С. А. ОЛЕЙНИКОВА,
Т.Н. НЕДИКОВА

МАТЕМАТИЧЕСКИЕ МОДЕЛИ И ЧИСЛЕННЫЕ МЕТОДЫ РЕШЕНИЯ ОПТИМИЗАЦИОННЫХ ЗАДАЧ

Учебное пособие

Печатается по решению Ученого Совета факультета информационных систем и компьютерной безопасности Воронежского государственного технического университета

Воронеж 2022

УДК
ББК
О

Рецензенты:
кафедра информационной безопасности и систем связи
Международного института компьютерных технологий
(зав. кафедрой
канд. техн. наук, доцент О.С. Хорпяков);

докт. техн. наук, проф. Белецкая С.Ю.

Олейникова С. А. Математические модели и численные методы решения оптимизационных задач/ С.А. Олейникова, Т.Н. Недикова - ФГБОУ ВО «Воронежский государственный технический университет». - Воронеж: Изд-во ВГТУ, 2022. – 80 с.

ISBN

Учебное пособие содержит необходимые теоретические сведения для освоения курса «Методы оптимизации». Особое внимание уделено численным методам решения задач непрерывной и дискретной оптимизации.

Предназначено для студентов направления 09.03.01 «Информатика и вычислительная техника» (профиль «Вычислительные машины, комплексы, системы и сети») очной и заочной форм обучения.

Ил. 15. Табл. 49. Библиогр.: 12 назв.

УДК 519.6(075.8
ББК 22.193я7

Печатается по решению редакционно-издательского совета
Воронежского государственного технического университета

ISBN 978-

Введение

Необходимость решения разнообразных оптимизационных задач возникает практически во всех сферах человеческой деятельности. Определение стратегий, позволяющих получить максимальную прибыль или минимизировать затраты, является одной из ключевых проблем при производственном планировании и управлении. В связи с этим, изучение методов, позволяющих с той или иной степенью точности найти решение, доставляющее оптимум некоторой функции, является актуальной и практически важной задачей.

В данном пособии содержатся математические модели и методы решения задач непрерывной и дискретной оптимизации. В связи с тем, что сложность исследуемых на практике задач, как правило, не позволяет применить к ним аналитический аппарат, основное внимание сосредоточено на численных методах, позволяющих найти решение, близкое к оптимальному, которое, как правило, удовлетворяет заданной точности. Первая глава описывает численные методы решения задач на безусловный экстремум. Рассмотрены методы нулевого, первого и второго порядков, приведены иллюстрации их работы на конкретных задачах. Вторая глава посвящена решению задач на условный экстремум. Остальная часть пособия направлена на изучение задач дискретной оптимизации. Третья глава посвящена описанию точных алгоритмических методов: метода ветвей и границ, а также метода динамического программирования. Приближенные методы решения дискретных задач (эвристические и эволюционные методы) приведены в четвертой части.

Все главы снабжены достаточным количеством примеров, облегчающих не только процесс изучения того или иного метода, но и его машинную реализацию.

Пособие может быть полезно бакалаврам, изучающим дисциплину «Методы оптимизации», а также студентам других специальностей, научный интерес которых лежит в области математического моделирования и оптимизационных задач.

Глава 1 Численные методы решения задачи на безусловный экстремум

Глава посвящена исследованию методов, позволяющих найти численное решение многомерной оптимизационной задачи без ограничений. В первой части главы приведены общие сведения о численных методах решения данной задачи: общая формула, сходимость, условие останова и т.д. Прямые методы поиска, использующие при формировании решения лишь сведения о значении функции, представлены во второй и третьей главах (метод Хука-Дживса и Нелдера-Мида соответственно). Особенностям градиентных методах первого порядка посвящена четвертая часть главы. В пятой части описан метод Ньютона и его модификации.

1.1. Общие сведения о численных методах решения задачи на безусловный экстремум

Как правило, исследуемая задача является достаточно громоздкой. В этом случае использование аналитических подходов к ее решению не всегда целесообразно, а иногда просто невозможно. Тогда для ее решения применяется аппарат численных методов.

Любой численный алгоритм решения задачи оптимизации основан на точном или приближенном вычислении значений целевой функции, функций, задающих допустимое множество (для задач условной оптимизации) и их производных. На основании полученной информации строится приближение к решению задачи.

Для каждой конкретной задачи вопрос о том, какие характеристики необходимо выбирать для вычисления решается в зависимости от свойств минимизируемой функции, ограничений и имеющихся возможностей по хранению и обработке информации. Так, для не дифференцируемой функции нельзя воспользоваться алгоритмом, предусматривающим возможность вычисления производной в точке градиента функции.

Алгоритмы, использующие информацию лишь о значениях минимизируемой функции, называются алгоритмами нулевого порядка; алгоритмы, использующие также информацию о значениях первых производных – алгоритмами первого порядка и т.д.

Работа алгоритмов, как правило, состоит из двух этапов. На первом этапе вычисляются предусмотренные алгоритмом характеристики задачи. На втором этапе по полученной информации строится приближение к решению.

Если все точки выбираются одновременно до начала вычислений, то такой алгоритм минимизации называется пассивным.

Однако, для решения большинства задач точки вычисления выбираются поочередно, т.е. точка x^{i+1} находится, когда уже выбраны точки $x^1, x^2,\ldots,x^k$. Такие алгоритмы называются последовательными.

В дальнейшем для записи методов минимизации будем использовать сокращение вида:

$$x^{k+1} = x^k + \alpha_k \cdot h^k. \quad (1.1)$$

Вектор h^k определяет направление (k+1)-го шага оптимизации, а коэффициент α_k - длину этого шага.

При этом конкретный алгоритм определяется заданием начальной точки x^0, правилами выбора векторов hk и чисел α_k на основе полученной в результате вычислений информации, а также условием остановки.

Правила выбора α_kи hk могут предусматривать и дополнительные вычисления, т.е. вычисления некоторых характеристик в точках, отличных от точек x^0, x^1, …, x^k.

Обычно название метода определяется способом выбора h^k, а его различные варианты связываются с разными способами выбора α_k.

Трудность решения задачи оптимизации определяется числом переменных, видом целевой функции, а также видом и числом ограничений.

Сходимость методов оптимизации

Важной характеристикой численных методов является их сходимость.

Говорят, что метод (2.1) сходится, если

$$x^k \underset{k \to \infty}{\longrightarrow} x^*. \quad (1.2)$$

Эффективность сходящегося метода можно охарактеризовать с помощью скорости сходимости.

Рис. 1.1. Иллюстрация численных методов решения задачи на безусловный экстремум

Пусть выполнено условие (1.2). Говорят, что последовательность x^kсходится к x^* линейно (с линейной скоростью, со скоростью геометрической прогрессии), если существуют такие константы q: 0≤q≤1и k_0, что

$$\|x^{k+1} - x^*\| \le q\|x^k - x^*\| \text{при } k \ge k_0. \quad (1.3)$$

Большинство теорем о сходимости методов оптимизации доказывается в предположении о выпуклости целевой функции, оценки же скорости сходимости устанавливаются при ограничительном предположении сильной выпуклости. Для невыпуклых задач численные методы обычно позволяют отыскивать лишь локальные решения, т.е. стационарные точки.

Установление факта сходимости и оценка скорости сходимости дают существенную информацию о выбранном методе минимизации. В первую очередь, требования, которые приходится накладывать на минимизируемую функцию, показывают область применимости метода.

Для задания конкретного вычислительного алгоритма бесконечношаговый метод необходимо дополнить условием остановки.

Условия остановки

Условие остановки может определяться имеющимися в наличии вычислительными ресурсами. Остановка может производиться и по достижении заданной точности. Однако, при решении реальной задачи сложно оценить истинную точность. Поэтому о достижении точности необходимо судить по косвенным признакам.

На практике часто используются следующие условия остановки:

$$\left\|x^{k+1}-x^{k}\right\| \le \varepsilon 1; \tag{1.4}$$

$$\left\|f(x^{k+1})-f(x^{k})\right\| \le \varepsilon 2; \tag{1.5}$$

$$\left\|f'(x^{k+1})\right\| \le \varepsilon 3. \tag{1.6}$$

До начала вычислений выбирается одно из условий останова (1.4)-(1.6) и соответствующая точность ε.

Критерий (1.6) относится в первую очередь к задачам безусловной оптимизации.

Направление убывания и методы спуска

Многие методы оптимизации относятся к методам спуска. В них направление движения к минимуму на каждом шаге выбирается из числа направлений убывания минимизируемой функции.

Говорят, что вектор h задает направление убывания функции f в точке x, если $f(x+\alpha h)<f(x)$ для всех достаточно малых $\alpha>0$. Сам вектор h иногда называют иногда направлением убывания.

