничения в виде равенств содержат линейные функции $h_k(x), k=\overline{1, K}$. Тогда, если существует решение $\mathbf{x}^*, \mathbf{u}^*, \mathbf{v}^*$, удовлетворяющее условиям Куна–Таккера (3.9), то \mathbf{x}^* – оптимальное решение задачи (3.8).

Теорему можно использовать для доказательства того, что найдено оптимальное решение.

Пример

$$\min f(x) = x_1^2 - x_2,$$

при

$$g_1(x) = x_1 - 1 \geq 0;$$

$$g_2(x) = 26 - x_1^2 - x_2^2 \geq 0;$$

$$h_1(x) = x_1 + x_2 - 6 = 0.$$

Надо проверить условия теоремы 2.

Имеем $\nabla f(x) = (2x_1, -1) \mathbf{H}_f(x) = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}$ – положительно полуопределена, следовательно, f – выпукла. Функция $g_1(x)$ выпукла и вогнута

одновременно, функция $g_2(x)$ – вогнута, так как $\mathbf{H}_g(x) = \begin{pmatrix} -2 & 0 \\ 0 & -2 \end{pmatrix}$, – отрицательно определена; $h_1(x)$ – содержит линейные функции, т. е. условия теоремы 2 выполнены.

Если теперь добавить проверку необходимых условий, то можно показать оптимальность решения.

Многие методы нелинейного программирования сходятся к точке Куна–Таккера. Если условия теоремы 2 выполнены, то это точка глобального минимума.

Упражнение

1. $f(x) = -x_1 \rightarrow \min; \mathbf{g}_1(x) = (1 - x_1)^3 - x_2 \geq 0.$

Проверяется точка $\bar{\mathbf{x}} = (1, 0).$

Имеем $\nabla f(\bar{\mathbf{x}}) = (-1, 0)^T;$

$\nabla \mathbf{g}_1(\bar{\mathbf{x}}) = (0, -1)^T;$

$$\nabla \mathbf{g}_2(\bar{\mathbf{x}}) = (0, 1)^T;$$

$c_1 \nabla \mathbf{g}_1(\bar{\mathbf{x}}) + c_2 \nabla \mathbf{g}_2(\bar{\mathbf{x}}) = 0$ не только при $c_1 = c_2 = 0$, поэтому $\nabla \mathbf{g}_1$ и $\nabla \mathbf{g}_2$ линейно зависимы, а значит, условия Куна–Таккера не выполняются.

$$2. f(x) = (x_1 - 3)^2 + (x_2 - 2)^2 \rightarrow \min; 5 - x_1^2 - x_2^2 \geq 0; 4 - x_1 - 2x_2 \geq 0.$$

Проверить точку а) $\bar{\mathbf{x}} = (2, 1)^T$ и б) $\bar{\mathbf{x}} = (0, 0)^T$.

Решение

1. В этой точке множество индексов активных ограничений $I = \{1, 2\}$. Следовательно, в соответствии с требованием дополнительной нежесткости $u_3 = u_4 = 0$, так как

$$\nabla f(x) = (-2, -2)^T; \nabla g_1(x) = (-2, -2)^T; \nabla g_2(x) = (-4, -2)^T, \text{ то}$$

$$\nabla f(x) - u_1 \nabla g_1(x) = 0 \text{ при } u_1 = \frac{1}{3}, u_2 = \frac{2}{3},$$

т. е. $u_i \geq 0$.

2. Проверить точку $\mathbf{x}^T = (0, 0)$.

Здесь $I\{3, 4\} \rightarrow u_1 = u_2 = 0$.

$$\nabla f(x) = (-6, -4)^T; \nabla g_3(x) = (1, 0)^T; \nabla g_4(x) = (0, 1)^T, \text{ то}$$

$$\nabla f(x) - u_3 \nabla g_3(x) - u_4 \nabla g_4(x) = 0$$

при $u_3 = -6, u_4 = -4$, т. е. условие неотрицательности нарушено.

В ряде случаев условия Куна–Таккера могут выполняться в нескольких точках. Чтобы определить, является ли найденная точка точкой локального минимума, следует воспользоваться необходимым условием второго порядка. Тогда определим возможную точку локального минимума (так как это необходимое условие). Чтобы окончательно подтвердить соответствие этой точке минимуму, следует проверить выполнение достаточных условий второго порядка. Разумеется, если удовлетворяются условия первого порядка (теорема 2), то это точка глобального минимума. Но это весьма жесткие условия и в ряде случаев они не выполняются, поэтому приходится довольствоваться проверкой достаточных условий второго порядка на точку локального минимума.

3.4. Практическая проверка условий оптимальности

Рассмотрим задачу

$$f(x) \rightarrow \min, \mathbf{g}(x) \geq 0, \mathbf{x} \geq 0.$$

Теорема

Если пара $(\mathbf{x}^*, \mathbf{v}^*)$ – седловая точка функции Лагранжа, то \mathbf{x}^* – оптимальное решение исходной задачи нелинейного программирования. Если целевая функция и функции ограничений дифференцируемы, то можно сформулировать локальные условия Куна–Таккера (через производные функции Лагранжа).

Учитывая, что $\mathbf{x} \geq 0$, имеем две ситуации:

$$- \text{если } x_i^* = 0, \text{ то } \left(\frac{\partial L}{\partial x_i} \right)^* \geq 0;$$

$$- \text{если } x_i^* > 0, \text{ то } \left(\frac{\partial L}{\partial x_i} \right)^* = 0.$$

Аналогично, для вектора $\mathbf{u} \geq 0$:

$$- \text{если } u_j^* = 0, \text{ то } \left(\frac{\partial L}{\partial u_j} \right)^* \leq 0;$$

$$- \text{если } u_j^* > 0, \text{ то } \left(\frac{\partial L}{\partial u_j} \right)^* = 0.$$

Заметим, что $\left(\frac{\partial L}{\partial u_j} \right)^* u_j^* \equiv 0$. Поскольку $L = f(x) - \sum u_j g_j(x)$, то

$$\left(\frac{\partial L}{\partial u_j} \right)^* = -g_j(x).$$

Из соотношений следует, что

1. Если $\mathbf{g}(x^*) \geq 0$, то $u_j^* = 0$ (неактивные ограничения).

2. Если $\mathbf{g}(x^*) = 0$, то $u_j^* > 0$ (активные).

3. Если $x_i^* > 0$, то $\left(\frac{\partial L}{\partial x_i} \right)^* = 0$, тогда дифференцирующую по x_i функ-

цию, имеем в точке \mathbf{x}^*

$$\left(\frac{\partial L}{\partial x_i}\right)^* = \sum_{l=1}^M u_l^* \left(\frac{\partial g_l}{\partial x_i}\right)^*.$$

4. Если $x_i^* = 0$, то $\left(\frac{\partial L}{\partial x_i}\right)^* \geq 0$, т. е.

$$\left(\frac{\partial f}{\partial x_i}\right)^* \geq \sum_{j=1}^m u_j^* \left(\frac{\partial g_j}{\partial x_i}\right)^*.$$

Для проверки оптимальности основным являются два последних соотношения. Первые два указывают, что можно учитывать только активные ограничения.

Алгоритм

1. Записать ограничения в стандартном виде (со знаком \geq , т. е. $\mathbf{g}(x) \geq 0$). Подставить в ограничения \mathbf{x}^* и проверить, как они выполняются. Для ограничений, выполненных как неравенства, положить $u_i = 0$ и во всех дальнейших операциях учитывать только активные ограничения.

2. Если активных ограничений меньше, чем переменных x , то следует произвольным образом разбить переменные на две группы (базисные и свободные). Число базисных переменных должно быть равно числу активных ограничений. Множители u_i , соответствующие активным ограничениям, могут быть найдены по формуле:

$$\hat{\mathbf{u}} = \frac{\partial f}{\partial x_{\delta}} \left(\frac{\partial \hat{\mathbf{g}}}{\partial x_{\delta}} \right)^{-1},$$

где $\hat{\mathbf{g}}$ – составлены только из активных ограничений; $\hat{\mathbf{u}}$ – множители, соответствующие этим ограничениям.

3. Проверить следующие соотношения:

$$\text{если } x_i^* > 0, \text{ то } \left(\frac{\partial L}{\partial x_i}\right)^* = 0$$

ИЛИ

$$\left(\frac{\partial f}{\partial x_i}\right)^* = \sum_{j=1}^M u_j^* \left(\frac{\partial g_j}{\partial x_i}\right)^*; \quad (3.10)$$

$$\text{если } x_i^* = 0, \text{ то } \left(\frac{\partial L}{\partial x_i}\right)^* \geq 0$$

или

$$\left(\frac{\partial f}{\partial x_i}\right)^* \geq \sum_{j=1}^m u_j^* \left(\frac{\partial g_j}{\partial x_i}\right)^*. \quad (3.11)$$

Пример

$$f(x) = x_1^2 - 10x_1 - 2x_1x_2 + 2x_2^2 \rightarrow \min$$

$$g_1(x) = 10 - x_1 - 2x_2 \geq 0, \quad x \geq 0;$$

$$g_2(x) = 6 - x_1 - x_2 \geq 0, \quad x^* = \begin{pmatrix} 23/5 \\ 7/5 \end{pmatrix}.$$

Итак, $g_1(x^*) > 0$; $g_2(x^*) = 0$,

т. е. $u_1^* = 0$, $u_2^* > 0$;

$$\hat{g}(x) = g_2(x) = 6 - x_1 - x_2;$$

$$\hat{u} = u_2^*.$$

Число активных ограничений меньше, чем число переменных.

Пусть $x_\delta = x_1$, тогда $\frac{\partial f}{\partial x_1} = 2x_1 - 10 - 2x_2$;

$$\frac{\partial \hat{g}}{\partial x_1} = -1.$$

$$u = (2x_1 - 10 - 2x_2)(-1) = 2x_2 + 10 - 2x_1 = \frac{14}{5} + 10 - \frac{46}{5} = \frac{18}{5}.$$

Соотношения (3.10) имеют вид

$$\frac{df(x)}{dx_1} = u_2^* \frac{dg_2}{dx_1};$$

$$\frac{df(x)}{dx_2} = u_2^* \frac{dg_2}{dx_2}.$$

Проверим, например, выполнение последнего соотношения:

$$\frac{df}{dx_2} = -2x_1 + 4x_2 = \frac{-46}{5} + \frac{28}{5} = -\frac{18}{5};$$

$$\frac{dg_2}{dx_2} = -1, \text{ т. е. соотношение выполнилось } \left(-\frac{18}{5} = -\frac{18}{5}\right).$$

Контрольные вопросы

Проверить условия оптимальности в задачах:

1. $f(x) = 3x_1^2 + x_1^2 - x_2 - 7x_3 \rightarrow \min;$

$$4x_1 + x_2 - 2x_3 \leq 5;$$

$$2x_2 + x_3 \leq 4;$$

$$x \geq 0; x^* = (2, 1, 2)^T.$$

2. $f(x) = x_1^2 + x_2^2 \rightarrow \min;$

$$2x_1 + x_2 \geq 0;$$

$$2x_1 - x_2 \leq 8;$$

$$x_1 + x_2 \leq 6; x^* = (0, 8, 0, 4)^T.$$

3.5. Функция Лагранжа и двойственность

Для любой задачи линейного программирования можно построить другую задачу нелинейной оптимизации, тесно связанную с исходной. Первая называется прямой, а вторая двойственной задачей. При некоторых предположениях о выпуклости, прямая и двойственная задачи имеют равные между собой оптимальные значения целевых функций. Это дает возможность получать решение исходной задачи, решая двойственную к ней (она часто проще).

Двойственная задача

$$\max \theta(u, v) \text{ при } u \geq 0,$$

$$\text{где } \theta(u, v) = \inf \left\{ f(x) - \sum_{i=1}^m u_i g_i(x) - \sum_{i=1}^l v_i h_i(x) : x \in X \right\}$$

(inf – infimum; $\inf \varphi(x)$ совпадает с $\min \varphi(x)$, если множество значений φ – конечно).

Функцию θ называют двойственной функцией Лагранжа (ограничения введены в целевую функцию с множителем Лагранжа u_i и v_i). Заметим, что множитель u_i , соответствующий ограничениям-неравенствам, неотрицателен, а v_i может иметь любой знак (v_i , u_i – двойственные переменные). Функция θ всегда вогнута. Так как задача заключается в максимизации нижней границы функции, то ее называют максиминной двойственной задачей.

Пример

$$(x_1^2 + x_2^2) \rightarrow \min_x \text{ при } \begin{cases} x_1 + x_2 - 4 \geq 0, \\ x_1, x_2 \geq 0. \end{cases}$$

Оптимальное значение целевой функции достигается в точке (2, 2) и равно 8. Двойственная функция

$$\begin{aligned} \theta(u) &= \inf \left\{ x_1^2 + x_2^2 - u(x_1 + x_2 - 4) : x_1 x_2 \geq 0 \right\} = \\ &= \inf \left\{ x_1^2 - u x_1 : x_1 \geq 0 \right\} + \inf \left\{ x_2^2 - 4x_2 : x_2 \geq 0 \right\} + 4u. \end{aligned}$$

Для определения \inf находим производные и решения уравнения, откуда обе нижние грани достигаются при

$$x_1 = x_2 = u/2, \text{ если } u \geq 0;$$

$$x_1 = x_2 = 0, \text{ если } u < 0.$$

Учитывая, что при $u \geq 0$ $\inf : \frac{u^2}{4} - \frac{u^2}{2} = -\frac{u^2}{2}$, получим

$$\begin{cases} -\frac{1}{2}u^2 + 4u, u \geq 0, \\ 4u, u < 0. \end{cases}$$

Построили функцию, которая является вогнутой функцией, достигающей максимума в точке $\bar{u} = 4$. Оптимальное значение $\theta^* = 8$ (как и в прямой задаче), так как θ – всегда вогнутая функция, то известно, что всякий ее локальный оптимум является глобальным. Таким образом, задача максимизации θ является привлекательной. Однако вначале должна быть решена задача минимизации функции Лагранжа по x (для нахождения значения функции θ в точке).

3.6. Задача, двойственная по Лагранжу

Прямая задача оптимизации в общем случае имеет вид

$$\min f(x) \text{ при } g_i(x) \geq 0, \quad i = \overline{1, M}, \quad x \in D;$$

$$h_i(x) = 0, \quad i = \overline{1, K}.$$

Задача, двойственная по Лагранжу, привлекательна тем, что приводит к различным алгоритмам решения, как линейных задач большой размерности, так и задач выпуклого и невыпуклого нелинейного программирования.

Геометрическая интерпретация условий Куна–Таккера

Из условий Куна–Таккера следует, что

$$\nabla f(x^*) = \sum_{j \in J} u_j \nabla g_j(x^*), \quad u_j \geq 0, \quad j \in J.$$

Известно, что любой вектор, представленный в виде

$$\sum_{j \in J} u_j \nabla g_j(x^*), \quad u_j \geq 0 \text{ при } j \in J,$$

принадлежит конусу, образованному градиентами функций, определяющих активные ограничения в точке x^* (рис. 16, а). В точке A удовлетворяется условие Куна–Таккера, а в точке B не удовлетворяется, так как $\nabla f(B)$ не принадлежит конусу. В точке A выполняются необходимые условия существования минимума функции при наличии ограничений.

При движении в любом допустимом направлении (внутри области ограничений) из точки A критерий оптимальности не может быть уменьшен.

Геометрическая интерпретация двойственной задачи по Лагранжу

Рассмотрим задачу $\min f(x)$ при $\mathbf{g}(x) \geq 0$, $\mathbf{x} \in D$. В плоскости (z_1, z_2) изображено множество $G = \{(z_1, z_2) : z_1 = \mathbf{g}(x), z_2 = f(x), \mathbf{x} \in D\}$, т. е. G – образ множества D при отображении \mathbf{g}, \mathbf{f} .

Прямая задача: найти точку из множества G , левее оси z_2 с минимальной ординатой (это точка A) (рис. 16, б), так как $z_1 \leq 0$.

Пусть задано $\mathbf{u} \leq 0$. Чтобы определить $\theta(\mathbf{u})$, надо определить $\min(f(x) + \mathbf{u}\mathbf{g}(x))$ при $\mathbf{x} \in D$, т. е. $\min(z_2 + \mathbf{u}z_1)$ на множестве G . Заметим, что $z_2 + \mathbf{u}z_1 = \alpha$ – уравнение прямой относительно оси z_2 . Чтобы минимизировать $z_2 + \mathbf{u}z_1$, на G надо перемещать прямую $z_2 + \mathbf{u}z_1 = \alpha$ параллельно самой себе, пока она не коснется множества G . Точка пересечения прямой с осью z_2 – искомое значение $\theta(\mathbf{u})$. Поэтому двойственная задача заключается в нахождении такого наклона гиперплоскости, при котором значение координаты z_2 точки ее пересечения с осью z_2 будет максимальным. Такая гиперплоскость имеет наклон и является касательной к G в точке A .

Таким образом, оптимальным решением двойственной задачи является $\bar{\mathbf{u}}$, а оптимальным значением целевой функции z_2 .

Если \mathbf{x} – допустимая точка D , то $f(\mathbf{x}) \leq \theta(\mathbf{u}, \mathbf{v})$, таким образом $\inf\{f(\mathbf{x}) : \mathbf{g}(\mathbf{x}) \leq 0, h(\mathbf{x}) = 0, \mathbf{x} \in D\} \geq \sup\{\theta(\mathbf{u}, \mathbf{v}) : \mathbf{u} \geq 0\}$.

Если $f(\mathbf{x}) \geq \theta(\mathbf{u}, \mathbf{v})$, то говорят о разрыве двойственности (в невыпуклом случае) (рис. 16, в) (на рисунке показаны : 1 – разрыв двойственности; 2 – оптимальное значение целевой функции прямой задачи; 3 – оптимальное значение целевой функции двойственной задачи). Разрыв требует дополнительных усилий для решения прямой задачи.

Для решения двойственной задачи разработаны различные алгоритмы (например, градиентный – движение по $\nabla\theta(\bar{\mathbf{u}}, \bar{\mathbf{v}})$ – направлению подъема для функции θ в точке $\bar{\mathbf{u}}, \bar{\mathbf{v}}$. При этом выполняется решение вспомогательной задачи

$$\min f(\mathbf{x}) + \mathbf{u}^{-T} \mathbf{g}(\mathbf{x}) + \mathbf{v}^{-T} h(\mathbf{x}), \text{ при } \mathbf{x} \in D,$$

где $\bar{\mathbf{u}}, \bar{\mathbf{v}}$ – заданные значения.

При некоторых предположениях о вогнутости множества D на каждой итерации алгоритма решения двойственной задачи можно получать допустимые точки прямой задачи с помощью решения следующей задачи линейного программирования:

$$\min \sum_{j=0}^k \lambda_j f(x_j) \text{ при } \sum_{j=0}^k \lambda_j \mathbf{g}(x_j) \leq 0; \sum_{j=0}^k \lambda_j h(x_j) = 0; \sum_{j=0}^k \lambda_j = 1,$$

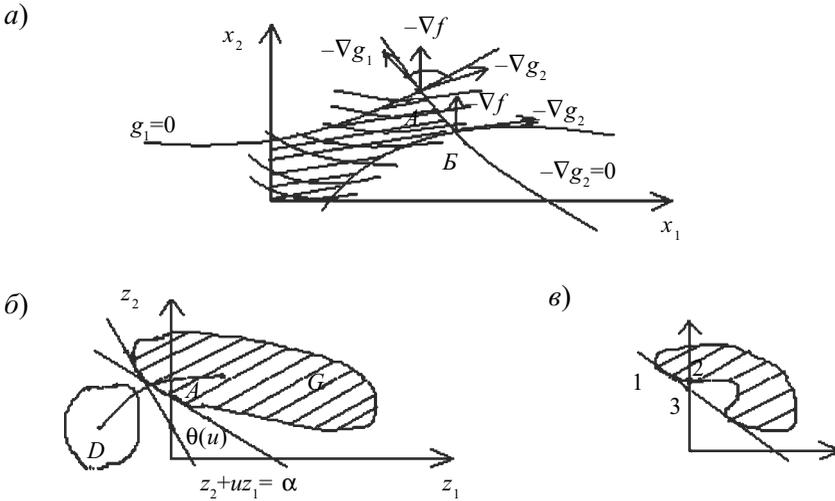


Рис. 16

$$\lambda_j \geq 0, j = \overline{0, k},$$

где $x_j \in D(u_j, v_j)$ при $j = \overline{1, k}$ – точки, полученные в процессе максимизации $\theta(u, v)$.

3.7. Методы оптимизации на основе преобразования задачи

Пусть рассматривается следующая задача нелинейного программирования:

$$f(x) \rightarrow \min, x \in R^n$$

$$\text{при } g_j(x) \geq 0, j = \overline{1, J};$$

$$h_k(x) = 0, k = \overline{1, K};$$

$$x_i^l \leq x_i \leq x_i^u, i = \overline{1, n}.$$

Предполагается, что для вектора x^* , являющегося решением этой задачи, известно некоторое начальное приближение $x^{(0)}$, возможно недопустимое (т. е. не удовлетворяющее ограничениям).

Методы рассматриваемого класса позволяют построить конечную последовательность точек $x^{(t)}$, $t=0, 1, \dots, T$, которая заканчивается точкой $x^{(T)}$, дающей наилучшее приближение к x^* среди всех точек построенной последовательности. В качестве $x^{(t)}$ берутся стационарные

точки штрафной функции – целевой функции вспомогательной задачи безусловной минимизации. С помощью этой функции исходная задача условной минимизации преобразуется в последовательность задач безусловной минимизации. Штрафная функция позволяет ограничиться решением только одной задачи безусловной минимизации, называемой точной.

Методы, использующие штрафные функции, определяются видом штрафной функции, а также правилами пересчета штрафных параметров.

Условия оптимальности, рассмотренные ранее, составляют основу методов и дают аппарат для их исследования (проверка полученных точек на оптимальность).

Вообще, идея преобразования задач без ограничений является заманчивой, поскольку позволяет использовать эффективные методы безусловной минимизации. Предполагается, что можно отыскать минимум с приемлемой точностью, решая несколько не очень сложных подзадач.

Штрафная функция определяется выражением

$$P(x, R) = f(x) + \Omega(R, g(x), h(x)),$$

где R – набор штрафных параметров; Ω – штраф (по сути R – весовой коэффициент, определяющий относительную значимость $f(x)$ и ограничений).

Если штраф создает барьер из больших значений P вдоль границы допустимой области, то эти методы называются методами барьеров. При этом последовательность точек приближается к оптимальному решению внутри допустимой области (относятся к методам внутренней точки). Возможны последовательности, состоящие из недопустимых точек (методы внешней точки).

Типы штрафов

1. Квадратичный штраф (для учета ограничений равенств)

$$\Omega = R \sum_{k=1}^k h_k^2(x).$$

При минимизации этот штраф препятствует отклонению величины $\mathbf{h}(x)$ от нуля (рис. 17, *a*). При возрастании R возрастает штраф за

нарушение ограничений, следовательно, \mathbf{x} стремится к \mathbf{x}^* (где $h(\mathbf{x}^{(T)}) = 0$). Это происходит потому, что с введением штрафа нарушение ограничений становится “невыгодным” (с точки зрения алгоритмов решения задач безусловной минимизации, так как они работают на уменьшение целевой функции).

Пример

$$f(x) = (x_1 - 4)^2 + (x_2 - 4)^2 \rightarrow \min_x;$$

$$h(x) = x_1 + x_2 - 5 = 0;$$

$$\mathbf{x}^* = (2,5; 2,5)^T;$$

$$P(x, R) = (x_1 - 4)^2 + (x_2 - 4)^2 + \frac{1}{2}(x_1 + x_2 - 5)^2.$$

Уравнения, определяющие стационарную точку функции $P(x, R)$

$$\frac{\partial P}{\partial x_1} = 2(x_1 - 4) + \frac{2}{R_1}(x_1 + x_2 - 5) = 0;$$

$$\frac{\partial P}{\partial x_2} = 2(x_2 - 4) + \frac{2}{R_1}(x_1 + x_2 - 5) = 0,$$

откуда $x_1 = x_2 = \frac{(10 + 8R_1)}{(4 + 2R_1)}$;

$$\lim_{R \rightarrow 0} \frac{10 + 8R_1}{4 + 2R_1} = \frac{10}{4} = 2,5, \mathbf{x}^* = (2,5, 2,5)^T.$$

Видно, что при достаточно малом R_1 (т. е. достаточно большом $R_1 = \frac{1}{R}$) решение вспомогательной задачи можно сделать сколь угодно близким к исходной. Выбрать сразу $R_1 = 0$ нельзя, но можно взять, скажем, $R_1 = 0,01$. Появление очень больших параметров ухудшает обусловленность матрицы Гессе, поэтому штраф изменяют поэтапно. Тогда стационарная точка P будет смещаться в сторону все более точного выполнения ограничений $h(x) = 0$ и тем самым все ближе подходить к \mathbf{x}^* .