Множество всех направлений убывания функции f в точке x обозначим через U(x,f). Т.о. если любой достаточно малый сдвиг из x в направлении вектора h приводит к уменьшению значения функции f, то h принадлежит U(x,f).

1.1. Методы решения задач нулевого порядка. Метод Хука-Дживса

Данный метод принадлежит к методам нулевого порядка (или методам прямого поиска) для определения минимума функции n переменных. Это означает, что при поиске минимума (максимума) используются только значения функции.

Данный метод представляет собой комбинацию исследующего поиска с циклическим изменением переменных и ускоряющего поиска по образцу. Исследующий поиск ориентирован на выявление локального поведения целевой функции и определение направления ее убывания вдоль «оврагов». Полученная информация используется при поиске по образцу.

Метод состоит из следующих шагов.

Предварительная (нулевая итерация). Задать начальную точку, шаг для каждой координаты и точность.

Основная итерация включает в себя исследующий поиск и, если он оказался удачным, то поиск по образцу. Опишем данные виды поиска подробнее.

Пусть имеется некоторая точка (назовем ее базисной) $\mathbf{b_1}$. Суть исследующего поиска заключается в последовательном изменении каждой из координат базисной точки и сравнении значения функции в новой и старой точках. Как только получено уменьшение функции, старая базисная точка заменяется на новую, и изменение последующих координат осуществляется уже относительно нее.

Иллюстрация исследующего поиска приведена на *рис. 1.2*.

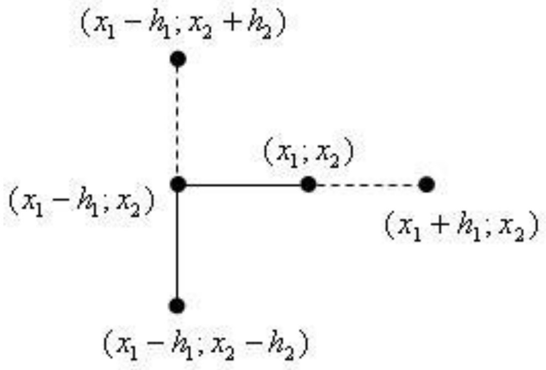

Рис. 1.2. Иллюстрация исследующего поиска метода Хука-Дживса

По завершении исследующего поиска будет получена новая базисная точка $\mathbf{b_2}$. Могут возникнуть две ситуации.

1. Если $b_2 = b_1$, т. е. уменьшение функции не было достигнуто. Это означает необходимость исследования вокруг этой же базисной точки, но с меньшим шагом. На практике удовлетворительным является уменьшение шага (шагов) в k раз от начальной длины, где k может принимать значения от 2 до 10.

2. Если $b_2 \neq b_1$, то производится *поиск по образцу*. Поиск по образцу заключается в движении по направлению от старого базиса к новому.

Поиск по образцу производится следующим образом.

1. Разумно двигаться из новой базисной точки $\boldsymbol{b_2}$ в направлении $\boldsymbol{b_2} - \boldsymbol{b_1}$, поскольку поиск в этом направлении уже привел к уменьшению значения функции. Вычисляют координаты новой точки в направлении $\boldsymbol{b_2} - \boldsymbol{b_1}$ (точку образца, ускоряющий множитель выбран равным 2):

$P_1 = b_1 + 2 \cdot (b_2 - b_1)$.

В общем случае

$P_1 = b_i + 2 \cdot (b_{i+1} - b_i)$.

2. Заменяют значения в базисной точке b_i на значения в b_{i+1}, значения во второй базисной точке заменяют на P_i, то есть:

$b_i = b_{i+1}$, а $b_{i+1} = P_i$ для всех i.

$$x^i_{\text{нов\_обр}} = x^i + 2(x^i_{\text{нов}} - x^i)$$

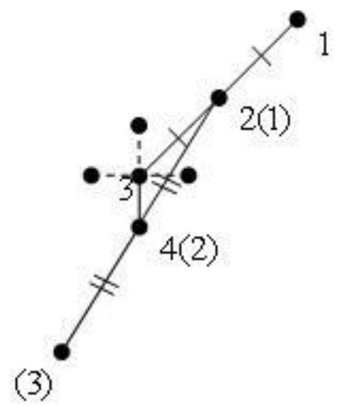

Рис. 1.3. Иллюстрация поиска по образцу метода Хука-Дживса

Процесс поиска решения заканчивают, когда длина шага (длины шагов) будет меньше заданного малого значения.

Недостаток метода Хука Дживса состоит в том, что в случае сильно вытянутых, изогнутых или обладающих острыми углами линий уровня целевой функции он может оказаться неспособным обеспечить продвижение к точке минимума. Представим это на следующем примере.

Пример.

$$f(x,y) = 3x^4 - 6x^2 + 4xy + y^2$$

Решим данный пример аналитически.

$$\frac{\partial f}{\partial x} = 12x^3 - 12x + 4y;$$

$$\frac{\partial f}{\partial y} = 4x + 2y.$$

Приравняв к нулю производные, получим систему уравнений:

$$\begin{cases} 12x^3 - 12x + 4y = 0; \\ 4x + 2y = 0. \end{cases}$$

Выразим из второго уравнения y и подставим в первое.

$$\begin{cases} y = -2x; \\ 12x^3 - 12x - 8x = 0. \end{cases}$$

Найдем x из кубического уравнения:

$$12x^3 - 20x = 0.$$

Корни данного уравнения следующие:

$$\begin{cases} x = 0 \\ x = -\sqrt{\frac{5}{3}} \approx -1.291; \\ x = \sqrt{\frac{5}{3}} \approx 1.291. \end{cases}$$

Таким образом, получим следующие точки, подозрительные на минимум:

(0,0); (-1.291, 2.582); (1.291, -2.582).

Найдем матрицу вторых производных и исследуем каждую из полученных точек на минимум.

$$\frac{\partial^2 f}{\partial x^2} = 36x^2 - 12;$$

$$\frac{\partial^2 f}{\partial x \partial y} = 4;$$

$$\frac{\partial^2 f}{\partial y^2} = 2.$$

Таким образом:

$$f'' = \begin{pmatrix} 36x^2 - 12 & 4 \\ 4 & 2 \end{pmatrix}.$$

1. Исследование на минимум точки (0,0).

$\Delta_1 = -12 < 0.$

Точка (0,0) не является точкой минимума.

2. Исследование на минимум точки (-1.291, 2.582).

$$\Delta_1 = 36 \cdot (-1.291)^2 - 12 = 48 > 0;$$
$$\Delta_2 = 48 \cdot 2 - 4 \cdot 4 = 80 > 0.$$

Следовательно, точка (-1.291, 2.582) является точкой минимума.

3. Исследование на минимум точки (-1.291, 2.582).

$$\Delta_1 = 36 \cdot (1.291)^2 - 12 = 48 > 0;$$

$$\Delta_2 = 48 \cdot 2 - 4 \cdot 4 = 80 > 0.$$

Следовательно, точка (1.291, -2.582) является точкой минимума.

Таким образом, в данном примере имеются две точки минимума: (-1.291, 2.582) и (1.291, -2.582).

Рассмотрим, каким образом будет работать метод Хука-Дживса в зависимости от выбора начальных условий.

Вариант 1. Начальная точка (-2, 2).

Шаг 1. Задаем исходные данные: начальная точка x^0=(-2,2); приращения (шаги): Δx=(1,1); коэффициент уменьшения шага: α = 2; ε=0.1

$$f(x^0) = 12.$$

Итерация №1.

1.1. Исследующий поиск.

Фиксируя переменную x_2 = 2, дадим приращение x_1:

x=-2 + 1 = -1;

$$f(-1,2) = -7 < 12.$$

Направление для x_1 выбрано удачно. Фиксируем его и задаем приращение для y:

y=2 + 1 = 3.

$$f(-1,3) = -6 > -7.$$

Зададим приращение в другом направлении:

y=2 - 1 = 1.

$$f(-1,1) = -6 > -7.$$

Таким образом, вторая координата остается без изменений.

На этом исследовательский поиск завершен. В результате получили новую точку (-1,2). Переходим ко второй части метода – поиска по образцу.

1.2. Поиск по образцу.

Новая точка образца определяется по формуле:

$$x_p^k = x^k + (x^k - x^{k-1}).$$

Подставив в данную формулу точку $x^k = (-1,2)$, получим:

$$x_p^k = (0,2).$$
$$f\left(x_p^k\right) = 4.$$

Поскольку значение не уменьшилось, оставляем в качестве базовой точку (-1,2).

Итерация №2. Базовая Точка (-1,2). Шаг 1.

1.1. Исследующий поиск.

Фиксируя переменную y = 2, делаем приращение для аргумента x: x=-1+ 1 = 0.

$f(0,2) = 4 > -7.$

Делаем приращение в другую сторону:

x=-1- 1 = -2.

$$f(-2,2) = 12 > -7.$$

Аргумент x остается без изменения. Изменяем аргумент y (фиксируя x=0):

y=2 + 1 = 3.

$$f(-2,3) = 9 > -7.$$

y=2 - 1 = 1.

$$f(-2,1) = 17 > -7.$$

Таким образом, исследующий поиск не был удачным.

Проверка на окончание поиска:

$$\sqrt{1^2 + 1^2} = 1.4142 > 0.1.$$

Неравенство не выполняется, поэтому уменьшаем приращение.

$$\Delta x = \frac{1}{2} = 0.5;$$

$$\Delta y = \frac{1}{2} = 0.5.$$

Итерация №3. Точка (-1,2). Шаг 0.5.

3.1. Исследующий поиск.

Фиксируя переменную y = 2, делаем приращение для аргумента x:

$$x = -1 + 0.5 = -0.5.$$

$f(-0.5,2) = -1.3125 > -7.$

Делаем приращение в другую сторону:

$$x = -1 - 0.5 = -1.5$$

$$f(-1.5,2) = -6.3125 > -7.$$

Аргумент x_1 остается без изменения. Изменяем аргумент y:

$$y = 2 + 0.5 = 2.5.$$

$$f(-1,2.5) = -6.75 > -7.$$

И в другом направлении:

$$y = 2 - 0.5 = 1.5.$$

$$f(-1,1.5) = 6.75 > -7.$$

Таким образом, и на этой итерации исследующий поиск не был удачным.

Проверка на окончание поиска:

$$\sqrt{0.5^2 + 0.5^2} = 0.7071 > 0.1.$$

Неравенство не выполняется, поэтому уменьшаем приращение.

$$\Delta x = \frac{0.5}{2} = 0.25;$$

$$\Delta y = \frac{0.5}{2} = 0.25.$$

Итерация №4. Точка (-1,2). Шаг 0.25.

4.1. Исследующий поиск.

Фиксируя переменную y = 2, делаем приращение для аргумента x:

$$x = -1 + 0.25 = -0.75.$$

$f(-0.75{,}2) = -4.42578 > -7.$

Делаем приращение в другую сторону:

$$x = -1 - 0.25 = -1.25$$

$$f(-1.25{,}2) = -8.05078 < -7.$$

Зафиксировав новое значение x_1, меняем аргумент y:

$$y = 2 + 0.25 = 2.25.$$

$$f(-1.25{,}2.25) = -8.23828 < -8.05078.$$

Получена новая точка (-1.25, 2.25). Выполняем поиск по образцу.

4.2. Поиск по образцу.

Получаем точку

$$x_p^k = (-1.5{,}2.5).$$

$$f(-1.5{,}2.5) = -7.0625 > -8.23828.$$

Таким образом, оставляем точку (-1.25, 2.25).

Итерация №5. Точка (-1.25,2.25). Шаг 0.25.

1.1. Исследующий поиск.

Фиксируя переменную y = 2.25, делаем приращение для аргумента x:

$$x = -1.25 + 0.25 = -1$$

$$f(-1{,}2.25) = -6.9375 > -8.23828.$$

Делаем приращение в другую сторону:

$$x = -1.25 - 0.25 = -1.5$$

$$f(-1.5{,}2.25) = -6.75 > -8.23828.$$

Аргумент x остается без изменения. Изменяем аргументу:

$$y = 2.25 + 0.25 = 2.5.$$

$$f(-1.25{,}2.5) = -8.30078 < -8.23828.$$

Получена новая точка (-1.25, 2.5).

1.2. Поиск по образцу.

$$x_p^k = (-1.25{,}2.75).$$

$$f(-1.25{,}2.75) = -8.23828 > 8.30078.$$

Таким образом, остается точка (-1.25, 2.5).

Итерация №6. Точка (-1.25,2.5). Шаг 0.25.

6.1. Исследующий поиск.

Фиксируя переменную y = 2.5, делаем приращение для аргумента x:

$$x = -1.25 + 0.25 = -1$$
$$f(-1,2.5) = -6.75 > -8.30078.$$

Делаем приращение в другую сторону:

$$x = -1.25 - 0.25 = -1.5$$
$$f(-1.5,2.5) = -7.06 > -8.30078.$$

Аргумент x_1 остается без изменения. Изменяем аргумент y:

$$y = 2.5 + 0.25 = 2.75.$$
$$f(-1.25,2.75) = -8.23828 > -8.30078.$$

Зададим приращение в другую сторону:

$$y = 2.5 - 0.25 = 2.25.$$
$$f(-1.25,2.25) = -8.23828 > -8.30078.$$

Таким образом, исследующий поиск не привел к уменьшению целевой функции. Следовательно, поиск по образцу не проводим, а проверяем условие останова.

$$\sqrt{0.25^2 + 0.25^2} = 0.35353 > 0.1.$$

Следовательно, алгоритм продолжает свою работу с уменьшением шага.

$$\Delta x = \frac{0.25}{2} = 0.125;$$
$$\Delta y = \frac{0.25}{2} = 0.125.$$

Итерация №7. Точка (-1.25,2.5). Шаг 0.125.

7.1. Исследующий поиск.

Фиксируя переменную y = 2.5, делаем приращение для аргумента x:

$$x = -1.25 + 0.125 = -1.25.$$

$f(-1.125,2.5) = -7.78833 > -8.30078.$

Делаем приращение первой координаты в другую сторону:

$$x = -1.25 - 0.125 = -1.375$$
$$f(-1.375,2.5) = -8.12036 > -8.30078.$$

Таким образом, координата x остается без изменения. Изменяем аргумент y:

$$y = 2.5 + 0.125 = 2.5.$$
$$f(-1.25,2.625) = -8.28516 > -8.30078.$$

И в другом направлении:

$$y = 2.5 - 0.125 = 2.375.$$
$$f(-1.25,2.375) = -8.28516 > -8.30078.$$

Таким образом, и на этой итерации исследующий поиск не был удачным.

Проверка на окончание поиска:

$$\sqrt{0.125^2 + 0.125^2} = 0.1768 > 0.1.$$

Неравенство не выполняется, поэтому уменьшаем приращение.

$$\Delta x = \frac{0.125}{2} = 0.0625;$$

$$\Delta y = \frac{0.125}{2} = 0.0625.$$

Итерация №8. Точка (-1.25,2.5). Шаг 0.0625.

8.1. Исследующий поиск.

Фиксируя переменную y = 2.5, делаем приращение для аргумента x:

$$x = -1.25 + 0.0625 = -1.18755.$$

$f(-1.1875{,}2.5) = -7.78833 > -8.30078.$

Делаем приращение первой координаты в другую сторону:

$$x = -1.25 - 0.0625 = -1.3125$$

$$f(-1.3125{,}2.5) = -8.12036 > -8.30078.$$

Таким образом, координата x остается без изменения. Изменяем аргумент y:

$$y = 2.5 + 0.0625 = 2.5625.$$

$$f(-1.25{,}2.5625) = -8.28516 > -8.30078.$$

И в другом направлении:

$$y = 2.5 - 0.125 = 2.375.$$

$$f(-1.25{,}2.375) = -8.28516 > -8.30078.$$

Таким образом, исследующий поиск не был удачным. Проверка на окончание поиска:

$$\sqrt{0.0625^2 + 0.0625^2} = 0.088 < 0.1.$$

Неравенство выполняется, поэтому прекращаем процесс итераций.

Ответ: x(-1.25;2.5) $f(-1.25{,}2.5) = -8.30078.$

Сгруппируем результаты каждой итерации в следующую таблицу.

Табл. 1.1.

Результаты метода Хука-Дживса с начальной точкой (-2,2)

№ п/п	поиск	x	y	шаг	f(x, y)
0		-2	2	1	12
1	И (+)	-1	2	1	-7
	О (-)	0	2	1	4
2	И (-)	-1	2	0.5	-7
	О	Не проводился			
3	И (-)	-1	2	0.25	-7
	О	Не проводился			
4	И (+)	-1.25	2.25	0.25	-8.23828
	О (-)	-1.5	2.5	0.25	-7.0625
5	И (+)	-1.25	2.5	0.25	-8.30078

	О (-)	-1.25	2.75	0.25	-8.23828
6	И (-)	-1.25	2.5	0.125	-8.30078
	О	Не проводился			
7	И (-)	-1.25	2.5	0.0625	-8.30078
	О	Не проводился			
8	И (-)	-1.25	2.5	останов	-8.30078
	О	Не проводился			

Поясним обозначения в данной таблице. В столбце «поиск» символ «И» означает исследующий поиск; О – поиск по образцу. Если исследующий поиск был удачен, то в скобках после символа «И» стоит знак «+», в противном случае – «-».