Теперь перейдем к ограничениям-неравенствам.

1. Бесконечный барьер (рис. 17, б). Отметим, что использовать $Rg^2(x)$ в качестве штрафа здесь оказывается неудобно, так как штраф будет взиматься при любом знаке $g(x)$.

$$\Omega = 10^{20} \sum_{j \in I} |g_j(x)|, \quad \bar{I} = \{i | g(x) < 0\}$$

– множество индексов нарушенных ограничений,
или

$$\Omega = R \sum_{i=1}^m (\min(0, g_i(x)))^2 \quad (\text{если } g(x) \geq 0, \text{ то штраф не берется}).$$

2. Логарифмический штраф $\Omega = -R \sum_{i \in I} \ln [g_i(x)]$, $g_i(x) \geq 0$. Отметим,

что функции $g_i(x)$ не могут быть отрицательными (рис. 17, в). Внутренним точкам отдается предпочтение перед граничными. Это барьерная функция (неопределенная в недопустимых точках). После решения каждой подзадачи безусловной минимизации параметр R уменьшается по некоторому правилу $R^{(t+1)} = \varphi(R^{(t)})$ и в пределе стремится к нулю. Заметим, что этот штраф не определен в недопустимых точках (где $g(x) < 0$), поэтому, если исходная точка – недопустима, нужна специальная процедура, обеспечивающая попадание в допустимую область. Отметим, что даже если в процессе поиска будем двигаться к границе допустимой области (а минимум может находиться там), то $R \rightarrow 0$, и, следовательно, штраф $\Omega \rightarrow 0$ (в оптимальной точке штраф должен быть равен нулю).

3. Квадрат срезки (рис. 17, г).

$$\Omega = R \sum_{j=1}^I < g_j(x) >^2,$$

где функция срезки определена следующим образом:

$$< a > = \begin{cases} a, & a \leq 0; \\ 0, & a > 0. \end{cases}$$

Этот внешний штраф и стационарные точки функции P могут оказаться недопустимыми, но они не создают сложностей по сравнению с допустимыми, как например, в случае логарифмического штрафа,

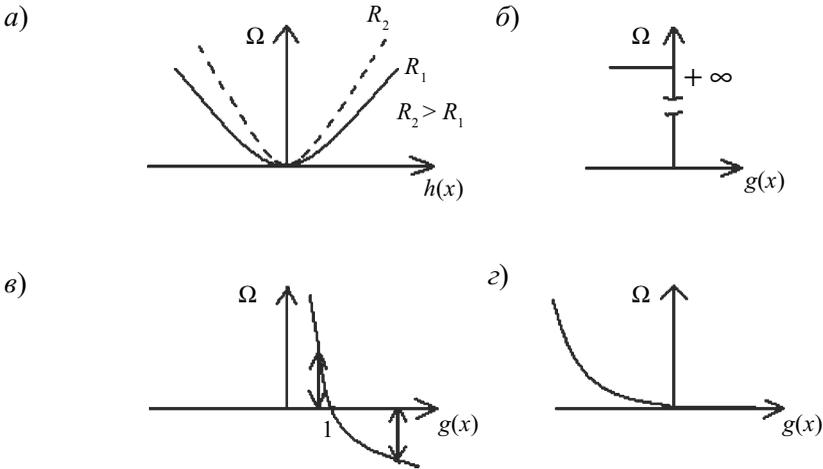


Рис. 17

так как в допустимых и граничных точках штраф равен нулю. Удобство в том, что функция P определена и непрерывна всюду (т. е. градиент не терпит разрывов). После решения очередной подзадачи R увеличивается (увеличение R препятствует нарушению ограничений).

Методы внешней точки

При любом $R \geq 0$ соответствующая стационарная точка является недопустимой (за исключением, конечно, R_{\max} , достигнутого в процессе поиска минимума), например для функции $P = (x_1 - 4)^2 + (x_2 - 4)^2 + R < 5 - x_1 - x_2 >^2$ при изменении R от 0 до 100 стационарная точка перемещалась от точки $[4, 4]^T$ (безусловный минимум) к точке $[2, 5; 2, 5]$ – условный минимум.

Сходимость метода штрафных функций связана со степенью вытянутости линий уровня штрафной функции, фактически, со степенью обусловленности задачи. Таким образом, при некоторых предельных значениях R можем просто не решить очередную подзадачу (возможно надо будет перейти к овражному методу минимизации).

В любом случае необходимо выбрать $R^{(0)}$ и его изменение. При этом R следует выбрать так, что при переходе от одной подзадачи к другой весовые коэффициенты в ограничениях, учитываемых внешним штрафом, увеличивались, а весовые коэффициенты в ограничениях, учитываемых внутренним штрафом (барьером), уменьшались. Наилуч-

шие результаты дает квадратичное изменение R . Причем, чтобы уменьшить эффект деформации линий уровня, точку, полученную при решении очередной подзадачи, выбирают в качестве начальной при решении последующей.

Обобщенный алгоритм минимизации с помощью штрафных функций.

1. Задать $n, J, K, \varepsilon_1, \varepsilon_2, \varepsilon_3, x^{(0)}, R^{(0)}$, где n – размерность вектора x ; J, K – число ограничений (равенств, неравенств); ε_1 – параметр линейного поиска; ε_2 – параметр окончания работы процедуры безусловной минимизации; ε_3 – параметр окончания работы алгоритма.

2. Построить $P(x, R) = f(x) + \Omega(R, g(x), h(x))$ и найти $x^{(t+1)}$, доставляющее минимум $P(x^{(t+1)}, R^{(t)})$ при фиксированном $R^{(t)}$. В качестве начальной точки использовать $x^{(t)}$, а в качестве параметра окончания шага ε_2 .

3. Если $|P(x^{(t+1)}, R^{(t)}) - P(x^{(t)}, R^{(t-1)})| \leq \varepsilon_3$, то $x^{(t+1)} = x^{(t)}$; идти на конец; иначе к п. 4.

4. $R^{(t+1)} = R^{(t)} + \Delta R^{(t)}$, перейти к п. 2, где $\Delta R^{(t)}$ – приращение соответствующего знака.

5. Конец.

Метод множителей

Метод позволяет найти решение без значительного ухудшения обусловленности задачи.

Преимущества

1. Сходится к точке Куна–Таккера (если удастся решить все подзадачи).

2. Задача $\min_x P(x, R)$ близка по сложности задаче $\min_x f(x)$.

Недостаток

$\mathbf{x}^{(t)} \rightarrow \mathbf{x}^*$ вне допустимой области.

Схема метода

$$P(x, \sigma^{(t)}, \tau^{(t)}) = f(x) + R \sum_{j=1}^J \left\{ \langle g_j(x) + \sigma_j^{(t)} \rangle^2 - \sigma_j^{(t)2} \right\} + \\ + R \sum_{k=1}^K \left\{ \left[h_k(x) + \tau_k^{(t)} \right]^2 - \tau_k^{(t)2} \right\},$$

где R константа, не зависящая от номера итерации t (но может зависеть от j), т. е. R не меняется в процессе итераций

$$\sigma^{(t+1)} = \langle g_j(x^{(t)}) + \sigma_j^{(t)}, j = \overline{1, J};$$

$$\tau_k^{(t+1)} = h_k(x_k^{(t)}) + \tau_k^{(t)}, k = \overline{1, K},$$

где $\sigma^{(0)} = \tau^{(0)} = 0$ (первая подзадача решается, как в методе штрафных функций).

Параметры σ, τ осуществляют сдвиг итерационных слагаемых, и этот сдвиг итеративно уточняется (за счет этого штрафная функция и изменяется). При сдвиге штраф за нарушение ограничений возрастает и $x^{(t)}$ приближается к допустимой области (движение из недопустимой точки). Поверхности уровня P в случае линейных функций $g(x)$ и $h(x)$, $(\nabla^2 P(x) \neq \varphi(\sigma, \tau))$, не меняют формы, а лишь сдвигаются относительно поверхностей $f(x)$. В нелинейном случае форма поверхностей незначительно меняется.

Учет ограничений типа $\alpha_i \leq x_i \leq b_i$

Сюда относят конструктивные, технологические ограничения или функции, неопределенные при некоторых значениях переменных, например при 0. Они могут учитываться как общие ограничения-неравенства (т. е. каждое добавляет еще два ограничения-неравенства). Однако лучше не увеличивать размерность задачи, а ввести процедуру, которая позволяет обнаруживать нарушения и фиксировать переменные на их границе. Можно все переменные, нарушающие ограничения, одновременно вернуть на свои границы. При этом хотя и меняется направление поиска, но этот способ считается наилучшим.

3.8. Методы прямого поиска в задачах условной оптимизации

Эти методы основываются на способах отыскания точек оптимума, когда используется только целевая функция и ограничения. Необходимость в использовании таких методов связана с тем, что в технических приложениях часто возникают задачи, в которых функции разрывны или недифференцируемы.

1. Подготовка задачи к решению.

Если очередная, найденная точка, не удовлетворяет ограничению-неравенству, то изменяя шаг, можно заменить ее другой точкой. В случае ограничений-равенств возможны трудности, так как даже если две точки x^0 и v удовлетворяют равенствам, то точки на прямой

$$x = x^0 + \alpha(v - x^0), 0 \leq \alpha \leq 1,$$

где x – точка на прямой; $(v - x^0)$ – направление, куда делается шаг, могут им не удовлетворять.

Проблема сводится к нахождению решения системы

$$h_k(x) = 0, \quad k = \overline{1, K}.$$

Надо подобрать точку так, чтобы она была лучше x^0 и удовлетворяла ограничению-равенству, используя значения функции h_k и подбирая коэффициент v . Такая процедура трудоемка, поэтому ограничения-равенства должны быть исключены перед решением.

Один из способов: решить $h_k(x)$ относительно одной из переменных и подставить это выражение в выражения, описывающие задачу с учетом границ переменных.

Пример

$$f(x) = x_1^2 + 4x_2^2 + x_3^2 \rightarrow \min \text{ при } h_1(x) = x_1 + x_2^2 - 2 = 0;$$

$$-1 \leq x_1 \leq 1; \quad 0 \leq x_2 \leq 2; \quad 0 \leq x_3 \leq 2;$$

$$x_1 = 2 - x_2^2, \quad f(x_2, x_3) = (2 - x_2^2)^2 + 4x_2^2 + x_3^2;$$

$$-1 \leq 2 - x_2^2 \leq 1.$$

Таким образом,

$$f(x) = (2 - x_2^2)^2 + 4x_2^2 + x_3^2;$$

$$1 \leq x_2 \leq \sqrt{3}, \quad 0 \leq x_3 \leq 2. \quad (3.12)$$

Однако часто переменную аналитически не удастся исключить ($\mathbf{h}(x)$ – нелинейно по \mathbf{x}) или это трудно сделать. Но всегда можно численно решить $\mathbf{h}(x)$ относительно зависимых переменных при заданных значениях независимых (оптимизируемых) переменных.

В предыдущем примере численно решаем $\mathbf{h}(x)$ относительно x при заданных x_2, x_3 .

2. Определение допустимой точки.

Алгоритмы прямого поиска начинают работать с допустимой точки. Если априори она не известна, то надо ее определить.

Одним из методов является метод случайного поиска пробной точки

$$x_i = x_i^{(l)} + \gamma_i(x_i^{(u)} - x_i^{(l)}),$$

$i = \overline{1, n}$, $x_i^{(l)}$, $x_i^{(u)}$ – нижняя и верхняя границы переменной; γ_i – случайные числа, равномерно-распределенные на интервале $(0, 1)$.

Затем эта точка проверяется на допустимость (процесс продолжается пока не найдем допустимую точку. Понятно, что для задач высокой размерности с узкой допустимой областью надо использовать другие методы (равенства должны быть исключены).

Модифицированный метод Хука–Дживса

Методы прямого поиска в задачах безусловной минимизации можно модифицировать, в частности метод Хука–Дживса, а именно: при решении задачи минимизации надо присвоить целевой функции большое значение там, где нарушаются ограничения. Поиск будет осуществляться снова в допустимой области в направлении к минимальной точке внутри этой области. Иллюстрация исследующего поиска приведена на рис. 18, а, поиск по образцу на рис 18, б. Заметим, что приходится уменьшать шаг в два раза, пока не найдем допустимую базовую точку. Имеется следующая трудность: с помощью этого метода нельзя двигаться вдоль границы области ограничений (так как все время как бы отскакиваем от границы). Сходимость достигается в первой же точке границы, что и предлагается в качестве решения. Следующий метод позволяет расширить множество направлений поиска.

Метод комплексов (метод Бокса)

Последовательно порождаются точки симплекса в пространстве поиска методом случайного поиска (3.12). Каждая точка проверяется на допустимость и если какие-либо ограничения нарушаются, то она сдвигается к центру тяжести уже построенных точек до тех пор, пока не получится допустимая точка. Заметим, что здесь берется центр тяжести не всех точек регулярного симплекса (как в безусловном варианте), а уже построенных. После того как множество точек построено ($p \geq n+1$), в каждой из них вычисляется значение целевой функции и точка с наибольшим значением функции отбрасывается. Новая точка получается отражением исключаемой точки через центр тяжести остальных точек, т. е.

$$x^m = \bar{x} + \alpha(\bar{x} - x^R),$$

где x^m – новая; \bar{x} – центр тяжести; x^R – отраженная точка. При значениях коэффициента $\alpha = 1$ – нормальное отражение; $\alpha > 1$ – растяжение; $\alpha < 1$ – сжатие.

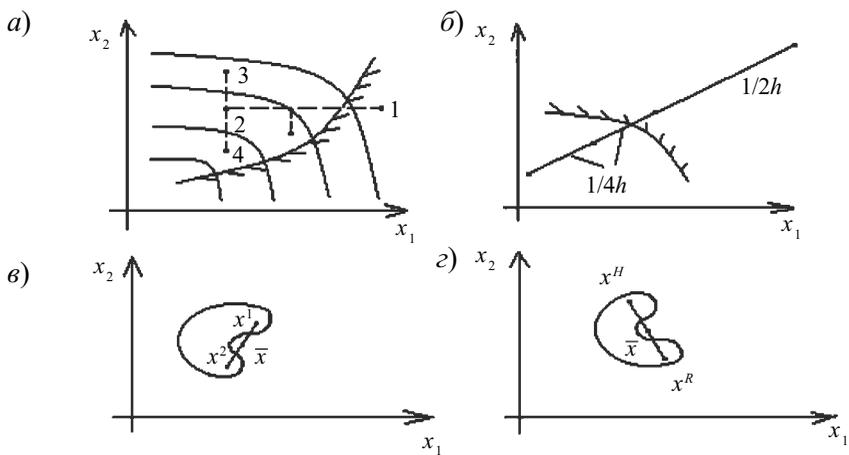


Рис. 18

Большее число вершин используется для предотвращения “уплотнения” комплекса при поиске вблизи границы. Возможны следующие случаи.

1. Новая точка допустимая и значение функции не совпадают с максимальным на всей совокупности точек. Тогда выбираем точку с максимальным значением функции и вновь делаем отражение.

2. Если точка допустимая и соответствует ранее найденному максимальному значению функции f , тогда, чтобы не зациклиться, передвигаем эту точку на половину расстояния до ранее найденного центра тяжести.

3. Найдена недопустимая точка. Тогда в два раза уменьшают расстояние до вычисленного центра тяжести (точку сдвигают).

Останов алгоритма производится в том случае, когда многогранник не будет стянут в центр тяжести и разница между значениями функции в вершинах не станет достаточно малой.

Если выходим за границы переменных, то соответствующая координата полагается равной граничному значению. Если допустимая область не выпуклая, то метод расходится (рис. 18, в, г).

Если пробные точки располагаются вдоль границы, то сходимость метода замедляется (до оптимума может быть далеко), поэтому вычисления несколько раз прерывают при выполнении какого-либо критерия останова. Наилучшие решения запоминаются и используются в качестве начальной точки для продолжения вычислений.

3.9. Методы случайного поиска

Эти методы целесообразно использовать либо в задачах небольшой размерности, либо как средство определения хорошей начальной точки с последующим улучшением получаемых оценок рассмотренными выше детерминированными методами. Дело в том, что случайный поиск позволяет охватить большую часть допустимой области, что дает возможность приблизиться к глобальному оптимуму.

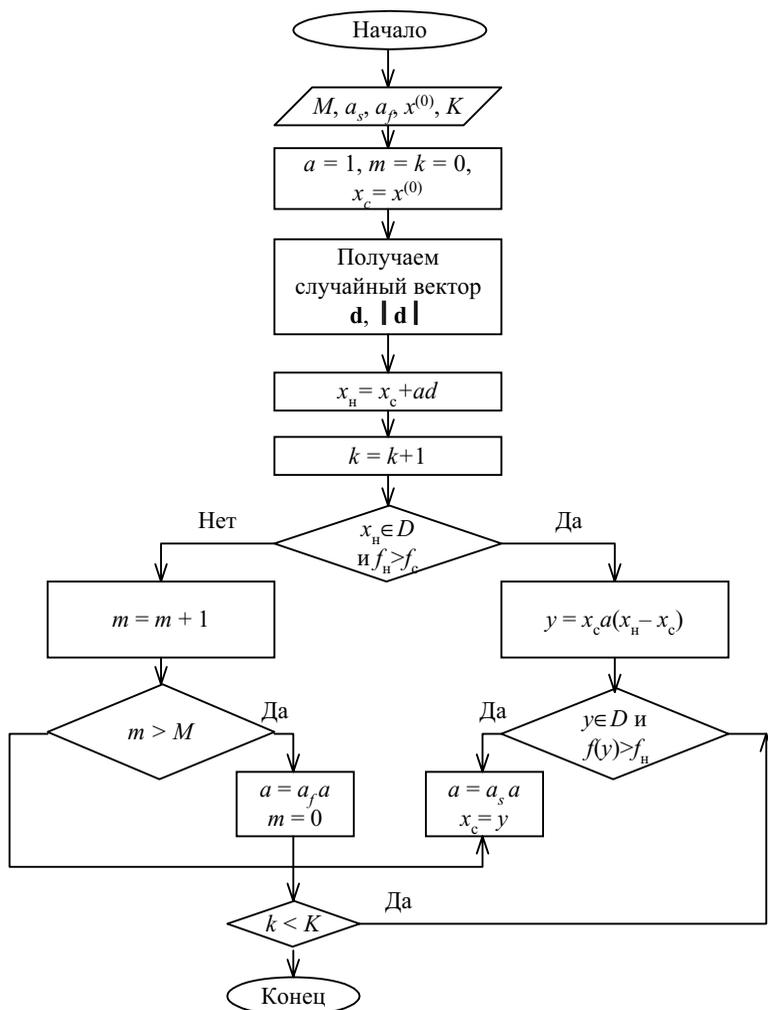


Рис. 19

Наиболее эффективны адаптивные эвристические процедуры, так как они минимизируют обращение к одновременным выборкам (не позволяют использовать накопленную информацию). Один из подходов использует следующую идею: случайные выборки использовать для определения направления поиска, а длина шага определяется в соответствии с достигаемым улучшением функции. Если две последовательные итерации дают улучшение, то шаг увеличивается в $\alpha_s = 1,618$ раз. Если же M последовательных итераций не дают улучшения, то шаг уменьшается в α_s раз. Блок-схема алгоритма приведена на рис. 19.

Вектор \mathbf{d} единичной длины формулируется по правилу $\mathbf{d}_i = \frac{x_i}{\|\mathbf{x}\|}$, где $x_i = x_i^{(l)} + r_i (x_i^{(u)} - x_i^{(l)})$; $x_i^{(l)}$, $x_i^{(u)}$ – прямые ограничения на x_i ; r_i – случайное число, равномерно распределенное на интервале от 0 до 1.

4. ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ (ЛП)

К задачам ЛП относятся задачи, в которых ограничения и целевая функция – линейны, например

$$f(x) = \left\{ \sum_{i=1}^n c_i x_i \right\} \rightarrow \min_x; \quad (4.1)$$

$$\text{при } \sum_{i=1}^n a_{ki} x_i \leq b_k; \quad k = \overline{1, m}, \quad i = \overline{1, n},$$

где \mathbf{c} – вектор оценок; \mathbf{x} – вектор переменных; \mathbf{b} – вектор ресурсов; a_{ki} – элементы матрицы коэффициентов; $x_i \geq 0$. Или в векторном виде

$$f(x) = \mathbf{c}\mathbf{x} \rightarrow \min, \quad \mathbf{A}\mathbf{x} \leq \mathbf{b}, \quad \mathbf{x} \geq 0.$$

Каноническая форма задачи ЛП имеет вид

$$f(x) = \mathbf{c}\mathbf{x} \rightarrow \min, \quad \mathbf{A}\mathbf{x} = \mathbf{b}, \quad \mathbf{x} \geq 0. \quad (4.2)$$

Методы ЛП широко используются по причине доступности ПО для решения задач ЛП высокой размерности и возможности анализа решений при вариации исходных данных.

Пример (графическое решение)

Фирма производит два типа шкафов: A и B . Для каждого изделия типа A требуется 3 м² досок, B – 4 м². Фирма может получить от поставщиков до 1700 м² досок в неделю. Для каждого изделия типа A надо 12 мин работы на станке, а для изделия B – 30 мин. В неделю можно использовать 160 ч машинного времени. Сколько изделий каждого типа надо выпускать в неделю, если каждое изделие типа A приносит 2 у. е. прибыли, а изделие B – 4 у. е. Таким образом, имеем целевую функцию

$$P = 2x_1 + 4x_2 \rightarrow \max, \quad x_1 \geq 0, \quad x_2 \geq 0,$$

где x_1 – число изделий A ; x_2 – число изделий B .

Ограничения, соответственно,

$$3x_1 + 4x_2 \leq 1700;$$

$$2x_1 + 5x_2 \leq 1600.$$

В качестве иллюстрации рассмотрим рис. 20, а.

Заметим, что $\frac{\partial P}{\partial x_1} = 2 > 0$, $\frac{\partial P}{\partial x_2} = 4 > 0$, т. е. никогда не равны 0, следова-

тельно, максимум функции располагается на границе, т. е. $P^* = 2 \cdot 300 + 4 \cdot 200 = 1400$. Это точка, где линия уровня функции P последний раз пересекает границу ограничения.

Во многих задачах, в частности в задачах ЛП, важно оценить влияние изменения параметров на полученное решение. Если окажется, что оптимальное решение можно значительно улучшить за счет небольших изменений параметров, то целесообразно сделать эти изменения. Из

анализа оценок величин $\frac{\partial P}{\partial b_i}$ можно отметить, что если при увеличе-

нии рабочего времени, т. е. b_2 (за счет сверхурочных работ) прирост дохода ΔP превышает дополнительные затраты на оплату труда, то сверхурочные работы экономически оправданы.

Если удастся определить, какие параметры наиболее важны для целевой функции, то они и требуют наиболее точного задания (при этом повышается надежность модели). С другой стороны, если какие-либо параметры модели слабо влияют на изменение целевой функции при изменении их до 0, то их можно опустить, тем самым упростив модель.

Возвращаясь к свойствам задачи ЛП, заметим следующее. Если в задаче ЛП существует оптимальное решение (а их может быть несколько), то по крайней мере одна из вершин допустимой области представляет собой оптимальное решение, т. е. оптимальное решение всегда можно найти перебором всех вершин допустимой области.

Пример

$$z = x_1 + 2x_2 \rightarrow \max,$$

$$\text{при } x_1 + 2x_2 \leq 10; x_1 + x_2 \geq 1; x_2 \leq 4;$$

$$x_1 \geq 0; x_2 \geq 0.$$

Вычисляя значение функции z в вершинах $A(1, 0); B(10, 0); C(2, 4); D(0, 4); E(0, 1)$, можно убедиться, что оптимальные значения достигаются в вершинах B и C $z(B) = z(C) = 10$.

4.1. Приведение задачи ЛП к каноническому виду

К стандартному каноническому виду (4.2) задачу приводят, чтобы воспользоваться симплекс-методом.

1. Преобразование неравенств

$$x_1 + 2x_2 + 3x_3 + 4x_4 \leq 25 \rightarrow x_1 + 2x_2 + 3x_3 + 4x_4 + s_1 = 25,$$

где $s_i > 0$ – остаточная переменная (соответствует разности правой и левой части неравенства). По значениям переменных s в оптимальном решении можно судить, являются ли ограничения активными (когда $s_i = 0$).

2. Иногда возникает необходимость рассмотрения переменных, которые могут принимать положительные и отрицательные значения. Чтобы привести к стандартному виду, используют замены

$$s_i = s_1^+ - s_1^-, s_1^+, s_1^- \geq 0.$$

4.2. Табличный симплекс-метод

Интуитивный путь решения – определение методом Жордана–Гаусса всех возможных допустимых базисных решений системы уравнений-ограничений с последующим выбором решения с наилучшим значением целевой функции. Однако число уравнений-ограничений, как правило, меньше числа переменных (т. е., $m < n$), поэтому множество решений бесконечно и, следовательно, выбор наилучшего решения – нетривиален.