Рассмотрим результаты данного метода в случае, если в качестве начального приближения взять другую точку: (2,-2). Сразу приведем результаты в виде таблицы.

Табл. 1.2.

Результаты метода Хука-Дживса с начальной точкой (2,-2)

№ п/п	поиск	x	y	шаг	f(x, y)
0		2	-2	1	12
1	И (+)	1	-2	1	-7
	О (-)	0	-2	1	4
2	И (-)	1	-2	0.5	-7
	О	Не проводился			
3	И (-)	1	-2	0.25	-7
	О	Не проводился			
4	И (+)	1.25	-2.25	0.25	-8.23828
	О (-)	1.5	-2.5	0.25	-7.0625
5	И (+)	1.25	-2.5	0.25	-8.30078
	О (-)	1.25	-2.75	0.25	-8.23828
6	И (-)	1.25	-2.5	0.125	-8.30078
	О	Не проводился			
7	И (-)	1.25	-2.5	0.0625	-8.30078
	О	Не проводился			
8	И (-)	1.25	-2.5	останов	-8.30078
	О	Не проводился			

Таким образом, выбор симметричных относительно начала координат точек привел к выбору другой точки минимума, которая также симметрична относительно первой точки относительно начала координат.

Выберем теперь точку, равноудаленную от каждой из двух точек минимума и рассмотрим работу данного метода. Результаты представлены в табл. 1.3.

Табл. 1.3.

Результаты метода Хука-Дживса с начальной точкой (2,-2)

№ п/п	поиск	x	y	шаг	f(x, y)
0		2	2	1	44
1	И (+)	1	1	1	2
	О (+)	0	0	1	0
2	И (+)	1	-1	1	-6
	О(+)	1	-2	1	-7
3	И (-)	1	-2	1	-7
	О	Не проводился			
4	И (-)	1	-2	0.5	-7
	О	Не проводился			
5	И (+)	1.25	-2.25	0.25	-8.23828
	О (-)	1.5	-2.5	0.25	-7.0625
6	И (+)	1.25	-2.5	0.25	-8.30078
	О(-)	1.25	-2.75	0.25	-8.23828
7	И (-)	-1.25	2.5	0.25	-8.30078
	О	Не проводился			
8	И (-)	-1.25	2.5	0,125	-8.30078
	О	Не проводился			
9	И (-)	-1.25	2.5	0,0625	-8.30078
	О	Не проводился			
10	И (-)	-1.25	2.5	останов	-8.30078
	О	Не проводился			

Следует отметить, что сходимость именно в такой точке минимума получилась из-за того, что вначале идет приращение по первой координате (в противном случае в результате работы была бы получена другая точка минимума).

Несомненным преимуществом данного метода является простота его реализации. Тем не менее, имеется серьезный недостаток, не позволяющий использовать данный алгоритм для любых функций. Он заключается в том, что если целевая функция сильно вытянутая, изогнутая или имеет достаточно острые углы линий уровня, то с помощью исследующего поиска и поиска по образцу невозможно обеспечить продвижение к точке минимума. Пример такой задачи приведен в [2].

1.3 Методы решения задач нулевого порядка. Метод Нелдера-Мида

Метод Нелдера – Мида, также известный как метод деформируемого многогранника – это еще один метод нулевого порядка, не использующий производной функции.

Суть метода заключается в последовательном перемещении и деформировании симплекса вокруг точки экстремума. В данном случае под симплексом понимается геометрическая фигура, являющаяся n-мерным обобщением треугольника.

Основная идея метода заключается в последовательном сужении симплекса вокруг точки минимума путем замены на каждом шаге точки с наибольшим значением функции на некоторую другую точку или сжатии всего симплекса вокруг точки с наименьшим значением функции.

Рассмотрим реализацию данной идеи и основные приемы метода Нелдера – Мида более подробно.

Пусть требуется найти безусловный минимум функции n переменных $f(x_1, x_2, \ldots, x_n)$. Предполагается, что ограничений на область определения функции нет, то есть функция определена во всех встречающихся точках.

Параметрами метода являются:

- коэффициент отражения $\alpha > 0$, обычно выбирается равным 1;
- коэффициент сжатия $\beta > 0$, обычно выбирается равным 0,5;
- коэффициент растяжения $\gamma > 0$, обычно выбирается равным 2;
- точность алгоритма ε.

Подготовка. Вначале задают n + 1 точки $x = (x_1, x_2, \ldots, x_n, x_{n+1})$, образующие симплекс n-мерного пространства. В этих точках вычисляются значения функции: $f_1=f(x_1)$, $f_2=f(x_2)$, …, $f_{n+1}=f(x_{n+1})$.

Дальнейший план действий.

1. Сортировка. Из вершин симплекса выбирают три точки: $x_{макс}$ с наибольшим (из имеющихся точек) значением функции, $x_{средн}$ со следующим по величине значением и $x_{мин}$ с наименьшим значением функции. Пусть $f_{макс}$, $f_{средн}$ и $f_{мин}$ – значения функций в соответствующих точках. Целью дальнейших действий будет уменьшение, по крайней мере, значения $f_{макс}$.

2. Определяют центр тяжести всех точек, за исключением $x_{макс}$:

$$x_c = \frac{1}{n} \sum_{\substack{i=1 \\ i \neq \text{макс}}}^{n} x_i. \tag{1.7}$$

Вычисляют значение функции f_c в точке x_c.

3. Отражение. Отражают точку $x_{макс}$ относительно x_c с коэффициентом α. В результате будет получена точка x_r; $f(x_r)$ – значение функции в этой точке. Координаты новой точки вычисляются по формуле:

$$x_r = (1 + \alpha)x_c - \alpha x_{\text{макс}}. \tag{1.8}$$

Операция отражения на примере треугольного симплекса приведена на *рис. 1.3*.

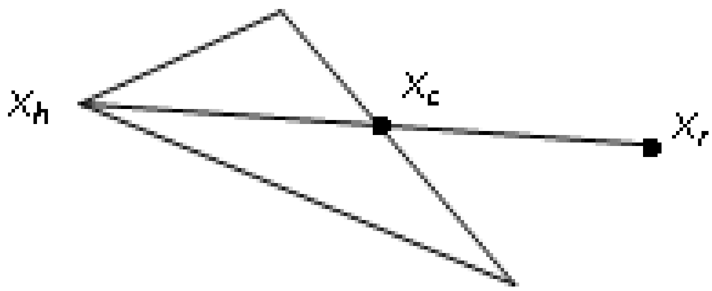

Рис. 1.4. Операция отражения на примере треугольного симплекса

4. Далее определяют, насколько удалось уменьшить функцию с помощью новой точки. Для этого проверяют ряд условий.

4.1. Если

$$f_r < f_{\text{мин}}, \tag{1.9}$$

то направление выбрано удачное, поскольку найдено наилучшее из имеющихся значений функций. В связи с этим осуществляется попытка растяжения. В результате определяется новая точка

$$x_b = (1 - \gamma)x_c + \gamma x_r. \tag{1.10}$$

Пример операции растяжения приведен на *рис. 1.4*.

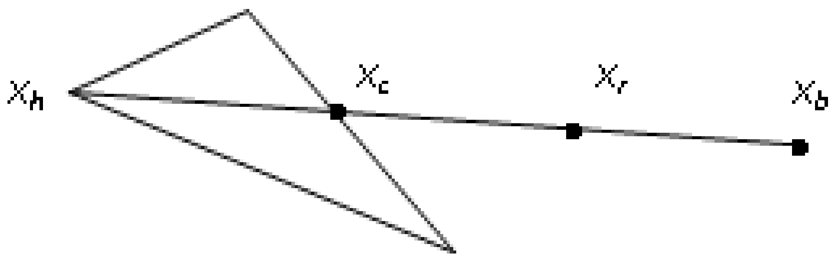

Рис. 1.5. Операция растяжения на примере треугольного симплекса

Далее определяется, насколько эффективна в данном случае процедура растяжения.

4.1.1. Если

$$f_b < f_{\text{мин}}, \tag{1.11}$$

то худшая точка заменяется точкой x_b:

$$x_{\text{мин}} = x_b. \tag{1.12}$$

На этом очередная итерация заканчивается, и осуществляется переход к п.8.

4.1.2. Если

$$f_b > f_{\text{мин}}, \quad (1.13)$$

то с учетом неравенства (1.10) на данном этапе найдена наилучшая точка x_r, поскольку именно в этой точке получено наименьшее значение функции. Поэтому наихудшую точку заменяют на x_r:

$$x_{\text{мин}} = x_r, \quad (1.14)$$

и также переходят к п.8.