Напомним, что идея метода Жордана–Гаусса состоит в сведении системы m уравнений с n неизвестными к каноническому виду при помощи операций над строками (умножение на число, прибавление к уравнению другого уравнения)

$$x_i + \sum_{s=m+1}^n a_{i,s} x_s = \bar{b}_i, \quad i = \overline{1, m}, \quad (4.3)$$

где $a_{i,s}, \bar{b}_i$ – коэффициенты после операции приведения к каноническому виду (здесь вектор x разбит на две группы переменных); x_i – базис-

ные (зависимые) переменные входят с единичным коэффициентом (их столько – сколько уравнений-ограничений); x_s – свободные (независимые) переменные ($n-m$ штук).

Переменным x_s присваивают произвольные значения (последовательно присваивают единичные векторы) и решают относительно зависимых. Пусть $\mathbf{b} = 0$, тогда

$$x_1 - 0,25x_2 + 0,75x_4 = 0;$$

$$0,35x_2 + x_3 - 0,85x_4 = 0.$$

Общее решение : пусть x_2, x_4 – свободные

$$1) \quad x_2^{(1)} = 0, x_4^{(1)} = 1 \rightarrow x^{(1)} = \begin{pmatrix} -0,75 \\ 0 \\ 0,85 \\ 1 \end{pmatrix}.$$

$$2) \quad x_2^{(2)} = 1, x_4^{(2)} = 0 \rightarrow x^{(2)} = \begin{pmatrix} 0,25 \\ 1 \\ -0,35 \\ 0 \end{pmatrix}.$$

Это базис в множестве всех решений.

Предложенный Данцигом симплекс-метод оказался эффективнее, так как в нем анализируется только часть допустимых базисных решений (направленный перебор угловых точек).

Укрупненно алгоритм симплекс-метода состоит из следующих шагов.

1. Выбор начального допустимого базисного решения.

Это решение уравнений-ограничений, полученное при нулевых x_s , т. е. $\mathbf{x}_i = \bar{\mathbf{b}}_i$, $\bar{\mathbf{b}}_i \geq 0$, $\mathbf{x}_s = 0$, $s = m + 1, n$ (базисные переменные – неотрицательны).

2. Переход от начального допустимого решения к другому допустимому базисному решению с лучшим значением целевой функции. Исключение из рассмотрения всех допустимых базисных решений, которые хуже текущего решения.

3. Продолжение поиска лучших допустимых базисных решений. Если решение нельзя улучшить, оно является оптимальным и производится остановка алгоритма.

Применительно к задаче минимизации $f(x)$ подробнее рассмотрим шаги 2 и 3. Если требуется искать $\max f$, то можно решать задачу $\min(-f)$. Учитывая, что свободные переменные равны 0, значение целевой функции f соответствует начальному допустимому базисному решению $f = c_1\bar{b}_1 + \dots + c_m\bar{b}_m$.

Чтобы проверить существование допустимого базисного решения с меньшим значением f , вначале проверим решение на оптимальность. Если оно не оптимально, переходим к смежному допустимому базисному решению с меньшим значением f . Смежное решение отличается от рассматриваемого только одной переменной. Метод превращает одну свободную переменную в базисную, а одну базисную в свободную. Необходимо выбрать базисную и свободную так, чтобы замена одной из них на другую давала максимальное приращение целевой функции. Для выбора вводимой в базис переменной следует присвоить свободной переменной значение, равное 1 (она как бы возрастает от 0 до 1), и вычислить изменение f .

Итак, пусть рассматривается свободная переменная x_s , тогда из (4.3) имеем $x_i = \bar{b}_i - a_{is}$, $i = \overline{1, m}$

$$x_s = 1, x_j = 0, j = \overline{m+1, n}, j \neq s$$

и, следовательно,

$$f = \sum_{i=1}^m c_i(\bar{b}_i - \bar{a}_{is}) + c_s \cdot 1,$$

где \mathbf{c} – вектор оценок, т. е., если $c_s < 0$, то f уменьшилась.

Обозначим через c_s^+ отрицательное приращение f , получающееся при увеличении x_s до 1

$$c_s^+ = f_{\text{нов}} - f_{\text{стар}} = \sum_{i=1}^m c_i(\bar{b}_i - \bar{a}_{is}) + c_s - \left(\sum_{i=1}^m c_i\bar{b}_i \right) = c_s - \sum_{i=1}^m c_i\bar{a}_{is},$$

где c_s^+ – относительная оценка свободной переменной x_s , c_s – исходная оценка x_s .

Исследование базисного решения (опорного плана) на оптимальность, а также дальнейший вычислительный процесс удобнее вести, если условия задачи и первоначальные данные записать в виде табл. 4.1.

Таблица 4.1

Базис	x_1	x_2	...	x_r	...	x_m	x_{m+1}	...	x_s	...	x_n	Значение
x_1	1	0	...	0	...	0	a_{1m+1}	...	a_{1s}	...	a_{1n}	b_1
x_2	0	1	...	0	...	0	a_{2m+1}	...	a_{2s}	...	a_{2n}	b_2
⋮
x_r	0	0	...	1	...	0	a_{rm+1}	...	a_{rs}	...	a_{rn}	b_r
⋮
x_m	0	0	...	0	...	1	c_{m+1}	...	c_{ms}	...	c_{mn}	b_m
$-f$	0	0	...	0	...	0	c_{m+1}	...	c_s	...	c_n	$-f_0$

В табл. 4.1 f_0 – значение целевой функции со знаком минус. Исходный план определяется системой единичных векторов x_1, \dots, x_n .

1. Найти переменные для включения в базис.

Переменные x_{m+1}, \dots, x_n – свободные. Выберем наименьший из коэффициентов c_{m+1}, \dots, c_n . Пусть это c_s . Если он отрицательный, то увеличение x_s приводит к убыванию функции f . Переменную x_s надо ввести в базис. Если все $c \geq 0$, то значение функции f не может быть уменьшено и найдено оптимальное решение.

2. Найти переменную для исключения из базиса.

Вначале свободные переменные равны 0. Насколько можно увеличить x_s , не нарушая условия неотрицательности текущих базисных переменных?

Учитывая (4.3), x_s можно увеличивать до значения

$$\max x_s = \min(b_i/a_{is}), a_{is} > 0, a_{is} \neq 0.$$

$i=1, m$

Если бы $a_{is} \leq 0$, то при увеличении x_s базисная переменная x_i будет возрастать, а надо, чтобы убывала до 0. Если этот минимум достигается в строке r , то x_r обращается в 0, когда x_s принимает значение b_r/a_{rs} . Элемент a_{rs} называется ведущим, s – ведущим столбцом.

3. Построить новую каноническую форму.

Новый базис: $x_1, \dots, x_r, \dots, x_m$. Коэффициент при x_s в ведущей строке сделаем 1, поделив строку на a_{rs} , чтобы образовать новую ведущую строку. Удалим x_s из других ограничений. Для этого из i -й строки ($i \neq r$) вычтем a_{is} (новая ведущая строка). Чтобы преобразовать целевую

функцию с коэффициентом ($c_s < 0$) при x_s , вычтем c_s (новая ведущая строка) из строки, соответствующей целевой функции. Новая симплекс-таблица (табл. 4.2) (следующая итерация).

Таблица 4.2

Базис	x_1	x_2	...	x_r	...	x_m	x_{m+1}	...	x_s	...	x_n	Значение
x_1	1	0		A_{1r}^+		0	a_{1m+1}^+		0		a_{1n}^+	b_1^+
x_2	0	1		A_{2r}^+		0	a_{2m+1}^+		0		a_{2n}^+	b_2^+
x_r	0	0		A_{rr}^+		a_{rm+1}^+	a_{rm+1}^+		1		a_{rn}^+	b_r^+
x_m	0	0		A_{mr}^+		a_{mm+1}^+	a_{mm+1}^+		0		c_{mn}^+	b_m^+
$-f$	0	0		Cr^+		0	c_{m+1}^+		0		c_n^+	$-f_0^+$

В табл. 4.2:

$$b_r^+ = \frac{b_r}{a_{rs}}, a_{rj}^+ = \frac{a_{rj}}{a_{rs}}, i = r, j = \overline{1, n};$$

$$b_i^+ = b_i - a_{is} b_r^+, a_{ij}^+ - a_{is} a_{rj}^+, i \neq r;$$

$$c_j^+ = c_j - c_s a_{rj}^+, f_0^+ = f_0 + c_s b_r^+. \quad (4.4)$$

Идти на шаг 1 и выбрать минимальное из чисел c_j^+ , пока не получим, что все c_j^+ положительны.

Замечание

Если задача имеет вырожденные опорные планы, то на одной из итераций она имеет несколько переменных плана, которые могут оказаться равными нулю. Таким образом, при переходе от одного опорного плана к другому значение функции может оказаться прежним. Возможен случай, когда функция сохраняет свое значение в течение нескольких итераций, а также возможен возврат к первоначальному базису. В последнем случае говорят, что произошло заикливание.

Пример

$$f(x) = 8x_1 + 12x_2 \rightarrow \max;$$

$$2x_1 + 4x_2 \leq 440;$$

$$\frac{1}{2}x_1 + \frac{1}{4}x_2 \leq 65;$$

$$2x_1 + \frac{5}{2}x_2 \leq 320.$$

Приведем к каноническому виду

$$y = -f(x) = -8x_1 - 12x_2 \rightarrow \min;$$

$$2x_1 + 4x_2 + x_3 = 440;$$

$$\frac{1}{2}x_1 + \frac{1}{4}x_2 + x_3 = 65;$$

$$2x_1 + \frac{5}{2}x_2 + x_3 = 320.$$

Составим таблицу вида $\frac{\mathbf{A}|\mathbf{b}}{\mathbf{c}|\mathbf{0}}$ (табл. 4.3).

Таблица 4.3

Базис	Свободные			Базисные		
	x_1	x_2	x_3	x_4	x_5	B
x_3	2	4	1	0	0	440
x_4	?	1/4	0	1	0	65
x_5	2	5/2	0	0	1	320
-	-8	-12	0	0	0	0

Надо найти начальную точку. Для этого возьмем за базисные переменные (с единичным коэффициентом) x_3, x_4, x_5 , тогда $x_1 = x_2 = 0$ – свободные и $x_3 = 440, x_4 = 65, x_5 = 320$.

Для перехода к лучшей угловой точке надо

- разрешить систему уравнений относительно базисных переменных;
- исключить базисные переменные из целевой функции.

Рассмотрим эту процедуру

$$\left. \begin{aligned} x_3 &= 440 - 2x_1 - 4x_2; \\ x_4 &= 65 - \frac{1}{2}x_1 - \frac{1}{4}x_2; \\ x_5 &= 320 - 8x_1 - \frac{5}{2}x_2; \end{aligned} \right\}$$

$$y(x) = -8x - 12x \rightarrow \min;$$

$$x_3, x_4, x_5 - \text{базисные (зависимые)} \quad (4.5)$$

Поскольку коэффициент при x_2 наименьший, то при увеличении x_2 он в большей степени уменьшает целевую функцию, чем коэффициент при x_1 . Когда одна из x_3, x_4, x_5 станет равна нулю (т. е. станет свободной), зафиксируем x_2 (станет базисной). Таким образом, получается новая угловая точка.

Какую переменную сделать свободной? Заметим, что x_3 первой обращается в нуль, так как $x_3 = 0$ при $x_2 = 110$, $x_4 = 0$ при $x_2 = 160$, $x_5 = 0$ при $x_2 = 128$ (см. (4.5)). Не можем увеличивать x_2 более чем до 110, не нарушая условие $x_3 \geq 0$.

Итак, $\mathbf{x}_8 = (x_2, x_4, x_5)$, $\mathbf{x}_c = (x_3, x_1)$.

Разрешим (4.5) относительно базисных

$$x_2 = 110 - \frac{1}{2}x_1 - \frac{1}{4}x_3;$$

$$x_4 = 65 - \frac{1}{2}x_1 - \frac{1}{4}(110 - \frac{1}{2}x_1 - \frac{1}{4}x_3);$$

$$x_5 = 320 - 2x_1 - \frac{5}{2}(110 - \frac{1}{2}x_1 - \frac{1}{4}x_3).$$

Исключим базисные из целевой функции (табл. 4.4)

$$y(x) = -8x_1 - 12(110 - 0,5x_1 - 0,25x_3) = -1320 - 2x_1 + 3x_3;$$

$$2x_1 + 3x_3 = 0 \quad (\text{так как } x_1, x_3 - \text{свободные}).$$

Таблица 4.4

Базис	x_1	x_2	x_3	x_4	x_5	b
x_2	1/2	1	1/4	0	0	110
x_4	3/8	0	-1/16	1	0	37,5
x_5	3/4	0	-5/8	0	1	45
-	-2	0	3	0	0	1320

Теперь для уменьшения целевой функции лучше увеличивать x_1 .

Первой обнуляется x_5 (при возрастании x_1). Переменная x_1 переходит в вектор \mathbf{x}_8 , x_5 в \mathbf{x}_c .

Таким образом, $\mathbf{x}_8 = (x_1, x_2, x_4)$; $\mathbf{x}_c = (x_3, x_5)$;

$$r = 1; s = 2;$$

$$b_1^+ = 110; a_{11}^+ = 2/4 = 1/2;$$

$$b_2^+ = 65 - 110/4 = 37,5;$$

$$a_{12}^+ = 1; a_{13}^+ = 1/4.$$

Таблица 4.5

Базис	x_1	x_2	x_3	x_4	x_5	x_6
x_2	0	1	2/3	0	-2/3	80
x_4	0	0	1/4	1	-1/2	15
x_5	1	0	-2/3	0	4/3	60
0	0	0	4/3	0	8/3	1440

Следовательно, это решение оптимально и улучшить его нельзя (табл. 4.5). Таким образом, $\mathbf{x}^* = (60, 80, 0, 15, 0)$. Оптимальное решение – это столбец переменных b_i . Заметим, что всегда в строке оценок C для базисных переменных стоят нули, а в симплекс-таблице на каждой итерации базисным переменным соответствуют единицы.

Проверим выполнение ограничений. Ограничения, в которые вошли свободные переменные ($x_3 = x_5 = 0$), выполнены

$$2x_1 + 4x_2 = 440;$$

$$2x_1 + \frac{5}{2}x_2 = 320.$$

Упражнение

Построение симплекс-таблицы осуществить с помощью формул (4.4).

4.3. Двойственные задачи в ЛП

Для рассмотрения сути двойственных задач в ЛП рассмотрим следующий пример. При производстве трех видов изделий A, B, C используется три различных вида сырья в количестве, соответственно, не большем 180, 210, 244 кг. Нормы затрат каждого из видов сырья на единицу продукции и цена единицы продукции приведены в табл. 4.6.

Таблица 4.6

Вид сырья	A	B	C
1	4	2	1
2	3	1	3
3	1	2	5
Цена за единицу продукции	10	14	12

Требуется найти план выпуска продукции, при котором обеспечивается ее максимальная стоимость и оценить каждый

из видов сырья, используемых для производства продукции. Оценки по видам сырья должны обеспечивать минимальность оценки всего используемого сырья, а суммарная оценка сырья, используемого на производство единицы продукции каждого вида, должна быть не меньше цены единицы продукции данного вида.

Пусть производится x_1 изделий A , x_2 изделий B и x_3 изделий C . Для определения оптимального плана производства необходимо решить задачу

$$\begin{aligned} f(x) &= 10x_1 + 14x_2 + 12x_3 \rightarrow \max; \\ 4x_1 + 2x_2 + x_3 &\leq 180; \\ 3x_1 + x_2 + 3x_3 &\leq 210; \\ x_1 + 2x_2 + 5x_3 &\leq 244; \\ x_1, x_2, x_3 &\geq 0. \end{aligned}$$

Припишем каждому из видов сырья, используемых для производства, двойственную оценку соответственно u_1, u_2, u_3 . Тогда для определения оптимальной системы оценок сырья в соответствии с условием требуется решать задачу

$$\begin{aligned} \varphi(u) &= 180u_1 + 210u_2 + 244u_3 \rightarrow \min; \\ 4u_1 + 3u_2 + u_3 &\geq 10; \\ 2u_1 + u_2 + 2u_3 &\geq 14; \\ u_1 + 3u_2 + 5u_3 &\geq 12; \\ u_1, u_2, u_3 &\geq 0. \end{aligned}$$

Задачи образуют симметричную пару взаимно двойственных задач. Заметим, что для любой производственной программы и при любом векторе оценок выполняется неравенство $f(x) \leq \varphi(u)$, т. е. ценность всей выпущенной продукции не превосходит суммарной оценки имеющихся ресурсов (иначе часть ценности продукции возникает из «ничего»). Величина $\varphi(u) - f(x)$ характеризует производственные потери. Для их уменьшения необходимо максимизировать $f(x)$ и минимизировать $\varphi(u)$. При оптимальных производственной программе и векторе оценок ресурсов производственные потери равны 0. Далее, если оценка единицы ресурса i -го вида положительна, при оптимальной производственной программе этот ресурс используется полностью, если же ресурс используется не полностью, то его оценка равна 0. Поэтому двойственные оцен-

ки определяют дефицитность используемых предприятием ресурсов. Более того, величина u_i показывает, на сколько возрастает максимальное значение целевой функции прямой задачи при увеличении количества ресурса i -го вида на 1 ед. Если j -е ограничение двойственной задачи выполняется как строгое неравенство, это означает, что двойственная оценка ресурса, используемого на производство одного изделия j -го вида, выше цены этого изделия и, следовательно, выпускать изделие этого вида невыгодно.

Анализ чувствительности в ЛП

Оценка влияния изменений ресурсов b_i на минимальное значение f

$$\frac{\Delta f}{\Delta b_k} = \frac{\partial \varphi}{\partial b_k} \rightarrow \Delta f^* = \frac{\partial \varphi^*}{\partial b_k} \Delta b_k = \frac{\partial}{\partial b_k} \left(\sum_{i=1}^m b_i u_i^* \right) \Delta b_k = u_k^* \Delta b_k,$$

т. е. решение двойственной задачи u_k^* позволяет выделить ограничения, которые являются наиболее или наименее существенными для *оптимума в прямой задаче*. Запасы какого ресурса в первую очередь надо увеличить (какой является наиболее дефицитным).

Взаимосвязь двойственных задач в ЛП

Рассмотрим более подробно связь двойственных задач в общем случае, а также некоторые их особенности.

Запишем задачи, учитывая размерности векторов

$$f(x) = \mathbf{c}[N] \times \mathbf{x}[N] \rightarrow \min_{x \in \Omega}$$

или $\langle c, x \rangle \rightarrow \min$

$$\mathbf{A}[M_1, N] \times \mathbf{x}[N] \geq \mathbf{b}[M_1];$$

$$\mathbf{A}[M_2, N] \times \mathbf{x}[N] = \mathbf{b}[M_2];$$

$$\mathbf{x}[N_1] \geq \mathbf{o}[N_1]. \tag{4.6}$$

Двойственная $\varphi(u) = \mathbf{b}[M] \times \mathbf{u}[M] \rightarrow \max_{u \in \Lambda}$ или $\langle \mathbf{b}, \mathbf{u} \rangle \rightarrow \max$

$$\mathbf{u}[M] \times \mathbf{A}[M, N_1] \leq \mathbf{c}[N_1];$$

$$\mathbf{u}[M] \times \mathbf{A}[M, N_2] = \mathbf{c}[N_2];$$

$$\mathbf{u}[M_1] \geq \mathbf{o}[M_1]. \tag{4.7}$$

Для двойственных задач в оптимальной точке выполняется равенство $\min_{x \in \Omega} f(x) = \max_{u \in \Lambda} \varphi(u)$.

Отметим, что если задача на максимум, в ограничениях должен быть знак « \leq », если на минимум – « \geq » (т. е. ограничения необходимо привести к такому виду).

Пример 1

$$x_1 + 2x_2 + 3x_3 \rightarrow \min ;$$

$$u_1 + 2u_2 \rightarrow \max .$$

$$\left. \begin{array}{l} u_1 \left| \begin{array}{l} x_1 + x_2 - x_3 \geq 1; \\ x_1 - x_2 = 2; \\ x_1 \geq 0. \end{array} \right. \\ u_2 \end{array} \right\} \quad (4.8)$$

$$\left. \begin{array}{l} x_1 \left| \begin{array}{l} u_1 + u_2 \leq 1; \\ u_1 - u_2 = 2; \\ -4u_1 = 3 \\ u_1 \geq 0 \end{array} \right. \end{array} \right\} \text{противоречиво, (4.9)}$$

следовательно, $\Omega \neq \emptyset$, так как $(3, 1, 0) \in \Omega$.

Здесь матрица A имеет вид

$$A = \begin{bmatrix} 1 & 1 & -4 \\ 1 & -1 & 0 \end{bmatrix},$$

$$M_1 = 1; N_2 = 1;$$

$$M_2 = 1; N_2 = 1;$$

$$N = 3; M = 2.$$

Известно, что прямая и двойственная задачи либо обе разрешимы, либо обе не разрешимы. Поскольку $\Lambda = \emptyset$, следовательно, (4.9) не имеет решения, целевая функция $f(x)$ не ограничена снизу на Ω , то (4.8) не имеет решения.

Пример 2

Рассмотрим задачу

$$f(x) = x_1 + x_2 + x_3 \rightarrow \min;$$

$$\varphi(u) = u_1 + u_2 + u_3 \rightarrow \max;$$

формально

$$\begin{array}{l} x_3 = 1; \\ x_2 = 1; \\ x_1 + x_2 + x_3 = 3; \end{array} \quad (u_1 \quad u_2 \quad u_3) \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} = (u_3 \quad u_2 + u_3 \quad u_1 + u_3),$$

т. е. $u_3 \leq 1$; $u_2 + u_3 \leq 1$; $u_1 + u_3 \leq 1$.

Это ограничения равенства, и должны быть такие знаки

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}; \mathbf{u} = [u_1 \quad u_2 \quad u_3];$$

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}; \mathbf{B} = \begin{bmatrix} 1 \\ 1 \\ 3 \end{bmatrix}; \mathbf{C} = [1 \quad 1 \quad 1].$$

Отметим, что в прямой задаче три переменных (по числу столбцов \mathbf{A}), в двойственной задаче три переменных (по числу строк в \mathbf{A}). Отсюда следует, что если в \mathbf{A} много строк и мало столбцов (т. е. мало ограничений будет в двойственной задаче) – удобнее решать двойственную задачу, так как эмпирически число итераций $\sim 1-3m$, где m – число ограничений.

Условия дополнителности

Если $\mathbf{x}_0, \mathbf{u}_0$ – оптимальные планы задач (4.6), (4.7), то имеют место условия дополнителности

$$\begin{aligned} (\mathbf{A}[i, N] \times \mathbf{x}_0[N] - \mathbf{b}[i]) * u_0[i] &= 0, \forall i \in M_1; \\ (c[j] - \mathbf{u}_0[M] * \mathbf{A}[M, j]) * x_0[j] &= 0, \forall j \in N_1; \end{aligned}$$

или

$$\begin{aligned} \mathbf{A}[i, N] \times \mathbf{x}_0[N] &= b[i], \text{ если } u_0[i] > 0, i \in M_1; \\ \mathbf{u}_0[M] \times \mathbf{A}[M, j] &= c[j], \text{ если } x_0[j] > 0, j \in N_1. \end{aligned}$$

Обратное также справедливо.

Возврат от двойственных переменных к исходным

Если \mathbf{x}^* – оптимальное решение $\Leftrightarrow \exists \mathbf{U}^* = \mathbf{U}^*[M]$, то значения целевых функций равны $\langle \mathbf{b}, \mathbf{u}^* \rangle = \langle \mathbf{c}, \mathbf{x}^* \rangle$ и выполняются соотношения (4.7). Таким образом, если определили \mathbf{u}^* , можно найти \mathbf{x}^* через

$$\mathbf{c}[N_1] \times \mathbf{x}^*[N_1] + \mathbf{c}[N_2] \times \mathbf{x}^*[N_2] = \mathbf{u}^*[M_1] \times \mathbf{b}[M_1] + \mathbf{u}^*[M_2] \times \mathbf{b}[M_2].$$

Связь \mathbf{u} и \mathbf{x} следует из следующего. Сложим условия дополнителности

$$\sum_i \left(u_i^* b_i - u_i^* \sum_j a_{ij} x_j^* \right) - \sum_j x_j^* c_j - x_j^* \sum_i a_{ij} u_i^* = 0,$$

отсюда имеем

$$x_j^* \left(\sum_i a_{ij} u_i^* - c_j \right) - u_i^* \sum_j a_{ij} x_j^* = -u_i^* b_i;$$

$$\sum_j x_j^* c_j = \sum_i u_i^* b_i.$$

Для закрепления рассмотренного материала ниже приводится ряд примеров и решения к ним.