4.2. Если неравенство (1.10) не выполняется (что означает, что в точке x_r не наилучшее из всех просмотренных точек значение), то проверяют следующее неравенство:

$$f_{\text{мин}} < f_r < f_{\text{средн}}. \quad (1.15)$$

Если оно выполнено, то поиск точки считается также неплохим, поскольку она лучше двух имеющихся в симплексе: $x_{\text{средн}}$ и $x_{\text{макс}}$. Поэтому осуществляется также действие (1.14), после чего – переход к п.8.

В противном случае точка считается неприемлемой, в связи с чем начинают движение в другую сторону (п. 5). Предварительно перед этим шагом отбрасывают наихудшую из точек x_r и $x_{\text{макс}}$. Для этого предварительно проверяют неравенство:

Если

$$f_{\text{макс}} > f_r, \quad (1.16)$$

то (точка x_r лучше для симплекса, чем $x_{\text{макс}}$)

$$x_{\text{макс}} = x_r. \quad (1.17)$$

Таким образом, убрали наихудшую из точек x_r и $x_{\text{макс}}$, и результат лучшей из этих точек занести в переменную $x_{\text{макс}}$. После этого переходят к процедуре сжатия (п.5).

5. Сжатие. Строят точку

$$x_s = (1 - \beta)x_c + \beta x_{\text{макс}}. \quad (1.18)$$

и вычисляют в ней значение f_s. Если операция (1.17) не выполнялась, то точка x_s будет располагаться следующим образом (*рис. 1.5*).

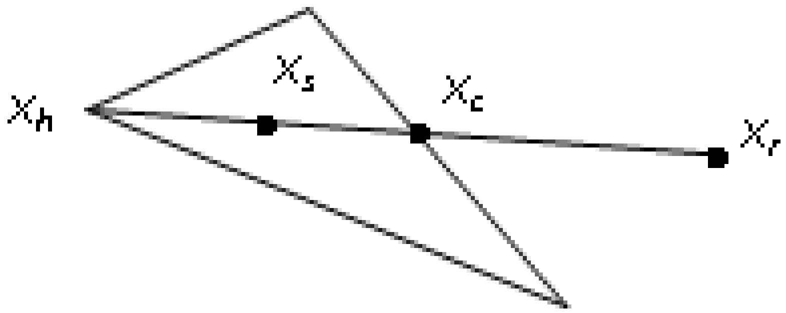

Рис. 1.6. Операция сжатия на примере треугольного симплекса (при выполнении операции (1.17))

Если операция не выполнялась, то расположение точки x_s будет следующим (*рис. 1.6*).

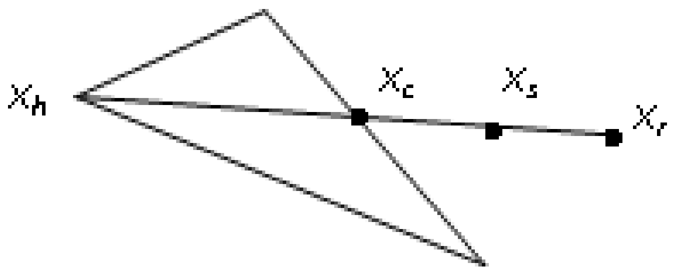

Рис. 1.7. Операция сжатия на примере треугольного симплекса (без выполнения операции (1.17))

6. Если

$$f_s < f_{\text{макс}}, \tag{1.19}$$

то

$$x_{\text{макс}} = x_s. \tag{1.20}$$

После этого переходят к п.8.

7. Если неравенство (1.19) не выполняется, то выполняют глобальное сжатие симплекса — гомотетию к точке с наименьшим значением x_l: $x_i = x_l + (x_i - x_l)/2$ для всех требуемых точек x_i (*рис. 1.7*).

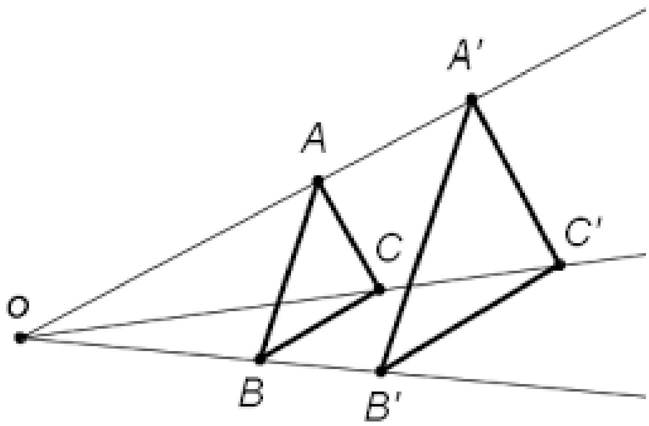

Рис. 1.8. Глобальное сжатие

8. Последний шаг — проверка сходимости. Может выполняться по-разному, например, оценкой дисперсии набора точек. Суть проверки заключается в том, чтобы проверить взаимную близость полученных вершин симплекса, что предполагает и близость их к искомому минимуму. Если требуемая точность ещё не достигнута, можно продолжить итерации с шага 1.

Пример.

Пример работы алгоритма Нелдера-Мида

Дано:

$$f(x,y) = x^2 + xy + y^2 - 6x - 9y$$

3 начальные точки V1=(0,0); V2=(1,0); V3=(0,1).

Шаг 1

Вычисляем значения функции в точках:

f(V1)=0; f(V2)=-5; f(V3)=-8.

1.Сортировка.

Переобозначим точки следующим образом:

x_h =V1; x_g =V2; x_l =V3.

2.Определение центра тяжести.

$$x_c = \frac{x_g + x_1}{2} = \left(\frac{1+0}{2}, \frac{0+1}{2}\right) = \left(\frac{1}{2}, \frac{1}{2}\right).$$

3.Отражение

$$x_r = (1+\alpha)\, x_c - \alpha x_h$$

$$\alpha = 1$$

$$x_r = 2\, x_c - x_h = 2\left(\frac{1}{2}, \frac{1}{2}\right) - (0{,}0) = (1{,}1)$$

$$f(x_r) = -12.$$

Проверка: $f(x_r)$<$f(x_1)$, следовательно, направление выбрано удачно, увеличиваем отрезок (операция растяжения):

$$x_b = (1-\gamma)\, x_c + \gamma x_r$$

$\gamma = 2.$

$x_b = -x_c + 2x_r$=- $\left(\frac{1}{2}, \frac{1}{2}\right)$+2(1,1)=$\left(\frac{3}{2}, \frac{3}{2}\right)$,

$$f(x_b) = -15.75.$$

Получили новые вершины V1=$\left(\frac{3}{2}, \frac{3}{2}\right)$ V2=(1,0); V3=(0,1).

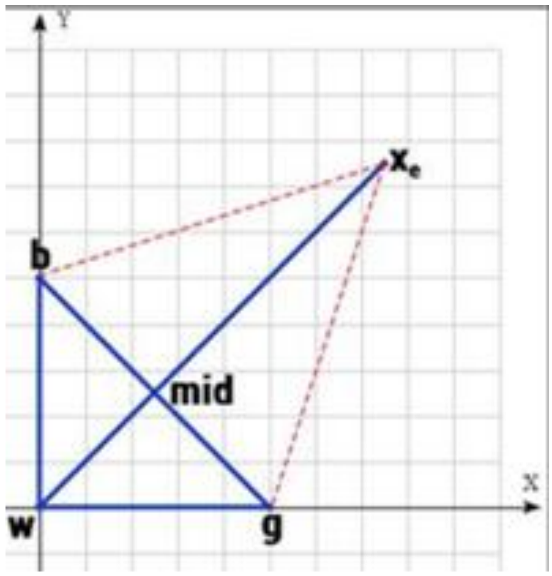

Рис. 1.9. Иллюстрация работа шага 1

Таблица 1.4

Результаты шага 1

Значение функции в лучшей точке	Значение функции в предпоследней точке	Значение функции в худшей точке
f($\frac{3}{2}$,$\frac{3}{2}$) = -15.75	f(0,1)=-8	f(1,0)=-5

<u>Шаг 2</u>

1. Сортировка. Переобозначим вершины: x_h=(1,0); x_g=(0,1); x_1=(1.5, 1.5)
2. Определение центра тяжести:
3.

$$x_c = \frac{x_g + x_1}{2} = \left(\frac{0+1.5}{2}, \frac{1+1.5}{2}\right) = (0.75{,}1.25).$$

4. Отражение:

$$x_r = 2x_c - x_h = 2(0.75{,}1.25) - (1{,}0) = (0.5{,}2.5)$$
$$f(x_r) = -17.75.$$

5. Проверка: $f(x_r)$<$f(x_1)$, следовательно, направление выбрано удачно, увеличиваем отрезок (операция растяжения):

$$x_b = -x_c + 2x_r = 2(0.5{,}2.5) - (0.75{,}1.25) = (0.25{,}3.75)$$
$$f(x_b) = -20.187.$$

6. Проверка: $f(x_b)$<$f(x_1)$, следовательно, точка (0.25,3,75) наилучшая, она будет являться заменой точке x_h.
7. Получили новые вершины V1=$\left(\frac{3}{2},\frac{3}{2}\right)$ V2=(1,0); V3=(0.25, 0.75).