Пример 3

$$1 \cdot x_1 - 8x_2 - 4x_3 \rightarrow \min;$$

$$u_1 \left| \begin{array}{l} x_1 + x_2 + 4x_3 = 2; \\ x_1 - x_2 + 2x_3 = 0; \end{array} \right.$$

$$x_j \geq 0, j \in 1-3;$$

$$f(x) = \langle \mathbf{c}, \mathbf{x} \rangle \rightarrow \min; \quad g(u) = \langle \mathbf{u}, \mathbf{b} \rangle \rightarrow \max;$$

$$\mathbf{Ax} = \mathbf{b}; \quad \mathbf{uA} \leq \mathbf{c};$$

$$\mathbf{x} \geq 0; \quad \mathbf{u} - \text{любого знака.}$$

Проверить $\hat{x} = (1, 1, 0)$ на оптимальность.

1. Решение удовлетворяет ограничениям.
2. Решение оптимально?

Запишем двойственную задачу

$$2u_1 \rightarrow \max;$$

$$u_1 + u_2 \leq 1;$$

$$u_1 - u_2 \leq -8;$$

$$4u_1 + 2u_2 \leq -4,$$

так как все ограничения-равенства, то должны быть знаки «<>».

Условие дополнителъности для $\hat{\mathbf{x}}$

$$(\mathbf{c} - \mathbf{uA})x_0 = 0, x_0 > 0;$$

$$\mathbf{c} - \mathbf{uA} = 0;$$

$$u_1 + u_2 = 1;$$

$$u_1 - u_2 = -8$$

так как у первой и второй компонент $\hat{\mathbf{x}}$ – ненулевые значения. Отсюда выражаем u_1 и u_2 (либо одно через другое)

$$u_1 = -\frac{7}{2};$$

$$u_2 = \frac{9}{2};$$

$\hat{\mathbf{u}} = \left(-\frac{7}{2}, \frac{9}{2}\right)$. Проверим, является ли $\hat{\mathbf{u}}$ планом двойственной задачи: $4u_1 + 2u_2 = -5 < -4 \Rightarrow$ – да, является (выполнились ограничения).

Таким образом,

$\hat{\mathbf{u}} = \left(-\frac{7}{2}, \frac{9}{2}\right)$ – план двойственной задачи и для пары $\left\{\hat{\mathbf{x}}, \hat{\mathbf{u}}\right\}$ выполнено условие дополнителъности $\Rightarrow \hat{\mathbf{x}}$ – оптимальный план.

Пример 4

Проверить $\hat{\mathbf{x}} = (0, 1, 1, 0)$ на оптимальность в задаче

$$-x_1 + 3x_2 - 3x_3 + x_4 \rightarrow \max;$$

$$u_1 \left| \begin{array}{l} x_1 + x_2 + x_3 + x_4 \leq 2; \\ x_1 + 2x_2 - 4x_3 + x_4 \leq -1; \end{array} \right.$$

$$u_2 \left| \begin{array}{l} x_1 + x_2 + x_3 + x_4 \leq 2; \\ x_1 + 2x_2 - 4x_3 + x_4 \leq -1; \end{array} \right.$$

$$x_j \geq 0;$$

$$2u_1 - u_2 \rightarrow \min;$$

$$u_1 + u_2 \geq -1;$$

$$u_1 - 4u_2 \geq -3;$$

$$u_1 + u_2 \geq 1;$$

$$u_1 + 2u_2 \geq 3;$$

$$u_1 \geq 0, u_2 \geq 0.$$

Условие дополнителности

$$\left. \begin{array}{l} u_1 + 2u_2 = 3 \\ u_1 - 4u_2 = -3 \end{array} \right\} u_1 = u_2 = 1;$$

$\hat{\mathbf{u}} = (1, 1)$ – решение двойственной задачи

На векторе \mathbf{x} целевая функция равна 0, на $\hat{\mathbf{u}}$ равна 1.

Вторая часть условий дополнителности не выполнена $\Rightarrow \hat{\mathbf{x}}$ – не оптимальный (но близкий к оптимальному)

$$\left. \begin{array}{l} x_2 + x_3 = 2 \\ 2x_2 - 4x_3 = -1 \end{array} \right\} \Rightarrow x_2 = \frac{7}{6}, x_3 = \frac{5}{6} \Rightarrow \hat{\mathbf{x}} = \left(0, \frac{7}{6}, \frac{5}{6}, 0 \right)$$

– оптимальный план.

Проверим!

1. План задачи.

2. Ограничения $u_1 + 2u_2 \leq -1;$

$$u_1 - u_2 \leq 1;$$

$$2u_1 + u_2 \leq -1.$$

3. Условие дополнителности

$$2u_1 + u_2 = -1;$$

$$u_2 = -2u_1 - 1.$$

Тогда имеем ограничения $\Rightarrow \begin{cases} u_1 - 4u_1 - 2 \leq -1, \\ u_1 + 2u_1 + 1 \leq 1, \end{cases}$ откуда $\begin{cases} -3u_1 \leq 1; \\ 3u_1 \leq 0, \end{cases}$

$$\text{т. е. } u_1 \in \left[-\frac{1}{3}, 0 \right];$$

$\hat{\mathbf{u}} = (u_1, -2u_1 - 1)$, где $u_1 \in \left[-\frac{1}{3}, 0 \right]$ – план двойственной задачи и для $\left(\hat{\mathbf{x}}, \hat{\mathbf{u}} \right)$

– выполняется условие дополнителности $\Rightarrow \hat{\mathbf{x}}$ – оптимальный план.

Пример 5

Проверить $\hat{\mathbf{x}} = (1, 1, 0)$ на оптимальность для примера 3.

1. План задачи.

2. Условие дополнителъности

$$\begin{cases} u_1 + 2u_2 = -1 \\ u_1 - u_2 = 1 \end{cases} \Rightarrow u_1 = \frac{1}{3}, u_2 = -\frac{2}{3}.$$

Проверим $\hat{\mathbf{u}} = \left(\frac{1}{3}, -\frac{2}{3}\right)$ – план?

$2u_1 + u_2 = 0$ – условие ≤ -1 не выполняется, $\Rightarrow \hat{\mathbf{x}}$ – не оптимальный план, так как иначе ему нашлась бы пара $\hat{\mathbf{u}}$, но тогда для $\left(\hat{\mathbf{x}}, \hat{\mathbf{u}}\right)$ выполнилось бы условие дополнителъности.

Пример 6 (без перехода к двойственным переменным)

$$\begin{aligned} -x_1 + 4x_2 - 5x_3 &\rightarrow \min; \\ -x_1 + x_2 + x_3 &\geq -2; \\ 2x_1 + x_2 + x_3 &= 4; \\ x_1 \geq 0, x_2 &\geq 0. \end{aligned}$$

Воспользуемся методом перебора базисных решений (БР).

Замена

$$\begin{aligned} u_1 &= x_1; & x_1 &= u_1; \\ u_2 &= x_2; & x_2 &= u_2; \\ u_3 &= -x_1 + x_2 + x_3 + 2; & x_3 &= u_1 - u_2 + u_3 - 2. \end{aligned}$$

Следовательно, задача

$$\begin{aligned} -6u_1 + 9u_2 - 5u_3 + 10 &\rightarrow \min; \\ 3u_1 + u_3 &= 6; \\ u_1 \geq 0, u_2 \geq 0, u_3 &\geq 0. \end{aligned}$$

Два БР

$$(1, 0, 0); \\ (0, 0, 1) \quad (0, 0, 0); f^* = -20,$$

следовательно, решение $(0, 0, 4); f^* = -20$.

Задача 1

$$\begin{aligned} -x_1 + 3x_2 - 3x_3 + x_4 &\rightarrow \max; \\ x_1 + x_2 + x_3 + x_4 &\leq 2; \\ x_1 + 2x_2 - 4x_3 + x_4 &\leq -1; \\ x_j &\geq 0, \quad j = 1-4. \end{aligned}$$

Проверить $\hat{\mathbf{x}} = (0, 1, 1, 0)$ на оптимальность.

Задача 2

$$\begin{aligned} -x_1 + x_2 - x_3 &\rightarrow \min; \\ u_1 \mid x_1 + x_2 + 2x_3 &= 2; \\ u_2 \mid 2x_1 - x_2 + x_3 &= 1; \\ x_j &\geq 0, \quad j = 1, 3. \end{aligned}$$

Проверить $\hat{\mathbf{x}} = (0, 0, 1)$ на оптимальность.

5. МЕТОДЫ ЛИНЕАРИЗАЦИИ ДЛЯ ЗАДАЧ УСЛОВНОЙ ОПТИМИЗАЦИИ

Нелинейную функцию $f(x)$ обычного вида можно линеаризовать в окрестности некоторой точки x'

$$f(x) = f(x') + \nabla f(x')(x - x') + O(\|x - x'\|^2).$$

Несмотря на то, что линеаризация является очень грубым приближением, такой прием часто используют с целью применения методов ЛП при решении задач высокой размерности. В данном разделе рассматривается использование линеаризации в задаче с линейными ограничениями.

Рассмотрим следующую задачу:

$$f(x) \rightarrow \min, \dots, \mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \geq 0.$$

Эта задача может быть преобразована к виду

$$f(x, x') \rightarrow \min, \dots, \mathbf{Ax} \leq \mathbf{x} \geq 0,$$

где $f(x, x')$ – линейная часть разложения $f(x)$.

Эта задача ЛП и должна иметь решение $\bar{\mathbf{x}}^*$ в допустимой угловой точке. Важен вопрос о соотношении \mathbf{x}^* и $\bar{\mathbf{x}}^*$ (истинное решение \mathbf{x}^* может лежать внутри допустимой области), т. е. даже в случае выпуклой функции $f(x)$ необходима коррекция $\bar{\mathbf{x}}^*$. Поскольку точка $\bar{\mathbf{x}}^*$ доставляет минимум функции \bar{f} , то

$$\bar{f}(x^0, x^0) \bar{f}(\bar{\mathbf{x}}^*, x^0), \text{ т. е. } f(x^0) \setminus f(x^0) + \nabla f(x^0)(\bar{\mathbf{x}}^* - x^0),$$

или $\nabla f(x^0)(\bar{\mathbf{x}}^* - \mathbf{x}^0) \langle 0$. Очевидно, вектор $\bar{\mathbf{x}}^* - \mathbf{x}^0$ задает направление спуска. Одномерный поиск должен проводиться на прямой $x^{(1)} = x^{(0)} + \alpha(\bar{\mathbf{x}}^* - \mathbf{x}^{(0)})$, $0 \leq \alpha \leq 1$.

Решение задачи $\min f(x^{(1)} + \alpha(\bar{\mathbf{x}}^* - x^{(0)}))$ позволяет найти α и, следовательно, $\mathbf{x}^{(1)}$ такое, что $f(x^{(1)}) \setminus f(x^{(0)})$. Точка $\mathbf{x}^{(1)}$ может служить точкой линеаризации для построения следующей аппроксимации (так как $\nabla f(x^{(1)}) \neq 0$).

5.1. Алгоритм Франка–Вульфа

Задать $\mathbf{x}^{(0)}$, $\varepsilon > 0$, $\delta > 0$, $t = 0$.

1. Вычислить $\nabla f(\mathbf{x}^{(t)})$, где $\mathbf{x}^{(t)}$ – точка линеаризации.
Если $\|\nabla f(\mathbf{x}^{(t)})\| < \varepsilon$ прекратить вычисления, иначе перейти к п. 2.
2. Решить следующую задачу ЛП

$$\nabla f(\mathbf{x}^{(t)})^T \mathbf{z} \rightarrow \min,$$

при $\mathbf{A}\mathbf{z} \leq \mathbf{b}$, $\mathbf{z} \geq 0$. Пусть $\mathbf{z}^{(t)}$ – оптимальное решение этой задачи.

Здесь опущены постоянные слагаемые в $f(\mathbf{x}, \mathbf{x}^{(t)})$.

3. Найти шаг $\alpha^{(t)}$ из решения задачи

$$\min f(\mathbf{x}^{(t)} + \alpha^{(t)}(\mathbf{z}^{(t)} - \mathbf{x}^{(t)})), \quad 0 \leq \alpha \leq 1.$$

4. Вычислить $\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} + \alpha^{(t)}(\mathbf{z}^{(t)} - \mathbf{x}^{(t)})$.

5. Если $\|\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)}\| < \delta$ и $|f(\mathbf{x}^{(t+1)}) - f(\mathbf{x}^{(t)})| \leq \varepsilon |f(\mathbf{x}^{(t+1)})|$, закончить, иначе перейти к п. 1.

Пример

$$f(x) = x_1^2 + 2x_2^2 - 2x_1 - 4x_2 \rightarrow \min;$$

$$x_1 + x_2 \leq 8;$$

$$2x_1 - x_2 \leq 12, \quad x \geq 0.$$

1-я итерация

$$\frac{\partial f}{\partial x} = (2x_1 - 2; 4x_2 - 4).$$

Начальная точка $\mathbf{x}^{(0)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $\varepsilon = 0,001$.

2-я итерация

$$\left(\frac{\partial f}{\partial x_1} \right)_{x_0} z_1 + \left(\frac{\partial f}{\partial x_2} \right)_{x_0} z_2 = -2z_1 - 4z_2 \rightarrow \min;$$

$$z_1 + 2z_2 \leq 8;$$

$$2z_1 - z_2 \leq 12, \quad z \geq 0.$$

Решение симплекс-методом дает $\mathbf{z}^0 = \begin{pmatrix} 0 \\ 4 \end{pmatrix}$.

Выполним одномерный поиск в направлении

$$x^{(1)} = \begin{pmatrix} x_1^{(0)} \\ x_2^{(0)} \end{pmatrix} + \alpha \begin{pmatrix} 0 \\ 4 \end{pmatrix} = \begin{pmatrix} x_1^{(1)} \\ x_2^{(1)} \end{pmatrix},$$

$$\text{т. е. } x_1^{(1)} = x_1^{(0)} + \alpha p_1^{(0)} = 0,$$

$$x_2^{(1)} = x_2^{(0)} + \alpha p_2^{(0)} = 0 + \alpha 4.$$

Итак, $f(\alpha) = 2(4\alpha)^2 - 4(4 \cdot 4) \rightarrow \min_{\alpha};$

$$\frac{\partial f}{\partial \alpha} = 64\alpha - 16 = 0 \rightarrow \alpha = \frac{1}{4}.$$

Таким образом, $\mathbf{x}^{(1)} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$

Теперь можно линеаризовывать в точке $(0, 1).$

На 3-й итерации получим

$$\mathbf{x} = \begin{pmatrix} 0,99528 \\ 0,96321 \end{pmatrix}.$$

Решение задачи $\mathbf{x}^* = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$

Упражнение

Проверить решение на оптимум с помощью локальных условий Куна–Таккера.

6. СЕПАРАБЕЛЬНОЕ ПРОГРАММИРОВАНИЕ (СП)

Метод СП позволяет преобразовывать нелинейные задачи оптимизации

определенного типа, а именно $f(x) = \sum_{i=1}^n f_i(x_i)$ к задаче, содержащей

только линейные функции. При этом используется специальная модификация симплекс-алгоритма. Обоснованием метода служит то обстоятельство, что в качестве хорошего приближения нелинейной функции на большом отрезке можно использовать ее кусочно-линейную аппроксимацию.

1. Кусочно-линейная аппроксимация.

Рассмотрим функцию одной переменной (рис. 20, в). На интервале изменения функции построена сетка, имеющая K узлов. Обычно используется равномерная сетка, хотя это и не обязательно. Узлы лучше располагать в окрестности точки перегиба и чаще в местах быстрого изменения функции (надо иметь график). Уравнение прямой, проходящей через две точки $(x^{(k)}, f^{(k)})$ и $(x^{(k+1)}, f^{(k+1)})$, имеет вид

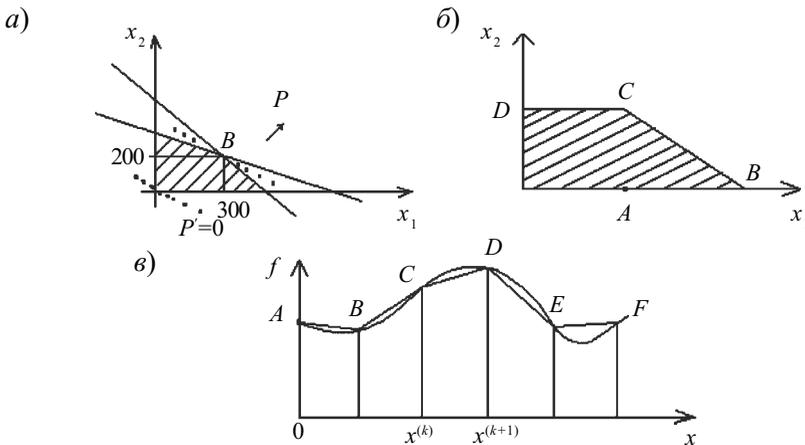


Рис. 20

$$\bar{f}(x) = f^{(k)} + \frac{f^{(k+1)} - f^{(k)}}{(x^{(k+1)} - x^{(k)})}(x - x^{(k)}).$$

Всего имеем K узлов, поэтому исходная функция аппроксимируется с помощью $(K - 1)$ -й линейной функции на соответствующих интервалах. Введение такой аппроксимации приводит к росту размерности задачи (за счет добавления новых переменных).

Заметим, что такую аппроксимацию используют также как первый этап для поиска глобального минимума произвольной кривой. Алгоритм основан на последовательном увеличении числа подынтервалов до тех пор, пока не будет достигнуто качественное соответствие исходной функции и ее аппроксимации. В подозрительных интервалах (где функция меньше) осуществляются линейные поиски.

Вернемся к задаче СП. Совокупность рассмотренных линейных функций записывают в некотором общем виде, учитывая, что любое значение x в интервале $x^{(k)} \leq x \leq x^{(k+1)}$, можно записать как $x = \lambda^{(k)} x^{(k)} + \lambda^{(k+1)} x^{(k+1)}$, где $\lambda^{(k)}, \lambda^{(k+1)}$ – неотрицательные числа, сумма которых равна 1.

В общем случае K узлов имеем

$$x = \sum_{k=1}^K \lambda^{(k)} x^{(k)}, \quad \tilde{f}(x) = \sum_{k=1}^K \lambda^{(k)} f^{(k)};$$

$$\sum_{k=1}^K \lambda^{(k)} = 1, \quad \lambda^{(k)} \geq 0, \quad k = \overline{1, K};$$

$\lambda^{(i)} \lambda^{(j)} = 0; j > i + 1; i = \overline{1, K - 1}$ (это означает, что не более двух соседних переменных λ положительны, а остальные нулевые, так как любое значение x лежит только на одном подынтервале и лишь две переменные λ , соответствующие этому подынтервалу, могут отличаться от нуля).

Метод кусочно-линейной аппроксимации обобщается только на слу-

чай нелинейной функции n переменных вида $f(x) = \sum_{i=1}^n f_i(x_i)$.

К такому виду можно приводить в ряде случаев, используя замены переменных, например $x_1 x_2$ можно записать, как $x_3^2 - x_4^2$, полагая

$$x_3 = \frac{1}{2}(x_1 + x_2), \quad x_4 = \frac{1}{2}(x_1 - x_2).$$

Для случая n переменных годятся приведенные выше соотношения (добавить индекс)

$$\tilde{f}(x) = \sum_{i=1}^n \left(\sum_{k=1}^{K_i} \lambda_i^{(k)} f_i^{(k)} \right), \quad x_i = \sum_{k=1}^{K_i} \lambda_i^{(k)} x_i^{(k)}, \quad i = \overline{1, n};$$

$$\sum_{k=1}^{K_i} \lambda_i^{(k)} = 1, \quad \lambda_i^{(k)} \geq 0, \quad k = \overline{1, K_i};$$

$$\lambda_i^{(i)} \lambda_i^{(j)} = 0, \quad j > i+1, \quad i = \overline{1, K_i - 1}.$$

Пример

$$f(x) = f_1(x_1) + f_2(x_2) = 1 - (1 - x_1)^3 + 0,1 x_2^4;$$

$$0 \leq x_1 \leq 2, \quad 0 \leq x_2 \leq 3.$$

Алгоритм

1. Определить точки разбиения $x_{ik}, k = \overline{0, K_i}$.

Пусть $k_1 = 4$, тогда, например, для x_1 $0; 0,5; 1,5; 2,0$

$$x_1 = 0\lambda_1^{(1)} + 0,5\lambda_1^{(2)} + 1,5\lambda_1^{(3)} + 2,0\lambda_1^{(4)},$$

(т. е. по $\lambda_i^{(k)}$ можно восстановить x_i)

$$\tilde{f}_1(x_1) = \tilde{f}_1(\lambda_1^{(k)}) = f_1(0)\lambda_1^{(1)} + f_1(0,5)\lambda_1^{(2)} \dots$$

Аналогично для $\tilde{f}_2(x_2)$ $\tilde{f} = \tilde{f}_1 + \tilde{f}_2$.

2. Вычисляются значения всех функций в точках разбиения $f_i(x_i), g_i(x_i)$.

3. Для решения сепарабельных задач с переменными $\lambda_i^{(k)}$ применим симплекс-метод, на стандартные условия которого при вводе переменных в базис накладываются дополнительные условия, а именно: в базис вводится ненулевая переменная, соседняя по отношению к имеющейся уже там переменной $\lambda_i^{(k)}$ – единственной, не равной нулю. Обе переменные должны относиться к одному x_i . Если это не выполняется, надо подыскать для ввода в базис другую переменную. По найденным λ^* определить значения переменных x^* и повторить с новым разбиением.

Замечания

1. Точность решения задачи, полученной методом сепарабельного программирования, не превосходит точности ее аппроксимации (число узлов, их расположение).

2. Метод нельзя применять к общим задачам нелинейного программирования. Он подходит для решения задач, которые в основном линейны (отклонения от линейности представлены сепарабельными функциями).

3. Если не требуется высокой точности, приближенное решение можно получить при помощи однократного решения подзадачи линейного программирования.

Упражнение

Составить задание для применения симплекс-метода при решении задачи

$$f(x) = x_1 + x_2^4 \rightarrow \max; 3x_1 + 2x_2^2 \leq 9;$$
$$x_1, x_2 \geq 0.$$

Обозначим

$$f_1(x_1) = x_1, f_2(x_2) = x_2^4, g_1^1(x_1) = 3x_1, g_2^1(x_2) = 2x_2^2.$$

Выполнить кусочно-линейную аппроксимацию функции $f_2(x_1)$ и $g_2^1(x_2)$ в $[0, 3]$.

7. ГЕОМЕТРИЧЕСКОЕ ПРОГРАММИРОВАНИЕ (ГП)

ГП – частный случай нелинейного программирования, применим тогда, когда целевая функция имеет специфический вид, состоящий в том, что она представляет собой сумму положительных слагаемых, которые являются произведениями переменных в произвольных степенях. Важно, что в этом случае решение может быть получено в формульном или замкнутом виде (по крайней мере, в таком виде получают оценку решения). Иначе говоря, можно получить решение, недоступное численным методам.

Если в целевой функции присутствует слагаемое со знаком минус, то подобные задачи решаются методом обобщенного геометрического программирования.

7. 1. Постановка задачи

Основа метода – геометрическое неравенство Коши (среднее геометрическое не превышает среднего арифметического), например,

$$\frac{\gamma_1 + \gamma_2}{2} \geq \sqrt{\gamma_1 \gamma_2};$$
$$\frac{2}{3} \gamma_1 + \frac{1}{3} \gamma_2 \geq \gamma_1^{\frac{2}{3}} \gamma_2^{\frac{1}{3}}.$$

В общем случае справедливо

$$\sum_{i=1}^l \gamma_i \delta_i \geq \prod_{i=1}^l \gamma_i^{\delta_i},$$

где $\gamma_i > 0$, $\delta_i > 0$, $\sum_{i=1}^l \delta_i = 1$; γ_i , δ_i – некоторые числа.

Равенство достигается, если $\gamma_1 = \gamma_2 = \dots = \gamma_l$. Рассмотрим j -е слагаемое $\gamma_j = \delta_j = u_j$, тогда неравенство примет вид

$$\sum_{j=1}^l u_j \geq \prod_{j=1}^l \left(\frac{u_j}{\delta_j} \right)^{\delta_j}. \quad (7.1)$$

Неравенство (7.1) обращается в равенство при $\frac{u_j}{\delta_j} = \text{const}$, $i = \overline{1, l}$.