Таблица 1.5

Результаты шага 2

Значение функции в лучшей точке	Значение функции в предпоследней точке	Значение функции в худшей точке
F(0.25, 3,75)=-20.1875	f($\frac{3}{2}$, $\frac{3}{2}$) = -15.75	f(0,1)=-8

Шаг 3

1. Сортировка. Переобозначим вершины: x_h=(0,1); x_g=(1.5,1.5); x_1=(0.25, 3.75)
2. Определение центра тяжести:

$$x_c = \frac{x_g + x_1}{2} = \left(\frac{0.25+1.5}{2}, \frac{3.75+1.5}{2}\right) = (0.875{,}2.625).$$

3. Отражение:

$$x_r = 2x_c - x_h = 2(0.875{,}2.625) - (0{,}1) = (1.75{,}4.25)$$
$$f(x_r) = -20.1875.$$

4. Проверка: $f(x_r)<f(x_1)$. Данное условие не выполняется ($f(x_r)=f(x_1) = -20.1875$).
5. Проверяем, лучше ли точка x_r, чем точки x_g и x_h. Исходя из табл. 3.2,

$$f(x_r) > f(x_g) > f(x_\mathrm{h})$$

Следовательно, меняем худшую точку на x_r. Результаты данного шага представлены в табл.1.6.

Таблица 1.6

Результаты шага 3

Значение функции в лучшей точке	Значение функции в предпоследней точке	Значение функции в худшей точке
F(0.25, 3,75)=-20.1875	F(1.75,4.25)=-20.1875	f($\frac{3}{2}$, $\frac{3}{2}$) = -15.75

Шаг 4

1. Сортировка. Переобозначим вершины: x_h=(1.5,1.5); x_g=(1.75, 4.25); x_1=(0.25,3.75).
2. Определение центра тяжести:

$$x_c = \frac{x_g + x_1}{2} = (1{,}4).$$

3. Отражение:

$$x_r = 2x_c - x_h = 2(1{,}4) - (1.5{,}1.5) = (0.5{,}6.5)$$
$$f(x_r) = -15.75.$$

4. Проверка: $f(x_r)<f(x_1)$. Условие не выполняется.
5. Проверка: $f_l < f_r < f_g$. Условие не выполняется.
6. Проверка: $f_h > f_r > f_g$. Условие не выполняется.
7. Проверка: $f_r > f_h$. Условие выполняется, переходим к операции сжатия. Для этого находим точку

$$x_s = \beta x_h + (1-\beta)x_c$$

$$\beta = 0.5$$
$$x_s = 0.5x_h + 0.5x_c = (1.25{,}2.75).$$

8. Вычисляем $f(x_s)$=-19.68.
9. Проверка $f_s < f_h$, следовательно, меняем x_h на x_s.

Таблица 1.7

Результаты шага 4

Значение функции в лучшей точке	Значение функции в предпоследней точке	Значение функции в худшей точке
F(1.75,4.25)=-20.1875	F(0.25, 3,75)=-20.1875	f(1.25,2.75)=-19.68

Шаг 5

1. Сортировка. Переобозначим вершины: x_h=(1.25,2.75); x_g=(0.25,3.75); x_1=(1.75, 4.25).
2. Определение центра тяжести:

$$x_c = \frac{x_g + x_1}{2} = (1{,}4).$$

3. Отражение:

$$x_r = 2x_c - x_h = 2(1{,}4) - (1.25{,}2.75) = (0.75{,}5.25)$$
$$f(x_r) = -19.6875.$$

4. Проверка: $f(x_r) < f(x_1)$. Условие не выполняется.
5. Проверка $f(x_r) < f\left(x_g\right)$. Условие не выполняется. Выполняем операцию сжатия относительно наихудшей точки:

$x_s = 0.5x_h + 0.5x_c$=(0.625,1.375)+(0.5,2)=(1.125,3.375)

$$x_s = 0.5x_h + 0.5x_c = (0.625{,}1.375) + (0.5{,}2) = (1.125{,}3.375)$$

f(1.125,3.375)=-20.6719.

6. Проверка $f_s < f_h$. Условие выполняется, следовательно, меняем x_h на x_s. Результаты пятого шага представлены в табл. 1.8.

Таблица 1.9

Результаты шага 5

Значение функции в лучшей точке	Значение функции в предпоследней точке	Значение функции в худшей точке
f(1.125,3.375)=-20.6719	F(1.75,4.25)=-20.1875	F(0.25,3.75)=-20.1875

Шаг 6

1. Сортировка. Переобозначим вершины: x_h=(0.25,3.75); x_g=(1.75,4.25); x_1=(1.125, 3.375).
2. Определение центра тяжести:

$$x_c = \frac{x_g + x_1}{2} = (1.4375{,}3.8125).$$

3. Отражение:

$$x_r = 2x_c - x_h = (2.625{,}3.875),$$
$$f(x_r) = -18.5469.$$

4. Проверка: $f(x_r)<f(x_1)$. Условие не выполняется.
5. Проверка: $f_h > f_r > f_g$. Условие не выполняется.
6. Проверка: $f_r > f_h$. Условие выполняется. Переходим к операции сжатия. Для этого формируем точку

$$x_s = 0.5x_h + 0.5x_c = (2.03125,3.84375),$$
$$f(x_s) = -20.0732.$$

Это наихудшее значение. Следовательно, выполняем глобальное сжатие в направлении значения x_1. В частности, приближаем в два раза точки (1.75,4.25) и (0.25,3.75) к точке (1.125,3.375).

Точка (1.75,4.25). Первая координата:
(1.75+1.125)/2=1.4375.
Вторая координата:
(4.25+3.375)/2=3.8125.
Точка (0.25,3.75). Первая координата:
(0.25+1.125)/2=0.6875.
Вторая координата:
(3.75+3.375)/2=3.5625.

$$f(1.4375,3.8125) = -20.8555,$$
$$f(0.6875,3.5625) = -20.5742.$$

Таблица 1.10

Результаты шага 6

Значение функции в лучшей точке	Значение функции в предпоследней точке	Значение функции в худшей точке
f(1.4375,3.8125)=-20.8555	f(1.125,3.375)=-20.6719	F(0.6875,3.5625)=-20.5742

<u>Шаг 7</u>

1. Сортировка. Переобозначим вершины: x_h=(0.6875,3.5625); x_g=(1.125,3.375); x_1=(1.4375,3.8125).
2. Определение центра тяжести:

$$x_c = \frac{x_g + x_1}{2} = (1.28125,3.59375).$$

3. Отражение:

$$x_r = 2x_c - x_h = (1.875,3.625),$$
$$f(x_r) = -20.4219.$$

4. Проверка: $f(x_r)<f(x_1)$. Условие не выполняется.
5. Проверка: $f_h > f_r > f_g$. Условие не выполняется.
6. Проверка: $f_r > f_h$. Условие выполняется. Переходим к операции сжатия. Для этого строим точку

$$x_s = 0.5x_h + 0.5x_c = (1.5781,3.6094),$$
$$f(x_s)=-20.739.$$

7. Проверка $f_s < f_h$, следовательно, меняем x_h на x_s.

Таблица 1.11

Результаты шага 7

Значение функции в лучшей точке	Значение функции в предпоследней точке	Значение функции в худшей точке
f(1.4375,3.8125)=-20.8555	f(1.125,3.375)=-20.6719	F(1.578125,3.609375)=-20.739

<u>Шаг 8</u>

1. Сортировка. Переобозначим вершины: x_h=(1.125,3.375); x_g=(1.578125,3.609375); x_1=(1.4375,3.8125).
2. Определение центра тяжести:

$$x_c = \frac{x_g + x_1}{2} = (1.5078, 3.7109).$$

3. Отражение:

$$x_r = 2x_c - x_h = (1.8906, 4.0488),$$
$$f(x_r) = -20.1628.$$

4. Проверка: $f(x_r) < f(x_1)$. Условие не выполняется.
5. Проверка: $f_h > f_r > f_g$. Условие не выполняется.
6. Проверка: $f_r > f_h$. Условие выполняется. Переходим к операции сжатия. Для этого находим точку

$$x_s = 0.5x_h + 0.5x_c = (1.6992, 3.8789),$$
$$f(x_s) = -20.5811.$$

Это наихудшее значение. Следовательно, выполняем глобальное сжатие в направлении значения x_1.
Точка (1.125,3.375). Первая координата:
(1.4375+1.125)/2=1.28125.
Вторая координата:
(3.8125+3.375)/2=3.59375.

Точка (1.578125,3.609375). Первая координата:
(1.578125+1.4375)/2=1.507813.
Вторая координата:
(3.609375+3.8125)/2=3.710938,
F(1.28125,3.59375)=-20.8701,
F(1.507813,3.710938)=-20.8054.

Таблица 1.12

Результаты шага 8

Значение функции в лучшей точке	Значение функции в предпоследней точке	Значение функции в худшей точке
F(1.28125,3.59375)=-20.8701	f(1.4375,3.8125)=-20.8555	F(1.507813,3.710938)=-20.8054

Шаг 9

1. Сортировка. Переобозначим вершины: x_h=(1.507813,3.710938); x_g=(1.4375,3.8125); x_1=(1.28125,3.59375).
2. Определение центра тяжести:

$$x_c = \frac{x_g + x_1}{2} = (1.3594,3.7031).$$

3. Отражение:

$$x_r = 2x_c - x_h = (1.210938,3.695313),$$
$$f(x_r) = -20.9269.$$

4. Проверка: $f(x_r) < f(x_1)$. Условие выполняется. Следовательно, направление выбрано удачно, увеличиваем отрезок (операция растяжения):

$x_b = (1-\gamma)\, x_c + \gamma x_r$

$$\gamma = 2$$

$x_b = -x_c + 2x_r$=(1.0625,3.6875)

f(x_b)=-20.918

5. Проверка f(x_b)<f(x_r). Условие не выполняется. Следовательно, меняем x_h на x_r.

Таблица 1.13

Результаты шага 9

Значение функции в лучшей точке	Значение функции в предпоследней точке	Значение функции в худшей точке
f(1.210938,3.695313)=-20.9269	F(1.28125,3.59375)=-20.8701	f(1.4375,3.8125)=-20.8555

Данный процесс продолжается до тех пор, пока не будет выполнено условие остановки.

Условие остановки

Метод завершает свою работу в случае, если полученный симплекс имеет величину меньше заданной точности.

Найдем следующие значения (в общем случае для n+1 точки):

Для координаты i:

1. Рассчитаем среднее значение по всем точкам:

$$\overline{x_i} = \frac{1}{n+1} \sum_{j=1}^{n+1} x_i^j. \qquad (1.21)$$

2. Определим дисперсию:

$$\sigma_i = \frac{1}{(n+1)} \sum_{j=1}^{n+1} \left(x_j^i - \overline{x_i}\right)^2. \qquad (1.22)$$

Приведем пример расчета для точек из последней таблицы. Имеем три точки (1.210938,3.695313), (1.28125,3.59375) и (1.4375,3.8125). Ищем среднее по первой координате:

$$\overline{x_1} = \frac{1}{3}(1.210938 + 1.28125 + 1.4374) = 1.309896,$$

$$\sigma_1 = \frac{1}{3}((1.210938 - 1.309896)^2 + (1.28125 - 1.309896)^2 +$$

$$1.4374 - 1.3098962 = 0.008965.$$

Аналогичным образом рассчитываются значения σ по остальным координатам (в данном случае, по второй координате).

Если все эти значения меньше заданной точности ε, то метод завершает свою работу.

1.4. Методы решения задач первого порядка. Градиентные методы

Градиентный метод относится к классическим численным методам безусловной оптимизации. Идея метода заключается в замене минимизируемой функции в окрестности очередной точки x^k первыми членами ее разложения в ряд Тейлора. В градиентном методе используется линейная часть разложения.

Общая схема метода:

$$x^{k+1} = x^k + \alpha_k h^k. \tag{1.23}$$

Вектор h^k определяет направление (k+1)-го шага оптимизации, а коэффициент α_k - длину этого шага.

В градиентном методе h^k берется равным антиградиенту функции f в точке x^k, т.е.

$$h^k = -f'(x^k). \tag{1.24}$$

Таким образом, итерационная формула градиентных методов имеет вид:

$$x^{k+1} = x^k - \alpha_k f'(x^k). \tag{1.25}$$

В градиентном методе используются разные способы выбора длины шага. Так, если длина шага выбирается из условия минимизации функции вдоль направления антиградиента, то получаем вариант метода, который называется методом наискорейшего спуска.

Дробление шага

Если h^k – направление убывания, то дробление шага можно осуществлять следующим образом.

Выбираются некоторые константы $\beta>0$; $0< \lambda<1$. Для коэффициента $\alpha=\beta$ проверяется выполнение условия:

$$f(x^k + \alpha_k h^k) < f(x^k). \tag{1.26}$$

Если оно выполнено, то предполагают $\alpha_k=\alpha$, если нет, то производится дробление шага:

$$\alpha=\lambda\beta. \tag{1.27}$$

и вновь проверяется выполнение условия (1.26). Процесс дробления продолжается до тех пор, пока условие (1.26) не станет истинным.

Сходимость градиентного метода

Сходимость градиентного метода зависит от специфики целевой функции. В частности, рассматривается два случая: если функция f выпукла и если она таковой не является.

1) Случай невыпуклой функции

В случае, когда минимизируемая функция f не предполагается выпуклой, градиентный метод может обеспечить сходимость к множеству стационарных точек функции f.

Теорема. Пусть функция f дифференцируема и ограничена снизу на R^n, а ее градиент удовлетворяет условию Липшица:

$$\left\|f'(x_2)-f'(x_1)\right\| \le M\|x_2-x_1\|. \tag{1.28}$$

Тогда если выполняются условия

$$f\left(x^k+\alpha_k f'(x^k)\right)-f(x^k) \le \varepsilon\alpha_k\left\|f'(x^k)\right\|, \tag{1.29}$$

$$\alpha_k \ge min\left(\beta, \frac{\lambda(1-\varepsilon)}{M}\right), \tag{1.30}$$

то при произвольной начальной точке x^0 для метода (1.25) имеем

$$\lim_{k\to\infty}\left\|f'(x^k)\right\|=0. \tag{1.31}$$

Доказательство приведено в [4].

Условие (1.29) означает, что любая предельная точка последовательности $\{x^k\}$, генерируемая градиентным методом, является стационарной точкой функции f. Однако возможны случаи, когда эта точка не является точкой минимума.

2) Случай сильно выпуклой функции

Пусть f дважды дифференцируема и сильно выпукла на R^n, а ее матрица вторых производных удовлетворяет условию:

$$\langle f''(x)h,h\rangle \le D\|h\|^2, \tag{1.32}$$

где

$$\alpha_k \ge \min\left(\beta, \frac{2\lambda(1-\varepsilon)}{D}\right). \tag{1.33}$$

Тогда при произвольной начальной точке x^0 последовательности $\{x^k\}$, генерируемая градиентным методом, сходится к точке минимума x^* со скоростью геометрической прогрессии:

$$f(x^k) - f(x^*) \le q^k\, f(x^0) - f(x^*); \tag{1.34}$$

$$\|x^k - x^*\| \le C\left(\sqrt{q}\right)^k. \tag{1.35}$$

Далее рассмотрим примеры применения градиентного метода для решения оптимизационных задач.

Пример. С помощью градиентного метода решить задачу безусловной оптимизации:

$$f(x_1, x_2) = 3x_1^2 - 4x_1 + x_2^2 - x_1 x_2 \to min$$

В качестве начального приближения взять точку (-2,3).

Точное решение задачи (8/11, 4/11)=(0,36, 0,72).

$$\frac{\partial F}{\partial x_1} = 6x_1 - x_2 - 4$$

$$\frac{\partial F}{\partial x_2} = -x_1 + 2x_2$$

Расчетные формулы:

$$x_1^k = x_1^{k-1} - \lambda_k \frac{\partial F}{\partial x_1}$$

$$x_2^k = x_2^{k-1} - \lambda_k \frac{\partial F}{\partial x_2}$$

А). Решим задачу с **<u>постоянным</u>** шагом.

Здесь важной задачей является выбор шага.