Можно показать, что из $\frac{a}{b} = \frac{c}{d}$ следует $\frac{a}{b} = \frac{a+b}{b+d}$. Тогда учитывая, что $\gamma_i = \text{const}$, можно записать

$$\frac{u_j}{\delta_j} = \frac{u_{j+1}}{\delta_{j+1}}, \quad \frac{u_1}{\delta_1} = \frac{u_2}{\delta_2} = \frac{u_1 + u_2}{\delta_1 + \delta_2} = \frac{u_3}{\delta_3}, \quad \text{т. е. } \frac{u_j}{\delta_j} = \frac{\sum u_j}{\sum \delta_j} = \sum u_j.$$

Итак, неравенство (7.1) обращается в равенство, если

$$\delta_j = \frac{u_j}{\sum u_j}. \quad (7.2)$$

Пример

$$f(x) = 4x_1^{-1}x_2^{-1}x_3^{-1} + 4x_1x_3 + x_1x_2 + 2x_2x_3 \rightarrow \min;$$

$$p_1(x) = x_1^{-1}x_2^{-1}x_3^{-1}, \quad p_2(x) = x_1x_3, \quad c = (4, 4, 1, 2),$$

где $f(x)$ – полиномиальная функция (слагаемые полиномы).

Матрица экспонент

$$\mathbf{A} = \begin{matrix} & \begin{matrix} x_1 & x_2 & x_3 \end{matrix} & \text{(переменные)} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{matrix} | & -1 & -1 & -1 \\ | & 1 & 0 & 1 \\ | & 1 & 1 & 0 \\ | & 0 & 1 & 1 \end{matrix} & \end{matrix};$$

(слагаемые)

$$f(x) = \sum_{j=1}^l c_j p_j(x) \rightarrow \min_x \text{ – (прямая задача),}$$

где $p_j(x) = \prod_{k=1}^n x_k^{a_{jk}}$.

Построим целевую функцию двойственной задачи

$$\prod_{j=1}^l \left(\frac{u_j}{\delta_j} \right)^{\delta_j} = \prod_{j=1}^l \left(\frac{c_j \prod_{k=1}^n x_k^{a_{jk}}}{\delta_j} \right)^{\delta_j} = \prod_{j=1}^l \left(\frac{c_j}{\delta_j} \right)^{\delta_j} \prod_{k=1}^n x_k^{\sum a_{jk} \delta_j} = z(\delta).$$

Из неравенства (7.1) имеем $f(x) \geq z(\delta)$.

Наложим ограничения так, чтобы исключить переменную x , т. е.

$$\sum_{j=1}^l a_{jk} \delta_j = 0,$$

тогда $z(\delta) = \prod_{j=1}^l \left(\frac{c_j}{\delta_j} \right)^{\delta_j}$ – целевая функция двойственной задачи.

При условии

$$\left. \begin{aligned} \sum_{j=1}^l a_{jk} \delta_j &= 0, \quad k = \overline{1, n}; \\ \sum_{j=1}^l \delta_j &= 1, \quad \delta_j > 0, \quad j = \overline{1, l}. \end{aligned} \right\} \quad (7.3)$$

Решение прямой $f(x) \rightarrow \min$ и двойственной $z(\delta) \rightarrow \max$ задач в точке оптимума совпадают (значения функций f и z совпадают, когда неравенство обращается в равенство, т. е. при условии (7.2)).

Исходная функция в общем случае невыпуклая и может иметь несколько локальных минимумов (поэтому лучше перейти к двойственной задаче).

Функция $z(\delta)$ всегда вогнута, а ограничения (7.3) линейны, следовательно, $z(\delta)$ имеет глобальный максимум и могут быть применены алгоритмы вогнутого программирования.

Если число слагаемых $l = n+1$, то число переменных δ_j равно числу линейных ограничений (7.3). В этом случае уравнение для ограничений имеет единственное решение, являющееся оптимальным решением двойственной задачи.

Степень трудности задачи определяется, как $d = l - (n+1)$. Например, нулевая степень трудности (число слагаемых на единицу больше числа переменных). В этом случае ограничения (7.3) – система уравнений с l неизвестными имеет единственное решение. Если неизвестно z^* , то согласно теории двойственности

$$z^* = f^* = \sum_{j=1}^l u_j^*.$$

Если найдено δ_j^* , то из (7.2) можно найти значение j -го слагаемого функции $f(x)$, а именно из $\frac{u_j^*}{\delta_j^*} j = \sum_j u_j^* = f^*$ следует $u_j^* = f^* \delta_j^*$. Заметим, что, с другой стороны, $u_j^* = c_j p_j(x^*)$, т. е. теперь можем найти x^* . Смысл двойственных переменных δ_j^* : они определяют относительный вклад j -го слагаемого целевой функции в оптимальное решение $u_j^* = f^* \delta_j^*$ (или процент от f^*).

Задача («перевозка сыпучего материала»)

Выбрать размеры контейнера для перевозки песка в м³. Стоимость одного рейса a рублей. После перевозки контейнер выбрасывается. Стоимость материала на дно b рублей, на стенки C , р./м². Выбрать размеры контейнера так, чтобы минимизировать затраты

$$f(x) = \frac{Ba}{x_1 x_2 x_3} + b x_1 x_2 + \left(\begin{array}{c} \text{число рейсов,} \\ \text{транспортные расходы} \end{array} \right) + \underbrace{b x_1 x_2 + 2C x_1 x_3 + 2C x_2 x_3}_{\text{(расходы на материалы)}} \rightarrow \min_x;$$

$$z(\delta) = \left(\frac{Ba}{\delta_1} \right)^{\delta_1} \left(\frac{b}{\delta_2} \right)^{\delta_2} \left(\frac{2c}{\delta_3} \right)^{\delta_3} \left(\frac{2c}{\delta_4} \right)^{\delta_4} \rightarrow \max_{\delta};$$

$$\begin{cases} \delta_1 + \delta_2 + \delta_3 + \delta_4 = 1; \\ \delta \mathbf{A} = 0; \end{cases}$$

$$(\delta_1 \delta_2 \delta_3 \delta_4) \cdot \begin{array}{c|ccc} & x_1 & x_2 & x_3 \\ \hline 1 & -1 & -1 & -1 \\ 2 & 1 & 1 & 0 \\ 3 & 1 & 0 & 1 \\ 4 & 0 & 1 & 1 \end{array} = 0;$$

$$d = l - (u + 1) = 4 - 4 = 0,$$

ПОЭТОМУ

$$\begin{cases} -\delta_1 + \delta_2 + \delta_3 = 0; \\ -\delta_1 + \delta_2 + \delta_4 = 0; \\ -\delta_1 + \delta_3 + \delta_4 = 0; \\ -\delta_1 + \delta_2 + \delta_3 + \delta_4 = 1; \end{cases} \rightarrow \delta^* = \left(\frac{2}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}\right) - \text{оптимальное решение};$$

$$f^* = z^* = \prod_{j=1}^4 (c_j / \delta_j^*)^{\delta_j^*} = \left(\frac{5}{2} Ba\right)^{2/5} (5b10c10c)^{1/5} = 5(Bac)^{2/5} b^{1/5}.$$

Найдем, x_1^*, x_2^*, x_3^* учитывая, что $c_j p_j(x^*) = \delta_j^* f^*$,

$$bx_1^* x_2^* = \frac{1}{5} f^*, \quad 2cx_1^* x_3^* = \frac{1}{5} f^*, \quad 2cx_2^* x_3^* = \frac{1}{5} f^*.$$

Пусть $x_1 = x_2$, тогда $bx_1^{*2} = \frac{1}{5} f^*$, а так как $f^* = z^*$, то

$$x_1^* = x_2^* = \left(\frac{Bac}{b}\right)^{1/5}, \quad x_3^* = \frac{1}{2} \left(\frac{Bab^3}{c^4}\right)^{1/5}.$$

7.2. Решение задачи геометрического программирования с ограничениями

Предполагается, что ограничения здесь также имеют вид позиномов

$$f(x) \rightarrow \min, \quad g_i(x) \leq 1, \quad i = \overline{1, m}.$$

Пусть $c = (c_1, \dots, c_l)$, где l – общее число слагаемых $f(x)$ и $g_i (i = \overline{1, m})$, записанных подряд $\delta = (\delta_1, \dots, \delta_l)$.

Введем множества индексов I_k $k = \overline{0, m}$. Здесь индекс 0 относится к целевой функции. Каждое множество I_k содержит порядковые номера слагаемых, относящихся к k -й функции (или к k -му позиному).

Построим матрицу экспонент

$$\mathbf{A} = \begin{matrix} & x_1, \dots, x_n \\ \delta_1 & \square \\ \vdots & \\ \delta_l & \square \end{matrix}.$$

Введем множители Лагранжа λ_k ($k = \overline{1, m}$), которые соответствуют ограничениям. Множитель λ_k , как известно, характеризует чувствительность целевой функции к вариантам k -го ограничения. Если в оптимальной точке какое-либо ограничение выполняется как неравенство, то оно не влияет на решение и целевая функция к его вариации не чувствительна. В этом случае соответствующий множитель Лагранжа равен нулю.

Доказано, что численно

$$\lambda_k = \sum_{i \in I_k} \delta_i, \delta_i \geq 0,$$

где δ_i относятся к k -му ограничению $\lambda_0 = \sum_{i \in I_0} \delta_i = 1$, (δ_i — относящиеся к

целевой функции).

Двойственная задача имеет вид

$$z(\delta) = \prod_{i=1}^l \left(\frac{c_i}{\delta_i} \right)^{\delta_i} \prod_{k=0}^m \lambda_k^{\lambda_k} \rightarrow \max_{\delta} \quad \delta \mathbf{A} = 0, \lambda_0 = 1.$$

Пример построения двойственной задачи

$$f(x) = 40x_1x_2 + 20x_2x_3 \rightarrow \min;$$

$$g_1(x) = \frac{1}{5}x_1^{-1}x_2^{-1/2} + \frac{3}{5}x_2^{-1}x_3^{-2/3} \leq 1;$$

$$l = 4, n = 3, \text{ поэтому } d = l - (n+1) = 0, m = 1;$$

$$I_0 = (1, 2), I_1 = (3, 4);$$

$$c = (40, 20, \frac{1}{5}, \frac{3}{5}), \delta = (\delta_1, \delta_2, \delta_3, \delta_4);$$

$$\lambda_0 = \delta_1 + \delta_2; \lambda_1 = \delta_3 + \delta_4;$$

$$z(\delta) = \left(\frac{40}{\delta_1}\right)^{\delta_1} \left(\frac{20}{\delta_2}\right)^{\delta_2} \left(\frac{1/5}{\delta_3}\right)^{\delta_3} \left(\frac{3/5}{\delta_4}\right)^{\delta_4} \lambda_0^{\lambda_0} \lambda_1^{\lambda_1};$$

$$(\delta_1 \delta_2 \delta_3 \delta_4) \cdot \begin{array}{c} x_1 \quad x_2 \quad x_3 \\ \left| \begin{array}{ccc} 1 & 1 & 0 \\ 0 & 1 & 1 \\ -1 & -1/2 & 0 \\ 0 & -1 & -2/3 \end{array} \right| \end{array};$$

$$\delta_1 - \delta_3 = 0; \quad \delta_1 + \delta_2 - \frac{1}{2} \delta_3 - \delta_4 = 0; \quad \delta_2 - \frac{2}{3} \delta_4 = 0;$$

$\delta_1 + \delta_2 = 1$ (пишутся те, которые относятся только к целевой функции)

$$\left[\begin{array}{cccc|c} 1 & 0 & -1 & 0 & 0 \\ 1 & 1 & -1/2 & -1 & 0 \\ 0 & 1 & 0 & -2/3 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{array} \right] \rightarrow \left[\begin{array}{cccc|c} 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1/2 & -1 & 0 \\ 0 & 1 & 0 & -2/3 & 0 \\ 0 & 1 & 1 & 1 & 1 \end{array} \right]$$

Доказаны следующие результаты о соотношении оптимальных решений прямой и двойственной задач геометрического программирования с ограничениями при условиях

- прямая задача разрешима;
- существует оптимальное решение;
- существует $x^0 : g_i(x^0) < 1, i = \overline{1, m}$ (условие Слейтера).

Тогда оптимальные значения целевых функций пары двойственных задач совпадают. Если $g_k(x^*) < 1$ в оптимальной точке, то $\delta_i^* = 0, i \in I_k$.

Если δ^* – максимизирующий вектор двойственной задачи, то любой минимизирующий вектор x^* прямой задачи удовлетворяет системе уравнений

$$u_i = c_i p_i(x^*) = \begin{cases} \delta_i^* f^*, & i \in I_0; \\ \frac{\delta_i^*}{\lambda_k}, & i \in I_k, \lambda_k > 0, k = \overline{1, m}. \end{cases}$$

Замечание

Если не все ограничения в оптимальной точке являются активными, (т. е. не все выполняются как равенства), то при поиске решения численным методом могут возникать трудности при поиске максимума двойственной задачи, так как некоторые переменные $\delta_i = 0$, а в точке, где $\delta_i = 0$, двойственная целевая функция становится недифференцируемой (содержит сомножители $\left(\frac{c_i}{\delta_i}\right)^{\delta_i}$).

Задача

1. Из круглого бревна диаметром $d = \frac{1}{\sqrt{c}}$ необходимо выпилить балку прямоугольного сечения так, чтобы величина bh^2 (характеризует ее сопротивляемость изгибу) была максимальна

$$f(b, h) = \frac{1}{bh^2} \rightarrow \min;$$

$$b^2 + h^2 \leq 1/c; \quad cb^2 + ch^2 \leq 1.$$

2. Свести к задаче геометрического программирования

$$f(x) = x_1 + \frac{1}{x_2 \sqrt{x_1 - x_2}} \rightarrow \min;$$

$$x_1 > x_2 > 0; \quad x_1 - x_2 \geq x_3, \quad x_3 > 0.$$

Указание: если ввести переменную (такое преобразование не искажает результаты решения), получим задачу ГП

$$f(x) = x_1 - x_2^{-1} x_3^{-0.5} \rightarrow \min;$$

$$x_1 \geq x_2 + x_3; \quad x_1^{-1} x_2 + x_1^{-1} x_3 \leq 1.$$

С той же целью рассмотрим задачи

$$f(x) = (x_1 + x_2)^{3/2} (x_2^{-2} + x_2^{-1}) \rightarrow \min;$$

$$x_1 + x_2 \leq x_3;$$

$$f(x) = x_2^{-2} x_3^{3/2} + x_2^{-1} x_3^{3/2} \rightarrow \min;$$

$$x_1 x_3^{-1} + x_2 x_3^{-1} \leq 1.$$

8. СВЕДЕНИЕ ЗАДАЧ ВЕКТОРНОЙ ОПТИМИЗАЦИИ К ОДНОКРИТЕРИАЛЬНЫМ ЭКСТРЕМАЛЬНЫМ ЗАДАЧАМ

В математической модели принятия рационального решения, связанной с задачей многокритериальной оптимизации, цель проектирования определяется совокупностью непрерывно дифференцируемых функций $f_i(x)$, $i = \overline{1, s}$, поэтому задача $\min_{x \in D} \mathbf{f}(x)$ приводит к не-

обходимости решения задач $\min_{x \in D} f_1(x); \dots; \min_{x \in D} f_s(x)$.

Выбор любого значения $x \in D$ соответствует конкретным значениям критерия $\mathbf{f}(x) = (f_1(x), \dots, f_s(x))$, т. е. в пространстве критериев R^S области допустимых решений D можно поставить в соответствие множество (область критериев)

$$D_f = \left\{ \mathbf{f} = (f_1 \dots f_s) \left(f_i = f_i(x), i = \overline{1, s}; x \in D \right) \right\}.$$

На рис. 21, *a* и *б* показана допустимая область решений $D \in R^2$ и соответствующая ей область критериев $D_f \in R^2$

$$\begin{aligned} \min_{x \in D} f_1(x) &= \min_{x \in D} (-5x_1 + 2x_2 + 36); \\ \min_{x \in D} f_2(x) &= \min_{x \in D} (x_1 - 4x_2 + 20), \end{aligned}$$

где $D = \{x | 0 \leq x_1 \leq 6; 0 \leq x_2 \leq 4; -x_1 + x_2 \leq 3; x_1 + x_2 \leq 8\}$.

Говорят, если векторный критерий $f^k = f(x^k) = (f_1^k, \dots, f_s^k)$, доминирует векторный критерий $f^l = f(x^l) = (f_1^l, \dots, f_s^l)$, $f^k, f^l \in D_f$. Если выполняется система неравенств $f_i^k \leq f_i^l$ для всех $i = \overline{1, s}$ и хотя бы для одного частного критерия выполняется строгое неравенство $f_j^k < f_j^l$, то при переходе от f^l к f^k ничего не будет проиграно ни по одному частному критерию f_i^k , но по j -му частному критерию f_j^k будет получен выигрыш. Множество критериев $D_f \in D$, для которых справедлив принцип доминирования, образуют подмножество $D_c \subseteq D_f$, называемое областью согласия. В качестве примера рассмотрим рис. 21, *в* и *г*,

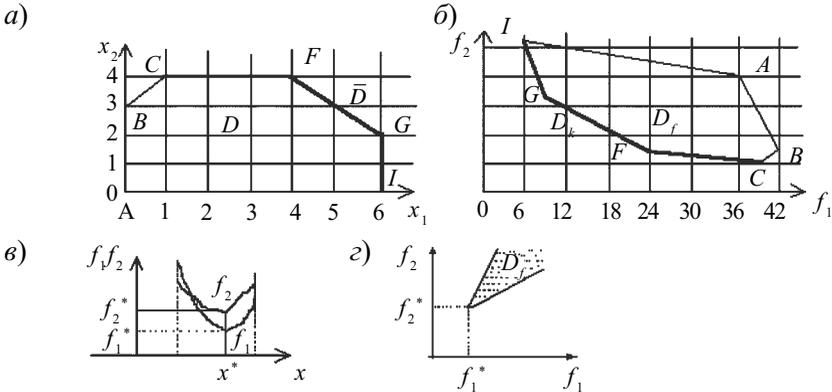


Рис. 21

где в данном случае D_e совпадает с D_f (уменьшение f_1 не приводит к увеличению f_2). Точки \bar{x} , для которых векторный критерий $f(x) \in D_f$, не удовлетворяет принципу доминирования относительно критерия $f(x)$, называются эффективными точками, а именно: не существует ни одной точки $x \in D$, такой, что $f_i(x) \leq f_i(\bar{x})$ для всех $i = 1, s$ и хотя бы для одной j выполняется строгое неравенство.

В эффективных точках векторный критерий $f(x)$ является не уменьшаемым одновременно по совокупности всех частных критериев f_i . Эти точки называют решениями, оптимальными по Парето. Множество эффективных точек образует подмножество $D_e \subseteq D$, называемое областью решений, оптимальных по Парето.

8.1. Решения, оптимальные по Парето

Множество в пространстве критериев, соответствующее множеству всех эффективных точек D_e , образует подмножество $D_k \subseteq D_f$, называемое областью компромиссов. В области D_k не выполняется принцип доминирования, а частные критерии являются противоречивыми.

Для любой пары эффективных точек (волнистая линия $CFG I$ на рис. 21, а) для соответствующих значений критериев $f(x^1), f(x^2) \in D_k$ (волнистая $CFG I$ на рис. 21, б) выполняется система неравенств

$$\text{если } f_1(x^1) > f_1(x^2), \text{ то } f_2(x^1) < f_2(x^2),$$

или

$$\text{если } f_1(x^1) < f_1(x^2), \text{ то } f_2(x^1) > f_2(x^2).$$

Что приводит к необходимости введения компромисса между частными критериями критериев $f(x^1)$ и $f(x^2) \in D_k$, чтобы решить, какой из них считать важнее?

Заметим, что D_f можно разделить на D_c и D_k , $D_f = D_c \cup D_k$;

$$D_c \cap D_k = 0.$$

Векторные критерии, представляющие интерес в многокритериальной задаче оптимизации, находятся в области D_k . Поэтому под рациональным решением будем понимать одну из эффективных точек $\mathbf{x} \in \bar{D}$, которая является предпочтительной с точки зрения лица, принимающего решение (ЛПР). Для того, чтобы определить какую из эффективных точек выбрать, требуется вводить информацию о важности частных критериев.

8.2. Обобщенные критерии оптимальности

Для учета важности значений векторного критерия $f(x)$, вычисленного в двух разных точках \mathbf{x}^l и \mathbf{x}^k , введем операцию бинарного отношения предпочтения \succ (для попарного сравнения критериев).

Отношение $f^k \succ f^l$, что f^k лучше f^l . Если нельзя выделить ни одно значение критерия, то эти значения эквивалентны ($f^k \sim f^l$). Из предпочтения между векторными критериями однозначно следует соответствующее отношение для эффективных точек, т. е. из $f^k \succ f^l \rightarrow \mathbf{x}^k \succ \mathbf{x}^l$. При решении однокритериальных задач оптимизации часто возникает необходимость в преобразовании вектора критерия f оператором $\Psi(\Psi_1, \dots, \Psi_s)$ в эквивалентный ему по важности другой вектор $\Psi(f) = (\Psi_1(f_1), \dots, \Psi_s(f_s))$, например, более удобный с вычислительной точки зрения. Причем, так как $f \sim \Psi(f)$, то из $f^k \succ f^l$ следует $\Psi(f^k) \succ \Psi(f^l)$.

В частности, преобразование $\Psi_i(f_i) = f_i^2$ позволяет заменить недифференцируемую функцию $f_i = (g_i(x))$ эквивалентным ей по важности выражением $\Psi_i(f_i) = g_i^2(x)$ (где Ψ_i – дифференцируемая функция).

Одним из применений преобразования Ψ является нормализация частных критериев f_i (приведение их к единому, безразмерному виду). Это необходимо для их ранжирования, так как $f_i(x)$ характеризуют физические свойства объекта проектирования и могут измеряться в различных масштабах и системах единиц:

$$\bar{f}_i = \Psi_i(f_i) = c_i f_i(x) + d_i, \quad c_i > 0,$$

где $c_i = \frac{1}{f_i^+ - f_i^*}$; $d_i = -\frac{f_i^*}{f_i^+ - f_i^*}$;

$$f^+ = f_i(x^+) = \max_{x \in D} f_i(x); \quad f_i^* = f_i(x^*) = \min_{x \in D} f_i(x).$$

Тогда $\bar{f}_i(x) = \psi(f_i)$ оказывается приведенными к интервалу $[0, 1]$. Поскольку $f_i(x) = 0$ в точке x^* , а $f_i(x) = 1$ в точке x^+ (что соответствует полной потере оптимальности), то промежуточные значения $0 < \bar{f}_i(x) < 1$ характеризуют степень удаления точки x от точки минимума x^* по i -му частному критерию. Для безразмерных критериев рассматриваем функцию полезности (функция полезности может быть и другого вида) $\psi_i(\bar{f}_i(x)) = 1 - f_i(x)$, $i = 1, s$, которая численно характеризует предпочтение i -го частного критерия (по мере уменьшения $\bar{f}_i(x)$ полезность не может ухудшиться).

Какова величина важности значений векторного критерия в различных точках? Для сравнения значений критерия используют скалярную функцию $\Phi(f)$.

Будем считать, что $\Phi(f^k) \leq \Phi(f^l)$ тогда и только тогда, когда $f^k \succ f^l$ (т. е. f^k лучше f^l). Функцию $\Phi(f)$ называют обобщенным критерием оптимальности.

Упорядочив значения критерия в области D_f и выводя функцию $\Phi(f)$, можно решать задачу многопараметрической оптимизации $\Phi(f(x)) \rightarrow \min_{x \in D}$, а именно: найти эффективную точку $\bar{x} \in D$, которая является рациональным решением (в смысле минимума критерия $\Phi(f(x))$).

Процедура построения критерия $\Phi(f)$ называется свертыванием векторного критерия f . Удачный выбор вида функции $\Phi(f)$ определяется содержательной постановкой задачи проектирования, информацией о важности частных критериев.

Рассмотрим следующую функцию:

$$\Phi(f) = \max_{\lambda \in D_\lambda} \left\{ \sum_{i=1}^s \lambda_i f_i(x) + \lambda_o \right\},$$

где f_i – нормализованные частные критерии; D_λ – область возможных значений весовых коэффициентов λ_i , $i = \overline{0, s}$.

Частные случаи операции свертывания функции, где

$$D_\lambda = \{\lambda \mid \lambda_i \geq 0\}, \quad \Phi(f) = \sum_{i=1}^s \lambda_i f_i(x) + \lambda_o,$$

т. е. свертывание здесь – просто сумма с неизвестными коэффициентами

$$D_\lambda = \left\{ \lambda \mid \sum_{i=1}^s \lambda_i / \rho_i = 1; \lambda \geq 0; i = \overline{1, s}, \lambda_o = 0 \right\},$$

тогда $\Phi(f) = \max(\rho_i f_i(x))$,

где $\rho_i > 0$ известные весовые коэффициенты.

Если можем задаться значениями частных критериев f_i^o , которые позволяют считать их удовлетворительными, т. е. $f_i(x) \leq f_i^o$ (f_i^o критерийные ограничения), тогда в области

$$D_\lambda = \left\{ \lambda \mid \lambda_o = -1 + \sum_{i=1}^s \lambda_i f_i^o; \lambda_i \geq 0, i = \overline{1, s} \right\}$$

свертка сводится к разбиению результатов на удовлетворительные и неудовлетворительные (где нарушается хотя бы одно из неравенств $f_i(x) \leq f_i^o$).

При $\lambda = 0$ получаем аддитивный критерий

$$\Phi(f) = \sum_{i=1}^s \lambda_i g_i(x),$$

где λ_i – весовые коэффициенты определяют важность i -го частного критерия;

f_i – нормализованы; $\lambda_i \geq 0$, причем $\sum_{i=1}^s \lambda_i = 1$.

Этот критерий может быть использован, если частные критерии f_i взвешены числами λ_i и допускают количественное сравнение в одной размерности (или они уже безразмерные). Справедливо следующее утверждение.

Если все частные критерии $f_i(x)$ – выпуклые функции, то для любой эффективной точки $\mathbf{x} \in D$ (D – выпуклое множество) существует такой вектор $\lambda (\lambda_i > 0, i = \overline{1, s}; \sum_{i=1}^s \lambda_i = 1)$, что оптимальное решение \mathbf{x}^* зада-

чи $\left\{ \sum_{i=1}^s \lambda_i f_i(x) \right\} \rightarrow \min_{x \in D}$ достигается в эффективной точке x^* . При задан-

ных $\lambda_i (\lambda_i > 0; \sum_{i=1}^s \lambda_i = 1)$ оптимальное решение x^* является эффективной точкой. Обычно вводят ограничения (пороги) на неудовлетворительные критерии $f_k(x) \leq f_k^0, k = \overline{1, p} < s$ (чтобы не оказалось, что некоторые критерии компенсируются за счет остальных), например уменьшение одного критерия вплоть до нуля может быть покрыто возрастанием другого. Очевидно, это существенный недостаток аддитивного критерия.

Пример (“автомат”)

Масса заряженного автомата не более 6 кг. Критерии, которые требуется максимизировать.

1. V – скорость выбрасывания пуль.
2. N – число патронов в магазине.

И тот и другой критерии связаны с массой, на которую наложено ограничение.

Известно, что $V = 140\sqrt{l}$, м/с, где l – длина ствола, м.

Задано : масса ствола (погонная) $m_1 = 1$ кг/м; масса корпуса $m_2 = 2$ кг; масса патрона $m_3 = 0,07$ кг.

Уравнение баланса масс

$$1 \cdot l + 0,07N + 2 = 6. \quad (8.2)$$

Пусть оба критерия одинаково важны, поэтому после нормирования можно искать оптимальное решение с помощью обобщенного аддитивного критерия. Для нормирования надо найти N_{\max} и V_{\max} .

Пусть известно, что $V_{\min} = 100$ м/с, тогда из уравнения баланса масс имеем $N = 50$. Пусть мало патронов ($N \rightarrow 0$) и из (8.2) $l_{\max} = 4$ м, тогда

$$V_{\max} = 140\sqrt{l_{\max}} = 280 \text{ м/с. Частные критерии } f_1 = \frac{N}{50}; \quad f_2 = \frac{V}{280};$$

$$f = f_1 + f_2 = \frac{N}{50} + \frac{V}{280}.$$

Используя метод множителей Лагранжа с помощью функции $L = f + \lambda(4 - l - 0,07N)$, можно найти f_{\max} и эффективную точку: $V^* = 124$ м/с, $N^* = 46$; $l^* = 0,76$ м (рис. 22, а).

Использование аддитивного критерия оптимальности $\Phi(f) = \sum_{i=1}^s f_i$

для сравнения между собой по важности любой пары векторных критериев f^k и f^l , принадлежащих области компромиссов D_k и являющихся противоречивыми, т. е. $f_i^k \leq f_i^l, i \in I_1; f_j^k \geq f_j^l, j \in I_2$ (часть частных критериев в точке k превосходят их в точке l и наоборот), позволяет сформулировать принцип справедливого компромисса, реализующего метод абсолютной уступки $f^k \geq f^l$, если выполняется

$$\sum_{j \in I_2} (f_j^k - f_j^l) < \sum_{i \in I_1} f_i^k - f_i^l, \text{ откуда следует, что если векторный критерий}$$

f^k предпочтительнее критерия f^l , тогда суммарное абсолютное уменьшение по одному или нескольким частным критериям должно превосходить суммарное абсолютное увеличение по остальным частным критериям (это условие предпочтительности f^k по сравнению с f^l).

Другая форма обобщенного критерия $\Phi(f)$ для однородных частных критериев $f_i(x)$

$$\Phi_p(f) = - \left(\frac{1}{s} \sum_{i=1}^s f_i^p(x) \right)^{1/p},$$

где $-\infty \leq p \leq +\infty$.

При любом p оптимальное решение x^* задачи $\Phi_p(x) \rightarrow \min_{x \in D}$ является эффективной точкой $\bar{x} \in D$. Изменяя p , получаем частные случаи

$$p = 1, \lambda_i = \frac{1}{s} \text{ — равноценные частные критерии;}$$

$$\Phi_1(f) = \frac{1}{s} \sum_{i=1}^s f_i(x);$$

$p = 0$, логарифмируя и переходя к пределу при $p \rightarrow 0$;

$$\Phi_0(f) = \left[\prod_{i=1}^s f_i(x) \right]^{1/s} \text{ — средний геометрический критерий.}$$

Если $p \rightarrow \infty$, то

$$\Phi_\infty(f) = \max_{1 \leq i \leq s} f_i(x),$$

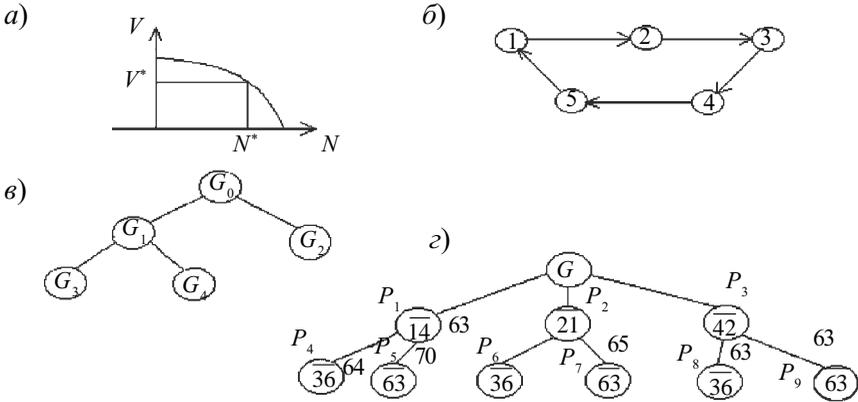


Рис. 22

т. е. обобщенный критерий оптимальности сводится к наибольшему из частных критериев.

В общем случае для любых функций $f_i(x)$, $i = \overline{1, s}$, и любой структуре области допустимых решений D , если $x \in D$ является эффективной точкой (причем все $f_i(x) > 0$), то существует вектор λ ($\lambda_i > 0$, $i = 1, \sum_{i=1}^s \lambda_i = 1$)

такой, что оптимальное решение задачи $\max_{1 \leq i \leq s} (\lambda_i f_i) \rightarrow \min_{x \in D}$ – минимаксный критерий (принцип гарантированного результата) достигается в эффективной точке x . Следует упомянуть также мультипликативный критерий $\Phi(x) = \prod_{i=1}^n f_i(x)$ (если все частные критерии имеют одинаковую важность), иначе надо ввести весовые коэффициенты λ_i

$$\Phi(x) = \prod_{i=1}^n f_i^{\lambda_i}(x).$$

Заметим, что здесь не требуется нормировка частных критериев, однако, возможна компенсация недостаточной величины одного критерия избыточной величиной другого.

Упражнение (пример “автомат”)

Решить рассмотренный ранее пример с обобщенным критерием вида $f=VN$. Функция Лагранжа $L=(V,N,\lambda) = VN+\lambda N+\lambda(4-1l-0,07N)$ (выражая L через V , найти V^*, N^*, l^*).

Выбор критерия

Аддитивный критерий выбирают тогда, когда существенные значения имеют абсолютные значения критериев при выбранном векторе x . Если существенную роль играют изменения абсолютных значений частных критериев при вариации вектора x , то используют мультипликативный критерий. Если требуется достигнуть, например, равенства нормированных значений конфликтных критериев, то используют минимаксный критерий (подтягиваем худший).

8.3. Пример использования минимаксной (максиминной) свертки векторного критерия

Рассмотрим задачу максимизации выхода годных изделий (уменьшение брака) в условиях статистического разброса параметров. Обычно предполагается, что в ТЗ оговорены требования на параметры изделия, т. е. если $\varphi_i(x) \leq t_i$ (где φ_i – выходной параметр, $i = 1, m$) – это годное изделие. Но из-за случайных отклонений параметров при изготовлении требование ТЗ может быть нарушено. Таким образом, задача о том, чтобы так выбрать вектор x^* и отклонения в технологическом процессе, и при эксплуатации в наименьшем числе случаев приводили к нарушению требований ТЗ.

Потребуем, чтобы требования ТЗ выполнялись с запасами

$$\varphi_i(x) + \delta_i \leq t_i, \quad \delta_i > 0,$$

где δ_i характеризует рассеяние i -го выходного параметра в результате случайных отклонений (δ_i – процент от $|t_i|$), что эквивалентно

$$f_i(x) = \frac{t_i \bar{\varphi}_i(x)}{\delta_i} - 1 \geq 0$$

– запас работоспособности по i -му выходному параметру. Таким образом, имеем многокритериальную задачу

$$f_i(x) \rightarrow \max, \quad i = \overline{1, m},$$

$x \in D$

где D – область, где выполняются прямые ограничения на x .

Часто используют максиминную свертку векторного критерия $\Phi(f) = \min_{i=1,m} f_i(x) \rightarrow \max_{x \in D}$ (т. е. подтягиваем к максимуму наихудший).

Очевидно, $\Phi(f)$ – негладкий функционал, поэтому для упрощения решения задачи его лучше сгладить. Задача $f_i \rightarrow \min$ эквивалентна задаче $e^{-f_i} \rightarrow \max$.

Учитывая, что $\operatorname{argmin} f_i \sim \operatorname{argmax} e^{-f_i}$, исходная задача эквивалентна задаче $\max_i e^{-f_i} \rightarrow \min_{x \in D}$ (максиминная задача стала минимаксной). Применяя среднестепенную свертку, имеем окончательный критерий (гладкий)

$$\Phi(x) = \sum_{i=1}^m e^{-\nu f_i(x)} \rightarrow \min_{x \in D}, \nu = 1, 2, \dots$$

8.4. Метод главного критерия

Предположим, что частные критерии качественно упорядочены в порядке убывания важности, т. е. $f_1 \succ f_2 \succ \dots \succ f_s$.

Одним из способов свертывания частных критериев, упорядоченных по важности, является метод выделения главного критерия. Идея метода состоит в минимизации главного критерия $f_1(x)$ при условии, что значения других критериев $f_i(x), i = 2, s$, не превышают “пороговых” значений $f_i^0(x)$. Приходим к решению следующей задачи:

$$f_1(x) \rightarrow \min_{x \in D \cap D_0},$$

где $D_0 = \{x \mid f_i(x) \leq f_i^0, i = \overline{2, s}\}$.

Если после выделения главного критерия предпочтения между $f_i(x), i = \overline{2, s}$,

можно задать с помощью весовых коэффициентов $\rho \left(\rho_i > 0, \sum_{i=2}^s \rho_i = 1 \right)$, то пороговые значения f_i^0 выбираются из соотношений

$$\rho_i \left(\frac{f_i^0 - f_i^*}{f_i^+ - f_i^*} \right) = k_0, i = \overline{2, s},$$

где $0 < k_0 \leq 1$; $f_i^+ = \max_{x \in D} f_i(x)$; $f_i^* = \min_{x \in D} f_i(x)$.

В этом случае применяют следующий алгоритм.

Задать k_0 и определить по формуле

$$f_i^0 = k_0(f_i^+ - f_i^*)/\rho_i + f_i^*, \quad i = \overline{2, s}.$$

Решить задачу

$$f_1(x) \rightarrow \min_{x \in D \cap D_0},$$

где $D_0 = \{x \mid f_i(x) \leq f_i^0, \quad i = \overline{2, s}\}$.

Если пороги таковы, что $D \cap D_0 = \emptyset$, т. е. нет точек пересечения множеств D и D_0 , то увеличить число k_0 : $k_0 = \beta k_0$, $\beta > 1$, перейти на 1. Иначе, найдена эффективная точка $\bar{x} \in D \cap D_0$.

Трудность: надо знать f^+ , f^* (оценки пределов изменения частных критериев).

8.5. Метод последовательных уступок

Этот метод позволяет учитывать интересы менее важных критериев, так как за счет незначительного увеличения более важных критериев, можно допустить уменьшение менее важных (смысл уступки).

Идея метода состоит в том, что на k -м шаге последовательной минимизации частных критериев (всего s шагов) вводится уступка Δf_{k-1} на допустимое отклонение $(k-1)$ -го критерия (более важного) от его минимального значения, полученного на предыдущем шаге

$$f_1(x_1^*) = \min_{x \in D} f_1(x)$$

...

$$f_k(x_k^*) = \min_{x \in D_k} f_k(x),$$

где $D_k = D \cap D_{k-1}$; $D_{k-1} = \{x \mid f_j(x) \leq f_j(x_j^*) + \Delta f_j, \quad j = \overline{1, k-1}\}$.

Область D_k зависит от D_{k-1} , а последняя от выбора уступок на всех шагах вплоть до $(k-1)$ -го (т. е. допускаем увеличение более важных).

Введение уступки Δf_{k-1} по $(k-1)$ -му (наиболее важному) частному критерию позволяет улучшить значение k -го (менее важного) частного критерия. При этом видно, ценой каких уступок по $(k-1)$ -му критерию приобретается тот или иной выигрыш.

8.6. Способы назначения весовых коэффициентов важности для частных критериев оптимальности

Если весовые коэффициенты λ_i не могут быть назначены из физической сущности задачи, существуют следующие способы их назначения.

1. Экспертные оценки.

Алгоритм индивидуальной экспертизы

Шаг 1: упорядочить (качественно) частные критерии (упорядочиванием индексов) $f_1 \succ f_2 \succ \dots \succ f_s$.

Шаг 2: критерию f_i ставится в соответствие оценка μ_i .

Оценкам назначаются некоторые числа относительно важности критериев, начиная с μ_s : $\mu_{i-1} \geq \mu_i$, $i = s, s-1, \dots, 2$.

Рассмотрим варианты логического выбора (табл. 8.1). Просматривая столбцы с 1-го по $(s-2)$ -й сверху вниз, в таблице эксперт фиксирует свои суждения о предпочтениях между левой и правой частью отношений (x и y).

Таблица 8.1

1		2		...	s-2	
x	y	x	y	...	x	y
$f_1 ?$	$f_2 + f_3 + \dots + f_s$	$f_2 ?$	$f_3 + \dots + f_s$...	$f_{s-2} ?$	$f_{s-2} + f_s$
$f_2 ?$	$f_2 + f_3 + \dots + f_{s-1}$
...
...
...
$f_i ?$	$f_2 + f_3$...	$f_3 + f_4$

Шаг 3: отношение ? надо заменить на $>$, $<$, \sim .

Шаг 4: в отношения, начиная с $(s-2)$ -го – столбца, подставляются оценки μ_i , $i = \overline{1, s}$. Столбцы просматриваются снизу вверх. Если суммарная величина оценок правой части не соответствует значению левой части в смысле выполнения одного из неравенств

$$\mu(x) > \sum_i \mu_i(y), \quad x > y; \quad \mu(x) < \sum_i \mu_i(y), \quad x < y;$$

$\mu(x) = \sum_i \mu_i(y)$, $x \sim y$, то левая оценка корректируется в минимально

возможной степени так, чтобы неравенства соответствовали решениям эксперта, поставленным в табл. 8.1.

При проверке каждого последующего отношения используют уже скорректированные оценки μ'_i .

Шаг 5: по уточненным значениям $\mu'_i, i = \overline{1, s}$ вычисляются весовые

коэффициенты $\lambda_i = \mu'_i / \sum_{k=1}^s \mu'_k, i = \overline{1, s}$.

Упражнение

Пусть есть пять частных критериев, расположенных в произвольном порядке: стоимость S , вес P , объем V , быстроедействие T , надежность N .

Упорядочим f_i

$$f_1 = N, f_2 = T, f_3 = S, f_4 = P, f_5 = V,$$

где $f_1 > f_2 > \dots > f_5$.

Примем $\mu_5 = 1; \mu_4 = 1,5; \mu_3 = 2; \mu_2 = 4; \mu_1 = 7$.

Составить таблицу вариантов и определить λ_i .

Если доступна информация об экстремальных значениях частных критериев оптимальности, т. е. $f_i^* = f_i(x_i^*) = \min_{x \in D} f_i(x), f_i^+ = \max_{x \in D} f_i(x),$

$i = \overline{1, s}$. Это помогает назначить значения весов λ .

Вычислим коэффициенты относительного разброса δ_i

$$\delta_i = (f_i^+ - f_i^*) / f_i^+ = 1 - \left| \frac{f_i^*}{f_i^+} \right|.$$

Они определяют максимально возможное изменение i -го критерия при изменении вектора x в области D . Веса λ_i получают наибольшие значения для тех частных критериев, относительный разброс которых в области D наиболее значителен

$$\lambda_k = \delta_i / \sum_{k=1}^s \delta_k, i = \overline{1, s}.$$

Заметим, что при $f_i^+ = f_i^* \left(f_i(x) = \text{const} \right)_{x \in D}$ имеем $\lambda_i = 0$, а при $f_i^+ \gg f_i^*, \delta_i \rightarrow 1$ и λ_i велико.

8.7. Функции “полезности”

Представление обобщенного критерия в виде суммы частных критериев или в другом искусственном виде носит субъективный характер. На самом деле должна существовать объективная зависимость между частными критериями, которую, к сожалению, трудно найти, т. е. $\psi = \psi(f_1, \dots, f_n)$. Один из путей основывается на теории полезности технических систем.

Полезностью называют некоторую количественную характеристику степени выполнения системой своего функционального назначения.

Например, полезность планетохода определяется величиной обследуемой площади поверхности планеты, достоверностью полученной информации, надежностью (при низкой надежности полезность может свестись к нулю).

Метод варьирования для построения функции полезности

Известно, что если уменьшение некоторого частного критерия f_i до величины f_i^* (минимальные значения) обращает полезность системы в нуль независимо от величины других частных критериев, то данный частный критерий входит в аналитическое выражение для полезности в виде множителя в положительной степени, т. е., если при $f_i = f_i^*$ имеем $\psi(f_1, \dots, f_i^*, \dots, f_n) = 0$, то при $f_i > f_i^*$ функция полезности принимает вид

$$\psi(f) = (f_i^{\beta_i} - f_i^{*\beta_i})\Phi(f_1, \dots, f_n),$$

где $\Phi(\dots) > 0$, $\beta_i > 0$.

Заметим, что если, например, остальные (кроме i -го) критерии равноценны, то $\Phi(\dots)$ – аддитивный обобщенный критерий (например, в ситуации, когда полезная функция обеспечивается несколькими подсистемами одного назначения).

Заметим, что не всегда можно указать такое f_i^* , при котором функция полезности обращается в нуль.

Пример (“полезность САПР”)

Пусть при ручном проектировании проект выполняется N_p числом специалистов. После внедрения САПР та же работа может быть сделана N_a специалистами. При $N_a = N_p$ функция полезности обращается в нуль, следовательно, при $N_a < N_p$ можем записать $\Phi(\dots) = (N_p - N_a)\Phi(\dots)$, но абсолютная разность $N_p - N_a$ плоха, так как если $N_p = 1000$, а $\Delta N = 20$, имеем неощутимый выигрыш по сравнению с $N_p = 40$. По этой причине лучше функция

$$\psi(\dots) = (\Delta N / N_p) \Phi(\dots) = (1 - N_a / N_p) \Phi(\dots).$$

Учтем другой частный критерий. Пусть с помощью САПР могут быть выполнены только q_a блоков проектируемого изделия, а $q_p = q - q_a$ блоков выполнить автоматизированно нельзя, тогда

$$\Delta N = N_p - [N_a(q_a / q) + N_p(q_p / q)] \text{ и } \psi(\dots) = (1 - N_a / N_p)(q_a / q) \Phi(\dots).$$

Если $q_a / q = 0$, то функция $\psi(\dots)$ обращается в нуль. Иначе говоря, чем меньшая доля блоков может быть спроектирована с помощью САПР, тем ниже ее полезность.

9. ДИНАМИЧЕСКОЕ ПРОГРАММИРОВАНИЕ

Эффективность численного решения задачи оптимизации во многом определяется ее размерностью. Ранее рассматривались методы, в которых компоненты вектора x выбирались одновременно, что при высокой размерности вектора x может сильно затруднить решение.

В основе динамического программирования лежит идея постепенной пошаговой оптимизации (если можно разбить процесс на шаги). Как правило, оптимизировать один шаг легче оптимизации всего процесса, поэтому выгоднее много раз решать простую задачу, чем один раз сложную.

Метод имеет сходство с методом ветвей и границ (методы направленного перебора вариантов). Рассмотрим частную задачу (сепарабельная функция)

$$f(x) = \sum_{i=1}^n \varphi_i(x_i) \rightarrow \max_x;$$

$$\sum_{i=1}^n a_i x_i \leq b, \quad x_i \geq 0, \quad a_i > 0.$$

Разобьем процесс оптимизации на шаги так, чтобы на каждом шаге решалась задача одномерной оптимизации, т. е. принималось решение относительно оптимального значения только одной переменной.

$$\max_x f(x) = \max_{x_1} [\varphi_1(x_1) + f_2(b - a_1 x_1)];$$

$\varphi_1(x_1)$ – первая подзадача;

$f_2(b - a_1 x_1)$ – вторая подзадача;

$$f_2(b_2) = \max \sum_{i=1}^n \varphi_i(x_i). \quad (9.1)$$

Новая переменная b_2 характеризует состояние процесса перед вторым шагом

$$\sum_{i=1}^n a_i x_i \leq b_2;$$

где $b_2 = b - a_1 x_1$.

Оптимальное значение первого слагаемого – это выигрыш на первом шаге, оптимальное значение второго слагаемого – выигрыш на всех последующих шагах. Выбранное значение x_1 называется шаговым управлением. Для выбора оптимального значения x_1 нужно.

1. Определить все возможные целочисленные значения x_1 , исходя из того, что b_2 в (9.1) должно быть неотрицательным ($a_i x_i \geq 0$), следовательно, $b_2 - a_1 x_1 \geq 0$.

Для общности, обозначим $b = b_1$, тогда $0 \leq x_1 \leq \frac{b_1}{a_1}$ – диапазон изменения x_1 .

2. Для каждого значения x_1 (из диапазона) определим численное значение b_2 , а именно

$$b_2 = b_1 - a_1 x_1.$$

Далее для каждого x_1 вычисляем величину первого, второго слагаемых и их сумму. Это делается путем формирования таблиц. Затем рассматривается значение суммы для каждого x_1 и находится оптимальное x_1 . Строится рекуррентная схема вычислений при разных правых частях b_i (различных состояниях)

$$f_i(b_i) = \max[\varphi_i(x_i) + f_{i+1}(b_i - a_i x_i)];$$

$$f_{n+1}(b_{n+1}) = 0, \quad i = \overline{n, 1}.$$

Заметим, что функции f_i могут быть многоэкстремальными (в общем случае). Более того, возможен неоднозначный выбор x_i .

Особенность n -го шага: второе слагаемое будет нулевым. Поэтому оптимизацию выполняют с n -го шага. При $i = n$ второе слагаемое нулевое, а ограничение остается $a_n x_n \leq b_n$.

Так как b_n неизвестно, оптимизация последнего шага выполняется для всех возможных состояний b_n . Для каждого конкретного значения b_n находим x_n – множество условно оптимальных значений. Теперь $i = n-1$ и определяем $x_{n-1}(b_{n-1})$.

Выполняем эту процедуру, пока не дойдем до первого шага, особенность которого в том, что не надо делать предположений о состоянии процесса b_1 , так как $b_1 = b$ и перед первым шагом известно.

Процесс, состоящий из k оставшихся шагов с индексами $k, k-1, \dots, 1$, для которого вектор фазовых переменных b_{k+1} , является начальным состоянием, называется k -шаговым конфинальным подпроцессом. Этот процесс описывает влияние вектора управляемых параметров только на “будущее” (последующие состояния b_k, \dots, b_1) в зависимости от “настоящего” (текущего состояния b_{k+1}) и не зависит от “прошлого” (предыдущих состояний b_{n+1}, \dots, b_{k+2}) (см. рекуррентную схему).

Строится оптимальная стратегия распределения

$$(x_1^*, \dots, x_n^*);$$

$$x_1 = x_1^*; f^* = f_1(b_1),$$

обеспечивающая получение значения $f_1(b_1)$. Выполняем вычисления в прямом направлении. Можем найти b_2 перед вторым шагом.

Ошибка в выборе x_k^* , сделанная на k -м шаге, не может быть исправлена на последующих шагах $k+1, \dots, n$

$$b_2^* = b_1 - a_1 x_1^* \rightarrow x_2^* = x_2(b_2^*);$$

$$b_n^* = b_{n-1} - a_{n-1} x_{n-1}^* \rightarrow x_n^* = x_n(b_n^*).$$

Для применения динамического программирования в задачах оптимизации надо решить три вопроса:

- расчленить процесс оптимизации на шаги;
- выяснить какие параметры могут однозначно характеризовать состояние системы (процесс оптимизации) перед i -м шагом;
- определить все возможные решения на i -м шаге.

Самое трудное – выбрать параметры, характеризующие состояния. Надо характеризовать состояние системы такими параметрами, чтобы они позволяли:

- осуществлять переход от одного этапа к другому;
- условное оптимальное решение на k -м шаге должно зависеть только от текущего состояния и не должно зависеть от предыстории процесса (такие процессы называются марковскими).

При выполнении этих условий можно найти решение на основе принципа оптимальности Беллмана.

Каково бы ни было состояние системы перед очередным шагом, надо выбирать решение на этом шаге так, чтобы выигрыш на данном шаге плюс оптимальный выигрыш на всех последующих шагах был оптимальным.

Рассмотренное рекуррентное состояние основано на этом принципе оптимальности. На языке графов: любой подпуть оптимального пути является также оптимальным путем.

Пример

Рассмотрим задачу о загрузке (о ранце), т. е. определение оптимального числа загружаемых предметов каждого типа (табл. 9.1).

Таблица 9.1

	x_1	x_2	x_3
a_i	2	3	1
c_i	65	80	30

Здесь a_i – вес предмета; c_i – полезность предмета; x_i – число загружаемых предметов i -го типа (может быть больше 1 или равно 0); грузоподъемность ранца $b = 5$; $n = 3$ – число типов предметов. Таким образом, имеем задачу

$$f(x) = c_1x_1 + c_2x_2 + c_3x_3 \rightarrow \max;$$

$$a_1x_1 + a_2x_2 + a_3x_3 \leq b.$$

Решение рассматривается как n -шаговый процесс. На каждом шаге решается вопрос о числе загружаемых предметов i -го типа (одного!). За параметр, характеризующий состояние системы перед i -м шагом, берут свободный вес, которым можно распорядиться на этом и последующих шагах.

Сначала оптимизируется третий шаг. Решаем вопрос: сколько выбрать предметов третьего типа, чтобы максимизировать $f_3(x_3) = c_3x_3$?

Вес перед этим шагом нам известен, поэтому делаем все возможные предположения относительно b_3 , т. е. 0, 1, 2, 3, 4, 5.

Составим таблицу для всех шагов. Сначала определим условно оп-

тимальные значения $i = 3$; $0 \leq x_3 \leq \frac{b_3}{a_3} = 5$.

Используя рекуррентную формулу, решаем одномерную задачу оптимизации

$$f_3(b_3) = \max_{0 \leq x \leq 5} \varphi_3(x_3) = \max(\varphi_3(0), \varphi_3(1), \dots, \varphi_3(5)),$$

так как $f_4(b_4) = 0$.

Решение совокупности одномерных задач оптимальности занесем в первую строку табл. 9.2 – 9.4:

$$i=2; b_2=b_1 - a_1x_1 = 5 - 2x_1,$$

учитывая, что

$$0 \leq x_1 \leq \frac{b_1}{a_1} = \frac{5}{2} = 2, \text{ т. е. } x_1 = (0,1,2),$$

тогда $b_2 = (5,3,1)$

$$\begin{aligned} f_2(b_2) &= \max_{0 \leq x_2 \leq \frac{b_2}{a_2}} (c_2x_2 + f_3(b_2 - a_2x_2)) = \\ &= \max(c_2 \cdot 0 + f_3(b_2), c_2 \cdot 1 + f_3(b_2 - a_2 \cdot 1)), \end{aligned}$$

так как $x_2 = (0,1)$.

Затем решается одномерная задача оптимизации. Для фиксированных значений $b_2 = 5; 3; 1$ переменная x_2 имеет значения 0; 1, из которых оптимальным является $x_2 = 0$ (функция достигает максимума).

Таблица 9.2

b	1	3	5
$x_2=0$	30	90	150
$x_2=1$?	80	140

Таблица 9.3

N	b_i	0	1	2	3	4	5
1	$x_3(b_3)$	0	<u>1</u>	2	3	4	5
	$f_3(b_3)$	0	<u>30</u>	60	90	120	150
2	$x_2(b_2)$		<u>0</u>		0		0
	$f_2(b_2)$		<u>30</u>		90		150
3	$x_1(b_1)$						<u>2</u>
	$f_1(b_1)$						160

Свободный вес перед первым шагом равен 5, поэтому его пишем в пятый столбец. Функцию f_1 можно не табулировать. Элементарный счет для третьей строки дает

$$f_1(b_1) = \max_{x_1} (c_1x_1 + f_2(b_1 - a_1x_1)) = \max_{x_1} (65x_1 + f_2(b_1 - 2x_1)).$$

Таблица 9.4

x_1	0	1	2
$65x_1$	0	65	130
b_2	5	3	1
$f_2(b_2)$	150	90	30
0	150	90	160

Определение оптимальных значений

$$b_2^* = b_1 - a_1 x_1^* = 5 - 2 \cdot 2 = 1;$$

$$x_2^* = x_2(b_2^*) = 0;$$

$$b_3^* = b_2^* - a_2 x_2^* = b_2^* = 1;$$

$$x_3^* = x_3(b_3^*) = 1.$$

Проверка: $f^* = 65 \cdot 2 + 80 \cdot 0 + 30 \cdot 1 = 160$ (т. е. совпадает с $f_1(x_1^*)$).

10. ДИСКРЕТНОЕ ПРОГРАММИРОВАНИЕ

В данной главе рассмотрены методы и алгоритмы, относящиеся к классу задач, в которых аргумент целевой функции может принимать дискретные значения на дискретно заданном множестве. Приведены задачи о назначениях, о коммивояжере и т. д.

10.1. Задача о назначениях

Пусть имеется n кандидатов на выполнение n работ. Назначение i -го кандидата на j -ю работу связано с расходами. Требуется назначить по одному кандидату на каждую работу так, чтобы суммарная эффективность была максимальна (или сумма расходов минимальна).

Рассмотрим две матрицы: C – матрица эффективностей ($n \times n$), X – матрица назначений ($n \times n$).

Так как каждый работник может выполнить только одну работу и каждый вид работ выполняется одним работником, то в строке и в столбце матрицы X может стоять только по одной единице.

Итак, имеем следующую задачу линейного программирования:

$$f(x) = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \max - (\text{суммарная эффективность назначений})$$

при

$$\sum_{j=1}^n x_{ij} = 1, \quad i = \overline{1, n};$$

$$\sum_{i=1}^n x_{ij} = 1, \quad j = \overline{1, n};$$

$$x \in \{0, 1\}.$$

Иначе говоря, необходимо определить позиции ненулевых элементов матрицы X .

Введем понятие эквивалентных матриц эффективности. Если к элементам любой i -й строки матрицы

$$\mathbf{D} = [d_{ij}]$$

прибавить произвольное число α_i , а к элементам любого j -го столбца прибавить число β_j и, то новая матрица $\mathbf{C} = [c_{ij}]$, где $c_{ij} = \alpha_i + \beta_j + d_{ij}$ называется эквивалентной матрице \mathbf{D} .

Теорема

Множество оптимальных назначений двух задач выбора с эквивалентными матрицами совпадают (т. е. отличия будут состоять только в значениях целевых функций, а не в \mathbf{x}^*).

Рассмотрим переход от задачи на максимум к задаче на минимум. Этот переход осуществляется в результате преобразования матрицы \mathbf{C} . Пусть в задаче на максимум имеем следующую матрицу \mathbf{C} :

$$\mathbf{C} = \begin{vmatrix} 3 & 4 & 2 & 2 & 1 \\ 4 & 5 & 3 & 1 & 3 \\ 4 & 3 & 1 & 1 & 1 \\ 3 & 1 & 2 & 2 & 2 \\ 1 & 3 & 1 & 2 & 1 \end{vmatrix}.$$

Воспользуемся теоремой для преобразования матрицы \mathbf{C} таким образом, чтобы она была неотрицательной и имелся хотя бы один нуль в каждом столбце и каждой строке. Нули должны быть в разных строках и столбцах, тогда эти нули независимые, и оптимальное решение есть позиция таких нулей (это глобальный минимум целевой функции).

На первом этапе в каждом столбце находим максимальный элемент и из него вычитаем каждый элемент данного столбца. Если после выполнения этой операции для всех столбцов в какой-либо строке не будет нуля, то надо в этой строке найти минимальный элемент и вычесть его из всех элементов данной строки. После указанных преобразований матрица \mathbf{C} имеет вид

$$\mathbf{C} = \begin{vmatrix} 1 & 1 & 1 & 0^* & 2 \\ 0 & 0 & 0 & 1 & 0^* \\ 0^* & 2 & 2 & 1 & 2 \\ 1 & 4 & 1 & 0 & 1 \\ 3 & 2 & 2 & 0 & 2 \end{vmatrix}.$$

10.2. Венгерский метод решения задачи о назначениях

Исходная задача преобразуется к задаче поиска минимума рассмотренным способом и затем строится максимально возможное число независимых нулей (для наискорейшего получения решения).

Далее на k -й итерации получается сумма независимых нулей, содержащая на один нуль больше. При этом увеличивается число независимых нулей, расположение которых может меняться. Это обеспечивается нахождением на k -й итерации новых нулей $0'$ (так называемых “дублеров”-претендентов на замену предыдущего независимого нуля). Замена идет по цепочкам вида

$$\begin{array}{c} 0' \rightarrow 0^* \\ \uparrow \\ 0^* \rightarrow 0' \\ \uparrow \\ 0' \end{array}$$

Конец итерации – нахождение нуля в строке, где ранее не было независимого нуля. Для его получения необходимы дополнительные преобразования матрицы C таким образом, чтобы все существующие раньше 0^* и $0'$ сохранялись. Для этого вводится система пометок некоторых строк и столбцов.

Примем соглашение вычитать из непомеченных строк и прибавлять к помеченным столбцам величину h – минимальный дважды непомеченный элемент матрицы C (он принадлежит одновременно непомеченной строке и непомеченному столбцу).

В результате выполнения такого преобразования:

- дважды непомеченные элементы уменьшаются на h ;
- дважды помеченные элементы увеличиваются на h ;
- остальные сохраняются без изменений.

Если 0^* и $0'$ будут однократно помечены, то они сохранят свое значение (например, если элемент принадлежит помеченному столбцу и непомеченной строке, он увеличится и уменьшится на h , т. е. не изменится), но появляется дополнительно один или несколько нулей.

Это эквивалентное преобразование, поэтому полученное оптимальное решение будет справедливо и для C .

Алгоритм

Шаг 0: начальное преобразование C , переход к задаче на минимум с образованием нулей, получение начальной (неполной) системы независимых нулей и пометка столбцов, содержащих независимые нули

$$\begin{array}{c}
 + \qquad \qquad + \quad + \\
 \left| \begin{array}{ccccc}
 1 & 1 & 1 & 0^* & 2 \\
 0 & 0 & 0 & 1 & 0^* \\
 0^* & 2 & 2 & 1 & 2 \\
 1 & 4 & 1 & 0 & 1 \\
 3 & 2 & 2 & 0 & 2
 \end{array} \right|,
 \end{array}$$

где 0 – нуль; 0' – дублер; $\bar{0}$ '' – дважды непомеченный нуль; 0* – независимый нуль.

Шаг 1: просмотр строк с целью поиска нуля в непомеченной строке, где ранее не было 0* или поиска “дублера” в строке, в которой есть 0*. Все эти нули должны быть дважды не помечены. При нахождении дублера (если есть возможность неоднозначного выбора дублера, можно выбирать любой, но один) помечается строка и снимается пометка у столбца, содержащего 0* данной строки

$$\begin{array}{c}
 + \qquad \qquad + \quad \oplus \\
 \left| \begin{array}{ccccc}
 1 & 1 & 1 & 0^* & 2 \\
 0 & 0' & 0 & 1 & 0^* \\
 0^* & 2 & 2 & 1 & 2 \\
 1 & 4 & 1 & 0 & 1 \\
 3 & 2 & 2 & 0 & 2
 \end{array} \right| \quad h = 1,
 \end{array}$$

где \oplus – снятие пометки.

Затем просмотр строк вновь начинается с первой строки.

Шаг 2: если все возможные “дублиры” найдены, а дважды непомеченный нуль в новой строке не обнаружен, то выполняется преобразование матрицы h и возврат на шаг 1.

После повторного выполнения шагов 1, 2:

$$\begin{array}{c}
 + \\
 + \left| \begin{array}{ccccc}
 1 & 0' & 0 & 0^* & 1 \\
 1 & 0' & 0 & 2 & 0^* \\
 0^* & 1 & 1 & 1 & 1 \\
 1 & 3 & 0^* & 0 & 0 \\
 3 & 1 & 1 & 0 & 1
 \end{array} \right|,
 \end{array}$$

Дважды непомеченный элемент уменьшен на h . Все пометки и дублиры убираются и начинается новая итерация. После третьей итерации

$$\begin{array}{c}
 + \\
 + \\
 \\
 +
 \end{array}
 \left| \begin{array}{ccccc}
 1 & 0' & 0 & 0^* & 1 \\
 1 & 0' & 0 & 2 & 0^* \\
 0^* & 1 & 1 & 1 & 1 \\
 1 & 3 & 0^* & 0' & 0 \\
 3 & 1 & 1 & \bar{0}'' & 1
 \end{array} \right|.$$

$$\oplus \quad \oplus \quad \oplus$$

Дублиры найдены, пометим соответствующие строки и снимем пометки со столбцов, обнаружим дважды непомеченные нули. Найдем дважды непомеченный нуль $\bar{0}''$.

Шаг 3: выполняется построение новой нерасширенной системы независимых нулей. Это построение выполняется цепочкой:

$$\bar{0}'' \rightarrow 0^* \rightarrow 0' \rightarrow \dots \rightarrow 0^* \rightarrow 0'.$$

Цепочка начинается с последнего найденного нуля в строке, где не было 0^* и заканчивается на дублире в столбце, в котором не было 0^* . Пометки всех строк снимаются, а пометки столбцов, содержащих 0^* , восстанавливаются.

Если найден $\bar{0}''$, но в столбце уже есть 0^* , то его 0^* убираем (вместо него вступает дублир). Вместо $\bar{0}''$ ставим 0^* .

Шаг 4: если система независимых нулей неполная, то идти на шаг 1, иначе – выход из алгоритма.

Здесь сразу получили полную систему независимых нулей, поэтому оптимальная матрица значений имеет вид:

$$\mathbf{X} = \left| \begin{array}{ccccc}
 0 & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1 \\
 1 & 0 & 0 & 0 & 0 \\
 0 & 0 & 1 & 0 & 0 \\
 0 & 0 & 0 & 1 & 0
 \end{array} \right|.$$

Оптимальное значение целевой функции:

$$f^* = c_{12} + c_{25} + c_{31} + c_{43} + c_{54}.$$

Одним из применений в САПР рассмотренной задачи о назначениях является задача размещения элементов на печатной плате. От размещения элементов зависит длина соединительных проводников, определяющая уровень помех и время распространения сигналов.

Рассмотрим следующую задачу.

Разместить компоненты e_1, \dots, e_n на множестве l_1, \dots, l_m ($m > n$) позиций монтажного пространства так, чтобы суммарная длина электрических соединений между компонентами была минимальна:

$$f(x) = \sum_{i=1}^{n-1} \sum_{j=i+1}^n \sum_{k=1}^m \sum_{j=k+1}^m x_{ik} x_{js} d_{ks} r_{ij};$$

$$\sum_{k=1}^n x_{ik}, i = \overline{1, n},$$

т. е. каждый элемент размещен только на одной позиции

$$\sum_{i=1}^n x_{ik} \leq 1, k = \overline{1, m},$$

т. е. на каждую позицию будет назначено не более одного элемента.

В выражении для минимизируемой функции d_{ks} – расстояние между позициями l_k, l_s (\mathbf{D} – матрица расстояний); r_{ij} – число связей между e_i и e_j (\mathbf{R} – матрица связей); $x_{ik} = 1$, если компонента e_i назначается на позицию l_k и 0 – в противном случае.

Это задача целочисленного дискретного программирования. Первое слагаемое имеет вид: $x_{11} x_{22} d_{12} r_{12}$, т. е. действительно первый элемент назначается на первую позицию, а второй на вторую.

10.3. Транспортная задача

Рассмотрим следующую задачу.

Есть магазины и склады или некоторые потребители и поставщики. Пусть, например, имеется два поставщика и три потребителя:



Необходимо определить матрицу перевозок $X=[x_{ij}]$ (количество товаров, перевозимых от i -го поставщика j -му потребителю). C – матрица стоимостей перевозок

	1	2	3
1	0,8	1,1	0,9
2	1,0	0,7	1,2

$\mathbf{a} = (12, 15)^T$, где \mathbf{a} – вектор поставок;

$\mathbf{b} = (8, 9, 10)$, \mathbf{b} – вектор потребления.

В данном примере оптимальная матрица перевозок

$$\mathbf{X}^* = \begin{vmatrix} 2 & 0 & 10 \\ 6 & 9 & 0 \end{vmatrix}.$$

Формальная запись

$$f(x) = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min \text{ (затраты на перевозку);}$$

$x \geq 0, \sum_{j=1}^n x_{ij} = a_i, i = \overline{1, m}$ – сумма элементов в i -й строке матрицы X должна быть равна объему товара, который есть у i -го поставщика (столько

можем перевезти)

$\sum_{i=1}^n x_{ij} = b_j, j = \overline{1, n}$ – сумма элементов в j -м столбце

должна быть равна объему потребления j -го потребителя

$$\sum_{j=1}^m a_j = \sum_{j=1}^n b_j. \tag{10.1}$$

Если объем товаров, имеющийся у всех поставщиков, равен объему товаров, который может быть потреблен, то задача называется закрытой.

Если поставок больше, то вводится фиктивный потребитель (расширение C на один столбец с нулевыми элементами).

Это задача линейного программирования со следующей особенностью: переменная входит в ограничения с коэффициентом единица и входит ровно в два ограничения.

Задача может быть решена венгерским методом, аналогичным алгоритму решения задачи о назначениях. При этом осуществляется процесс минимизации невязок

$$S_i = a_i - \sum_{j=1}^n x_{ij} \quad (\text{по строке});$$

$$S_j = b_j - \sum_{i=1}^m x_{ij} \quad (\text{по столбцу}).$$

Как только невязки станут нулевыми, т. е. выполнятся все ограничения, то найдено оптимальное решение.

В отличие от задачи о назначениях здесь объем перевозок может быть не единичным и в строке (столбце) может быть больше одного нулевого элемента.

10.4. Задача о коммивояжере

Построим граф, где каждой строке соответствует своя вершина (рис. 22, б), на основе следующей матрицы \mathbf{X}^* :

$$\mathbf{X}^* = \begin{vmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{vmatrix}.$$

Если дуги образуют единый замкнутый контур (цикл), то матрица \mathbf{X} называется матрицей циклической подстановки.

Введем понятие – гамильтонов контур. Это контур (цикл), включающий каждую вершину графа ровно один раз.

Гамильтонов путь – это разомкнутый путь, проходящий через вершины ровно один раз.

Задача

Коммивояжер должен посетить ряд городов по одному разу и возвратиться в исходный город. Надо выбрать такой маршрут, чтобы минимизировать путь.

Общее число маршрутов для N городов $(N-1)!/2$. С ростом N число маршрутов M растет по экспоненциальному закону

N	M
5	12
10	181440
50	*****

Такого типа задачи называют NP -полными. Они не решаются алгоритмами полиномиальной сложности. Чтобы уйти от полного перебора, предложен ряд различной вычислительной сложности.

Математически задача совпадает с формулировкой задачи о назначениях при условии, что матрица решений X является матрицей циклической подстановки (исходной информацией является матрица расстояний C , а результатом – матрица решений X)

$$\sum x_{ij} c_{ij} \rightarrow \min.$$

Примем, что $x_{ij} = 1$, если переезд совершается из i в j и $x_{ij} = 0$ в противном случае. При этом x_{ij} должны выбираться так, чтобы не попасть дважды в один из городов, в частности нельзя дважды попасть из i в i , т. е. $c_{ii} = \infty$.

Задача эквивалентна поиску минимального гамильтонова контура, с которой, в свою очередь, связана задача поиска минимального гамильтонова пути. Если задача на максимум, то преобразуем матрицу C

$$c'_{ij} = c_m - c_{ij},$$

$$c_m = \max_{i \neq j} c_{ij}.$$

Пример приложения задачи о коммивояжере

Выбор оптимальной последовательности обработки деталей

N деталей должны пройти последовательную обработку на двух станках (a и b). Времена обработки i -й детали на станках a и b соответственно a_i и b_i . Надо выбрать такую последовательность обработки, чтобы минимизировать время простоя второго станка. Понятно, что если быстрее успеваем обработать деталь на станке a , следовательно, этот станок меньше стоит. Тогда может быть выведено соотношение. Если $\min(a_i, b_i) \leq \min(a_j, b_j)$, то сначала обрабатывается i -я деталь, а затем j -я. В этом случае в матрице стоимостей $c_{ij} = 1$, а элемент $c_{ji} = 0$ (этот путь запрещен, разрыв) (табл. 10.1).

Таблица 10.1

Номер детали	a_i	b_i
1	10	5
2	7	14
3	5	3
4	10	3
5	13	8

На основе этой таблицы можем построить матрицу стоимости C

C	1	2	3	4	5
1	×	×	1	1	×
2	1	×	1	1	1
3	×	×	×	1	×
4	×	×	1	×	×
5	1	×	1	1	×

Например : $\min(10, 14) > \min(7, 5)$, тогда $c_{21} = 1$, $c_{12} = 0$.

Решением задачи о коммивояжере является минимальный гамильтонов путь (он проходит через все вершины по одному разу). В рассмотренной задаче решением является следующий путь: (2, 5, 1, 4, 3) – последовательность обработки деталей.

10.5. Метод ветвей и границ для решения задачи о коммивояжере

Метод имеет разные реализации, общей является только идея – разбиение (ветвление) множества допустимых решений G на дерево подмножеств – дерево решений. Процесс ветвления основан на построении оценки снизу (границы) для целевой функции на некотором множестве (подмножестве) допустимых решений (если задача на максимум, то оценка не снизу, а сверху).

Для каждой задачи с учетом ее специфики разрабатывается:

- правило ветвления;
- способ вычисления оценок;
- способы нахождения решений.

Процесс ветвления иллюстрирует рис. 22, в.

Разбиваем согласно правилу разбиения на непересекающиеся подмножества:

$$G_1 \cup G_2 = G_0,$$

$$G_1 \cap G_2 = \emptyset.$$

Для дальнейшего ветвления выбирается подмножество с наименьшей оценкой снизу (если задача на минимум), так как в нем с большей вероятностью содержится оптимальное решение.

Если же в процессе ветвления получаются подмножества с оценками большими, чем у подмножеств ранее признанных непересекающимися, происходит возврат к подмножеству с наименьшей оценкой.

Если получено подмножество, состоящее из одного решения и ему соответствует наименьшая оценка, то это решение является оптимальным. Если решение имеет значение целевой функции, отличающееся от наименьшего значения на величину ξ , то оно может быть принято за приближенное оптимальное решение. В худшем случае (неудачная реализация метода) – полный перебор.

Применение метода ветвей и границ к задаче о коммивояжере

Можно решение задачи о коммивояжере свести к многократному решению задачи о назначениях. Поскольку здесь тоже нужно выбирать по одному элементу из каждого столбца и каждой строки матрицы C , то они должны образовывать контур.

Метод заключается в отыскании гамильтонова контура с минимальным значением исходя из решения задачи о назначениях с минимальным значением. Это указывает оценку снизу длины любого гамильтонова контура.

Если решение исходной задачи дает гамильтонов контур (что определяется проверкой), это есть оптимальное решение. В данном случае это подмножество, содержащее один единственный маршрут.

Если же решение представляет собой совокупность частных контуров, то для продолжения решения выбирается один из таких контуров (для ветвления). Удобно выбирать контур с минимальным числом дуг.

Пусть выбран контур $x_1, x_2, \dots, x_k, x_1$. Тогда далее надо решить одну за другой k новых задач назначения, отличающихся тем, что в одной из них запрещается дуга x_1x_2 , во второй x_2x_3 и так далее по всем дугам контура, а решение этих задач задают оценки снизу длин соответствующих множеств гамильтоновых контуров.

Если в результате решения одной из таких задач удалось получить гамильтонов контур и его длина оказалась меньше или равной всем

ранее полученным оценкам снизу, то это и есть искомое решение задачи. В противном случае, продолжается построение дерева решений.

Для ветвления выбирается подмножество с наименьшей оценкой. В отличие от рассмотренных ранее ветвей и границ, подмножество решений разбивается не на две части, а на k – частей, $k \geq 2$ и меняется в течение процесса.

$$C = \begin{vmatrix} \infty & 27 & 43 & 16. & 30 & 26 \\ 7. & \infty & 16 & 1 & 30 & 30 \\ 20 & 13 & \infty & 35 & 5. & 0 \\ 21 & 16. & 25 & \infty & 18 & 18 \\ 12 & 46 & 27 & 48 & \infty & 5. \\ 23 & 5 & 5. & 9 & 5 & \infty \end{vmatrix}$$

$$X = \begin{vmatrix} & & & 1 & & \\ 1 & & & & & \\ & & & & 1 & \\ & & & & & \\ & 1 & & & & \\ & & & & & 1 \\ & & 1 & & & \end{vmatrix}$$

Один контур : (1, 4), (4, 5), (5, 3), (3, 6), (6, 2), (2, 1)
 $16 + 18 + 27 + 0 + 5 + 7 = 73.$

Два контура : (1, 4), (4, 2), (2, 1), (3, 5), (5, 6), (6, 3),

$$\left. \begin{array}{l} 16 + 16 + 7 = 39 \\ 5 + 5 + 5 = 15 \end{array} \right\} 39 + 15 = 54$$

– произвольный маршрут (оптимальный не может его превосходить).

Необходимо выбрать по одному элементу из каждой строки каждого столбца так, чтобы из индексов элементов матрицы можно было организовать цикл, т. е. матрица решений X должна быть матрицей циклической подстановки.

Это задача на минимум. Решив задачу для исходного полного решения, находим оценку снизу. Рассмотрим рис. 22, г.

Узел с пометкой $\overline{(i, j)}$ соответствует подмножеству “всех маршрутов”, не содержащих звено (i, j) , $P_0 \rightarrow 1421; 3563$ – контуры.

Проверим, является ли это решение гамильтоновым циклом? Нет не является (в каждом контуре по 3 дуги). Выполняем разбиение исходного множества на подмножестве для первого контура (1421). Для нахождения оценки снизу для получаемых подмножеств исключаем из исходной матрицы элемент (1, 4), на его место ∞ . Решаем задачу о назначении (расстановка нулей).

Для преобразованной матрицы не может быть получено гамильтонова контура меньше 63. Аналогично, для дуг (2, 1), (4, 2). Получаем $P_1 = P_2 = P_3$.

Результат решения задачи о назначениях. Два контура : 12451 ; 363
 $\Sigma = 58, \Sigma = 5$.

Таково решение задач о назначении с запрещенной дугой (1, 4).

По определению $\Sigma = 58 + 5 = 63$ – наименьшая сумма.

Решения не дали гамильтонова контура. Начинаем следующий этап ветвления. Разбиение по второму контуру. При разборе каждого следующего множества предыдущие возвращаются. Например, если рассмотреть 21, 36, 63, то разрешим дуги 14, 36, 63. Результат решения задачи в назначениях:

P_4 : 1241, 3563;

P_5 : 1243651 – гамильтонов контур(!), но может быть не минимальный?

$P_6 = P_4$;

$P_7 = 1423651$ – гамильтонов контур;

$P_8 = 1435621$ – гамильтонов контур, так как его длина равна 63, т. е. меньше или равно, чем все ранее полученные оценки снизу.

Если бы не нашли оптимального решения, то дальнейшее рассмотрение велось бы для подмножества с наименьшей оценкой $P_9 = P_8$.

11. ПРИНЦИПЫ ОРГАНИЗАЦИИ ДИАЛОГОВОЙ ПОДСИСТЕМЫ ПАРАМЕТРИЧЕСКОЙ ОПТИМИЗАЦИИ

Формально проектирование рассматривают как совокупность двух взаимосвязанных задач: структурной и параметрической оптимизации. Процесс синтеза структуры ПО в общем случае трудно поддается оптимизации и здесь не рассматривается. Другой класс задач, связанный с параметрическим синтезом, является предметом рассмотрения данной главы. Эффективность диалогового решения задач оптимизации определяется:

- выбором архитектуры ПО;
- организацией вычислительного процесса;
- организацией диалогового взаимодействия.

Прежде всего отметим, что в настоящее время отсутствуют универсальные методы минимизации, одинаково эффективные для широкого круга задач, поэтому возникает необходимость в программной реализации набора алгоритмов минимизации. В связи с этим стоит вопрос выбора структуры организации библиотеки алгоритмов.

11.1. Архитектура

Наиболее рациональным представляется путь построения библиотеки в виде многоуровневой структуры. Это связано с тем, что работу алгоритма минимизации можно условно разделить на ряд этапов:

- управление ходом минимизации (обращение к различным этапам вычислений и обработка поступающей информации);
- выбор направления движения к минимуму на каждой итерации (сведение многомерной задачи к одномерной);
- линейный поиск (вдоль выбранного направления);
- вычисление градиентов (кроме прямых методов);
- вычисление значений целевой функции.

В связи с этим структура библиотеки может быть представлена в следующем виде (рис. 23, а).

В первый уровень объединены программы, вычисляющие значение целевой функции.

Во второй уровень – программы линейного поиска:

- подпрограммы, не использующие производные целевой функции;
- подпрограммы, не использующие первые производные для линейного поиска.

В третий уровень объединены подпрограммы, вычисляющие градиенты целевой функции:

- подпрограммы численного дифференцирования;
- вычисления по явным выражениям.

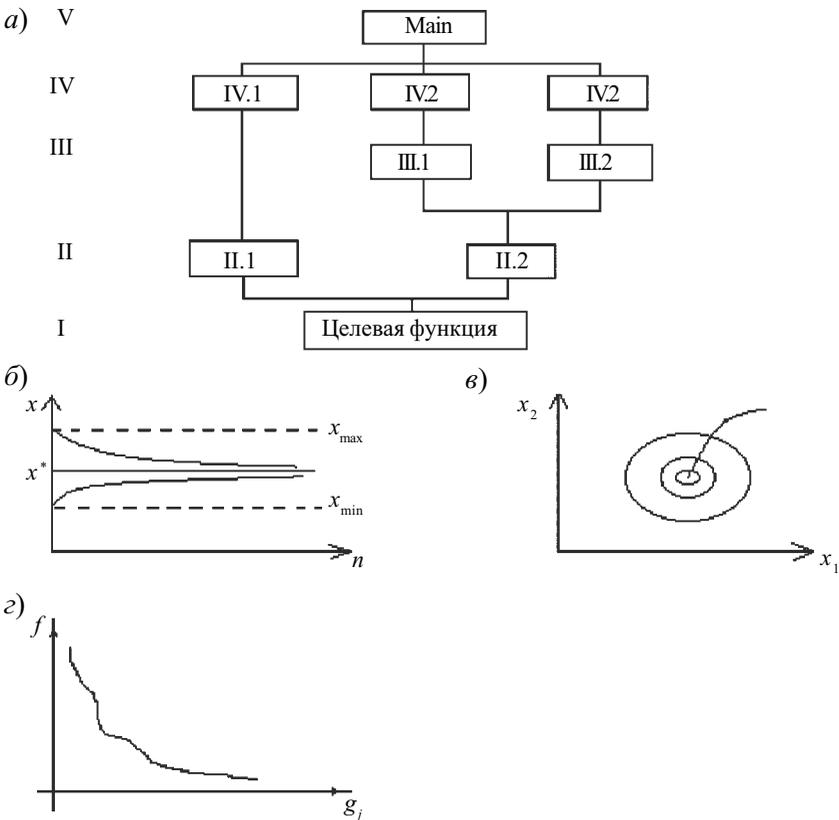


Рис. 23

В четвертый уровень объединены:

- вычисление вектора направления и управление ходом минимизации;
- прямые методы;
- квазиградиентные (первые производные вычисляются численным дифференцированием);
- градиентные (аналитическое вычисление первых производных);

Пятый уровень – вызывающая программа.

Пользователь должен иметь возможность организации своей сборки комплекса (набор подпрограмм из разных уровней) помимо сборок, рекомендуемых по умолчанию (для прямых, квазиградиентных, градиентных методов).

Набор методов минимизации может быть расширен пользователем (должна быть обеспечена открытость системы). Должны присутствовать не только средства, обеспечивающие подключение новых методов поиска в виде отдельных подпрограмм на языке программирования с существующими библиотеками, но и возможность описания алгоритмов оптимизации на пользовательском языке с вставками стандартных функций. Например, следующий фрагмент программы реализует градиентный метод:

```
Repeat {  
  Evaluate h = grad(x[Iter]);  
  Lambda = step(x[Iter],h);  
  Update x[Iter+1] = x[Iter]-lambda*h;  
  Iter++;  
  Output;  
}
```

Функция “step” вычисляет величину шага, x[Iter] содержит результаты итерационных вычислений вектора управляемых параметров.

Целесообразно включить подпрограммы проверки критерия конца итерационного процесса (варианты критерия окончания итерации). Кроме того, библиотека должна включать алгоритмы глобального поиска (например, библиотеку методов случайного поиска).

Наиболее важен вопрос с выбором метода четвертого уровня. Каждый метод оптимален для определенного класса целевых функций. В идеале, конечно, хотелось бы обеспечить автоматизированный выбор алгоритма оптимизации. При этом можно использовать таб-

лицы решений, по которым на основе классификации задачи – условий (наличие ограничений, их вид, унимодальность или многоэкстремальность целевой функции, доступность производных и т. п.) производится выбор подходящих алгоритмов, заполняя таблицу типа “да” – “нет”. Если пользователь сам выбирает цепочку методов, то таблица решений служит для проверки правильности выбора. Известен подход, основанный на анализе поверхностей функции. Такой анализ позволяет выявить наличие многоэкстремальности, приблизиться к глобальному минимуму, а также определить параметры, слабо влияющие на критерий и снизить размерность задачи.

Алгоритм, основанный на методе ранговой корреляции [20] при анализе сечений $f(x)$, рассматривает сечение целевой функции $f(x)$ по переменной x_f (остальные переменные фиксируются в определенной точке). Проводятся n случайных испытаний, распределенных по равномерному закону. Если соседние значения x_f отличаются не более чем на ϵ , то f не вычисляется. Каждому i -му значению параметра x_f^i присваивается ранг (порядковый номер) в порядке возрастания (значения f также должны быть упорядочены по возрастанию). На основе сформированного таким образом массива K рангов массива f вычисляется коэффициент ранговой корреляции $\tau = 4P/(n(n-1)) - 1$ (где P – сумма количеств тех пар последовательностей рангов исследуемой функции $f(x)$, которые образуют возрастающий порядок). Если коэффициент корреляции равен 1 функция $f(x_f)$ возрастает в зависимости от x_f , если равен -1 , то $f(x_f)$ убывает, если τ лежит в диапазоне от -1 до 1 – неполная корреляция (при τ малых соответствие между последовательностями отсутствует, т. е. переменная x_f не влияет на f). Если f по каждой переменной изменяется монотонно или имеет один экстремум, можно перейти к детерминированному методу поиска с начальной точкой, ближайшей к экстремуму. Если хотя бы одно сечение получается многоэкстремальным, говорят о многоэкстремальности функции в целом, т. е. необходим переход к глобальному методу поиска. Если сечения одноэкстремальны или монотонны, то вероятность многоэкстремальности мала.

11.2. Организация вычислительного процесса

К задаче выбора метода тесно примыкает вопрос организации вычислительного процесса. Некоторые методы минимизации хорошо

сходятся вблизи оптимальной точки и плохо вдали, другие, наоборот, быстро сходятся вдали от окрестности оптимальности и медленно в самой окрестности. Поэтому выделим следующие стратегии поиска:

1. На первых этапах целесообразно применять процедуры, обладающие глобальными свойствами (например, случайный поиск) для получения грубой оценки глобального экстремума. Затем производится уточнение решения быстро сходящимися в окрестностях оптимума локальными алгоритмами.

2. Параллельное решение задачи одновременно двумя и более алгоритмами. При этом каждый алгоритм выполняет по несколько итераций поиска, после чего управление передается контролирующему блоку. Он сравнивает результаты по значениям целевой функции. Если разница в значениях $f(x)$ достаточно велика, то отстающий процесс целесообразно переключить на рекордную точку лидирующего процесса и так далее, поощряя каждый раз более удачный подпроцесс. Можно производить поиск вдоль прямой, соединяющей текущие точки двух подпроцессов.

Заметим, что точки останова некоторых алгоритмов являются начальными для запуска более перспективных процедур. Промежуточные решения должны запоминаться в соответствующем файле. В системе необходимо предусмотреть выбор точки, с которой может быть продолжено решение.

11.3. Элементы пользовательского интерфейса

1. Практика показывает, что наиболее гибкой является следующая форма задания математической модели (ММ).

а) Целевая функция и ограничения могут быть заданы в аналитическом виде. В этом случае, при наличии в системе редактора аналитических выражений может быть использована, например, следующая запись:
Funct ($ax_1^2 + bx_2^2 + \dots$), constr ($cx_1 + 5x_2 > 0, \dots, 2x_1^3 + 3x_2 = 0$).

б) Чтение файла с целевой функцией из библиотеки ММ.

в) При алгоритмическом задании целевой функции пользователь должен иметь возможность подключения собственных функций на специальном языке.

Числовые данные, например: численные значения буквенных коэффициентов, диапазоны изменения неизвестных параметров и т. д. могут быть заданы в виде таблиц.

*Примерная структура окон монитора
для управления решением задачи*

1. Окна управления и данных.

Задание:

1 – общие параметры управления вычислительным процессом;

2 – выбор метода решения;

3 – числовые данные.

Окно содержит параметры, общие для всех методов:

– точность решения задачи по функционалу;

– точность решения задачи по ограничениям;

– число итераций;

– ведение протокола.

Эти параметры имеют значения по умолчанию, которые могут изменяться пользователем (табл. 11.1 и 11.2)

Таблица 11.1

Имя поля (параметра)	Значения по умолчанию
Точность функционала	1,0e-004
Точность ограничения	1,0e-004
Число шагов	5
Протокол	√

Точность функционала оценивается по относительному изменению целевой функции за один шаг. “Точность по ограничениям” определяется как суммарный штраф за нарушение ограничений. Если эти две величины меньше заданных, то процесс прекращается.

“Число шагов” определяет число итераций, которое можно выполнить тем или иным методом.

Протокол указывает, с какой подробностью следует вести протокол (вывод значений целевой функции, значений всех ограничений, текущая точка):

√ – наличие протокола, меню, содержащее “счет”, “стоп”, “выход” (по “выходу” можем попасть в редактор выражений для коррекции исходной постановки задачи).

2. Блок выбора метода.

“Выбор метода” позволяет установить из библиотеки нужный метод или задать цепочку выполнения подпрограмм.

Каждый метод в дополнение к общим параметрам всех методов обладает собственным набором управляющих параметров (шаг спуска, коэффициент изменения штрафа и т. д.). У каждого метода список и назначение этих параметров различны. С каждым параметром связано свое поле. Эти поля появляются в окне при установке в меню конкретного метода.

3. Блок численных данных (для оперативного анализа результатов) содержит две группы числовых полей.

Таблица 11.2

1	2
Точка	–
Функция	Штраф
Ограничения	Изменение функции

В поле ТОЧКА выводятся значения координат текущей точки. В полях ФУНКЦИЯ, ОГРАНИЧЕНИЯ помещаются значения минимизируемой функции и ограничений. Перевод курсора на поле с именем ФУНКЦИЯ вычисляет значение функции в текущей точке. Аналогично, перевод курсора на поле любого элемента вектора ограничений приводит к вычислению всех значений ограничений данного типа в текущей точке B_n . В поле ШТРАФ выводится величина

$$\sum_{i \in I} |g_i|,$$

где I – множество ограничений (принадлежит ли точка допустимому множеству и если нет, то как далеко она лежит от его границ). Поле ИЗМЕНЕНИЕ ФУНКЦИИ показывает относительное изменение минимизируемой функции.

В окне сообщений должна выводиться информация об ошибках, число итераций в подпрограмме вычисления минимизируемой функции и ее градиента (должна быть возможность сброса этих полей в нуль).

11.4. Порядок работы

Можно представить себе следующую схему работы с программой. После задания модели (целевой функции, ограничений, числовых значений параметров) осуществляется выбор методов минимизации и последовательности их применения. Задаются параметры настройки для

выбранного метода (точности, число итераций и пр.). Затем выполняется собственно поисковая процедура оптимизации. Делается анализ полученного решения и заключение о продолжении или прекращении поиска (выдача графической информации характеризует сходимость процесса) (рис. 23, б и в).

Необходимо иметь возможность прервать работу метода, выполнить анализ решения, изменение параметров метода, самого метода, начального приближения, возможно, самой целевой функции и возобновление процесса.

11.5. Графический контроль решения

На рис. 23, з в качестве горизонтальной оси графика выбрано изменение номера итераций N для показа траекторий изменения координат.

По виду графика можно судить о качестве решения одновременно по значениям минимизируемой функции и ограничений. Номер ограничения, модуль которого имеет максимальное значение, может меняться от итерации к итерации. Вертикальная ось соответствует нулевому значению ограничений. Если решение задачи находится на границе допустимого множества, то для методов внутренней точки будет характерным приближение к нулевому значению ограничений слева. Для методов внешней точки – приближение справа. По другой оси откладывается текущее значение оценки оптимального значения минимизируемой функции.

Для визуального контроля необходим пункт меню, позволяющий вывести траектории изменения управляемых параметров на координатную плоскость (путем выбора любых двух элементов вектора параметров для задания плоскости проекции – рис. 23, в). Иначе говоря, это показ линий равного уровня (изолиний) ЦФ на экран для контроля положения минимума относительно направления поиска.

Алгоритм состоит в изменении одного из аргументов, например x_i , с некоторым шагом и решения нелинейного уравнения $f(x_i) - \text{const} = 0$ относительно одного неизвестного с последующим отделением вещественных корней. Значения постоянной const представляют собой градации размаха значений ЦФ в диапазоне изменения аргументов (из расчета показа, например, 5 уровней).

Другой способ состоит в подстановке точек экрана, снабженных координатами x_i и x_j , в функцию, отборе и построении точек, которые

обращают левую часть в нуль. Заметим, что поскольку интересует лишь качественная картина, то погрешность может быть достаточно высокой. Для обеспечения непрерывности кривые можно интерполировать с помощью B -сплайнов. Если минимум неединственный, имеется седловая точка и т. п. Ситуация усложняется необходимостью указания программе, какие точки сечения не могут быть соединены.

Файл, формируемый программой, содержит информацию о числе сечений, количестве точек в каждом сечении и координатах точек. Последнее сечение хранит последовательность координат точек, найденных процедурой минимизации на последних пяти шагах алгоритма.

Библиографический список

1. Банди Б. Методы оптимизации: Вводный курс. М.: Радио и связь, 1988.
2. Реклейстис Г., Рейвиндран А., Регодел К. Оптимизация в технике. М.: Мир, 1986.
3. Гилл Ф., Мюррей У., Райт М. Практическая оптимизация. М.: Мир, 1985.
4. Филлипс Д., Герено-Диас А. Методы анализа сетей. М.: Мир, 1984.
5. Химмельблау Дж. Прикладное нелинейное программирование. М.: Мир, 1975.
6. Батищев Д. И. Методы оптимального проектирования. М.: Сов. радио, 1984.
7. Уайлд Д. Оптимальное проектирование. М.: Мир, 1981.
8. Таха Х. Введение в исследование операций. М.: Мир, 1985.
9. Банди Б. Основы линейного программирования. М.: Радио и связь, 1988.
10. Муртаф В. Современное линейное программирование. М.: Мир, 1984.
11. Майника Э. Алгоритмы оптимизации на графах и сетях. М.: Мир, 1981.
12. Зенер К. Геометрическое программирование и техническое проектирование. М.: Мир, 1973.
13. Сухарев А. Г., Тимохов А. В., Федоров В. В. Курс методов оптимизации. М.: Наука, 1986.
14. Поляк Б. Т. Введение в оптимизацию. М.: Наука, 1983.
15. Численные методы условной оптимизации/ Пер.с англ. под ред. Ф. Гилла и У. Мюррея. М.: Мир, 1977.
16. Базара М., Шетти К. Нелинейное программирование. Теория и алгоритмы: Пер. с англ. М.: Мир, 1982.
17. Гуснин С. Ю., Омелянов Т. А. и др. Минимизация в инженерных расчетах на ЭВМ. М.: Машиностроение, 1981.
18. Брахман Т. Р. Многокритериальность и выбор альтернатив в технике. М.: Радио и связь, 1985.
19. Сольницев Р. И. Основы автоматизации проектирования гироскопических систем. М.: Высш. шк., 1985.
20. Ильин В. Н., Камнева Н. Ю. Автоматическое исследование вида функции качества электронных схем// Изв. вузов СССР. Радиоэлектроника. 1981. Т. 24. С. 67–72.

Оглавление

Предисловие	3
1. Задача оптимального проектирования в САПР	5
1.1. Понятие о структурном и параметрическом синтезе	6
1.2. Примеры постановок задач параметрической оптимизации ..	9
1.3. Формализация процесса принятия оптимальных решений. Математическая модель ОП	13
1.4. Формализация технико-эксплуатационных требований, предъявляемых к объекту проектирования	14
1.5. Математические модели принятия оптимальных решений ..	16
1.6. Классификация экстремальных задач, описывающих процесс принятия оптимальных решений	18
1.7. Примеры схем оптимального параметрического синтеза	20
1.8. Способ построения функционала при проектировании	22
1.9. Функции многих переменных	23
1.10. Критерии положительной определенности матриц	24
2. Методы безусловной оптимизации	27
2.1. Линейный поиск без использования производных	27
2.2. Линейный поиск с использованием производной миними- зируемой функции	33
2.3. Многомерный поиск без использования производных	35
2.4. Градиентные методы	43
2.5. Методы сопряженных градиентов	48
2.6. Квазиньютоновские методы	50
2.7. Методы минимизации овражных функционалов	53
2.8. Практические вопросы	56
3. Методы условной оптимизации	61
3.1. Критерии оптимальности в задачах с ограничениями	61
3.2. Экономическая интерпретация множителей Лагранжа	64
3.3. Условия оптимальности Куна–Таккера	65
3.4. Практическая проверка условий оптимальности	69
3.5. Функция Лагранжа и двойственность	73
3.6. Задача, двойственная по Лагранжу	75
3.7. Методы оптимизации на основе преобразования задачи	77
3.8. Методы прямого поиска в задачах условной оптимизации ..	83
3.9. Методы случайного поиска	87

4. Линейное программирование (ЛП)	89
4.1. Приведение задачи ЛП к каноническому виду	91
4.2. Табличный симплекс-метод	91
4.3. Двойственные задачи в ЛП	98
5. Методы линеаризации для задач условной оптимизации	108
5.1. Алгоритм Франка–Вульфа	109
6. Сепарабельное программирование (СП)	111
7. Геометрическое программирование (ГП)	115
7.1. Постановка задачи	115
7.2. Решение задачи геометрического программирования с ограничениями	119
8. Сведение задач векторной оптимизации к однокритериальным экстремальным задачам	123
8.1. Решения, оптимальные по Парето	124
8.2. Обобщенные критерии оптимальности	125
8.3. Пример использования минимаксной (максиминной) свертки векторного критерия	131
8.4. Метод главного критерия	132
8.5. Метод последовательных уступок	133
8.6. Способы назначения весовых коэффициентов важности для частных критериев оптимальности	134
8.7. Функции “полезности”	136
9. Динамическое программирование	138
10. Дискретное программирование	144
10.1. Задача о назначениях	144
10.2. Венгерский метод решения задачи о назначениях	146
10.3. Транспортная задача	149
10.4. Задача о коммивояжере	151
10.5. Метод ветвей и границ для решения задачи о коммивояжере	153
11. Принципы организации диалоговой подсистемы параметри- ческой оптимизации	157
11.1. Архитектура	157
11.2. Организация вычислительного процесса	160
11.3. Элементы пользовательского интерфейса	161
11.4. Порядок работы	163
11.5. Графический контроль решения	164
Библиографический список	166

Учебное издание

Андронов Сергей Александрович

**МЕТОДЫ
ОПТИМАЛЬНОГО
ПРОЕКТИРОВАНИЯ**

Текст лекций

Редактор *А. В. Семенчук*
Компьютерная верстка *А. Н. Колешко*

Лицензия ЛР №020341 от 07.05.97. Сдано в набор 16.09.01. Подписано к печати 29.12.01.
Формат 60×84 1/16. Бумага тип. №3. Печать офсетная. Усл. печ. л. 9,8. Усл. кр.-отт. 10,7.
Уч. -изд. л. 10,5. Тираж 100 экз. Заказ №

Редакционно-издательский отдел
Лаборатория компьютерно-издательских технологий
Отдел оперативной полиграфии
СПбГУАП
190000, Санкт-Петербург, ул. Б. Морская, 67