Пусть, например, выбран шаг λ=0.5

Расчеты приведены в следующей таблице:

Таблица 1.14

Расчеты градиентного метода с шагом 0.5

x1	x2	f(x1,x2)	df/dx1	df/dx2
-2	3	35	-19	8
7,5	-1	147,25	42	-9,5
-13,5	3,75	665,4375	-88,75	21
30,875	-6,75	2990,266	188	-44,375
-63,125	15,4375	13419,61	-398,188	94
135,9688	-31,5625	60206,33	843,375	-199,094
-285,719	67,98438	270094,8	-1786,3	421,6875
607,4297	-142,859	1211669	3783,438	-893,148
-1284,29	303,7148	5435633	-8013,45	1891,719

2722,436	-642,145	24384623	16972,76	-4006,72

Очевидно, что итерационный процесс будет расходиться.

Изменим шаг до 0.1. Аналогичным образом будем считать производные в полученных точках и по формуле (4.2) искать значение в следующей точке.

Результаты будут следующие:

Таблица 1.15

Расчеты градиентного метода с шагом 0.1

x1	x2	f(x1,x2)	df/dx1	df/dx2
-2	3	35	-19	8
-0,1	2,2	5,49	-6,8	4,5
0,58	1,75	0,7367	-2,27	2,92
0,807	1,458	-0,3251	-0,616	2,109
0,8686	1,2471	-0,73897	-0,0355	1,6256
0,87215	1,08454	-0,97632	0,14836	1,29693
0,857314	0,954847	-1,13117	0,189037	1,05238
0,83841	0,849609	-1,23533	0,180853	0,860808
0,820325	0,763528	-1,30587	0,158422	0,706731
0,804483	0,692855	-1,3537	0,134042	0,581227
0,791079	0,634732	-1,38614	0,111739	0,478386
0,779905	0,586894	-1,40814	0,092534	0,393883
0,770651	0,547505	-1,42307	0,076402	0,32436
0,763011	0,515069	-1,43319	0,062997	0,267128
0,756711	0,488357	-1,44006	0,051912	0,220002
0,75152	0,466356	-1,44472	0,042765	0,181193
0,747244	0,448237	-1,44788	0,035225	0,149231
0,743721	0,433314	-1,45002	0,029013	0,122907
0,74082	0,421023	-1,45148	0,023896	0,101227
0,73843	0,410901	-1,45247	0,019681	0,083371
0,736462	0,402564	-1,45313	0,01621	0,068665
0,734841	0,395697	-1,45359	0,01335	0,056553
0,733506	0,390042	-1,4539	0,010995	0,046577
0,732407	0,385384	-1,45411	0,009056	0,038361
0,731501	0,381548	-1,45425	0,007459	0,031595
0,730755	0,378388	-1,45434	0,006143	0,026022
0,730141	0,375786	-1,45441	0,005059	0,021432
0,729635	0,373643	-1,45445	0,004167	0,017651
0,729218	0,371878	-1,45448	0,003432	0,014538
0,728875	0,370424	-1,4545	0,002827	0,011973
0,728592	0,369227	-1,45452	0,002328	0,009861
0,72836	0,368241	-1,45453	0,001917	0,008122
0,728168	0,367429	-1,45453	0,001579	0,006689

Получили решение с заданной точностью за 34 итерации.

Б) Решение задачи с **дроблением шага**.

Основным недостатком постоянного шага для градиентного метода является невозможность гарантировать его сходимость. В частности, при больших значениях λ метод может расходиться, а при маленьких заведомо будет тратиться очень большое количество операций для вычислительного поиска. Решить данную проблему можно, используя дробление шага.

Как было отмечено ранее, для процедуры дробления шага необходимо сначала задать его первоначальное (достаточно большое) значение. Если значение функции, найденное на следующей итерации с заданным шагом оказывается больше значения функции на текущей итерации, шаг

Рассмотрим значения, полученные в результате выполнения первого шага:

$$x_1^1 = -2 - 1 \cdot (-19) = 17,$$
$$x_1^2 = 3 - 1 \cdot 8 = -5.$$

Получили новую точку (17, -5). Вычислим значение функции в данной точке:

$$f(x_1, x_2) = 3x_1^2 - 4x_1 + x_2^2 - x_1x_2 = 909.$$

Поскольку значение функции в этой точке существенно превосходит значение в исходной точке (-2,3), уменьшим λ вдвое.

$$x_1^1 = -2 - 0.5 \cdot (-19) = 7.5,$$
$$x_1^2 = 3 - 0.5 \cdot 8 = -1,$$
$$f(x_1, x_2) = 147.25.$$

Это значение также больше, чем значение 35, полученное в начальной точке. Следовательно, необходимо еще уменьшить значение λ:

$$x_1^1 = -2 - 0.25 \cdot (-19) = 2.75,$$
$$x_1^2 = 3 - 0.25 \cdot 8 = 1,$$
$$f(x_1, x_2) = 9.9375.$$

Поскольку это значение меньше, чем 35, оставляем точку (2.75,1) как результат первого шага. Вычисляем производную в данной точке:

$$\left.\frac{\partial F}{\partial x_1}\right|_{x_1=-2.75;\, x_2=1} = 6 \cdot (2.75) - 1 - 4 = 11.5,$$
$$\left.\frac{\partial F}{\partial x_2}\right|_{x_1=-2.75;\, x_2=1} = -2.75 + 2 \cdot 1 = -0.75.$$

Определяем новую точку по формуле:

$$x_1^1 = 2.75 - 0.25 \cdot (11.5) = -0.125,$$
$$x_1^2 = 3 - 0.25 \cdot (-0.75) = 1.1875.$$

Вычисляем значение функции в данной точке:

$$f(x_1, x_2) = 2{,}105469.$$

Так как это значение меньше предыдущего (9.9375), то шаг λ=0.25 оставляем и переходим к следующему шагу.

Результаты сведены в следующую таблицу.

Таблица 1.16

Расчеты градиентного метода с дроблением шага

	x1	x2	f(x1,x2)	df/dx1	df/dx2	alpha
нач усл	-2	3	35	-19	8	1
шаг 1	17	-5	909			0,5
	7,5	-1	147,25			0,25
	2,75	1	9,9375	11,5	-0,75	0,25
шаг 2	-0,125	1,1875	2,105469	-5,9375	2,5	0,25
шаг 3	1,359375	0,5625	-0,34204	3,59375	-0,23438	0,25
шаг 4	0,460938	0,621094	-1,10689	-1,85547	0,78125	0,25
шаг 5	0,924805	0,425781	-1,3459	1,123047	-0,07324	0,25
шаг 6	0,644043	0,444092	-1,42059	-0,57983	0,244141	0,25
шаг 7	0,789001	0,383057	-1,44394	0,350952	-0,02289	0,25
шаг 8	0,701263	0,388779	-1,45123	-0,1812	0,076294	0,25
шаг 9	0,746563	0,369705	-1,45351	0,109673	-0,00715	0,25
шаг 10	0,719145	0,371493	-1,45422	-0,05662	0,023842	0,25
шаг 11	0,733301	0,365533	-1,45444	0,034273	-0,00224	0,25
шаг 12	0,724733	0,366092	-1,45451	-0,0177	0,007451	0,25
шаг 13	0,729157	0,364229	-1,45454	0,01071	-0,0007	0,25

Как можно видеть из данной таблицы, для нахождения решения потребовалось гораздо меньше времени, чем в предыдущем случае.

В). Рассмотрим решение задачи методом **скорейшего спуска**

Пусть x^0=(-2,3) .

ШАГ 1

$$\frac{\partial F}{\partial x_1} = -19$$

$$\frac{\partial F}{\partial x_2} = 8$$

$$x_1^1 = -2 + \lambda * 19$$

$$x_2^1 = 3 - \lambda * 8$$

Найдем λ, исходя из требования минимизации функции F

$$F(x_1, x_2) = 3(-2 + \lambda * 19)^2 \ - 4(-2 + \lambda * 19) + (3 - \lambda * 8)_2^2 + (-2 + \lambda * 19) * (3 - \lambda * 8)$$

$$\frac{\partial F}{\partial \lambda} = 6(-2 + \lambda * 19) * 19 - 4 * 19 + 2(3 - \lambda * 8) * (-8) + 19 * (3 - \lambda * 8) - 8 * (-2 + \lambda * 19) = 0$$

2598λ-425=0

λ=0.164

$$x_1^1 = -2 + 0.164 * 19 = 1.116$$

$$x_2^1 = 3 - 0.164 * 8 = 1.688$$

F=0.23

ШАГ 2

$$\frac{\partial F}{\partial x_1} = 6x_1 - x_2 - 4 = 1.008$$

$$\frac{\partial F}{\partial x_2} = -x_1 + 2x_2 = 2.26$$

$$x_1^2 = 1.116 - \lambda * 1.008$$

$$x_2^2 = 1.688 - \lambda * 2.26$$

$$F(x_1, x_2) = 3(1.116 - \lambda * 1.008)^2 \ - 4(1.116 - \lambda * 1.008) + (1.688 - \lambda * 2.26)_2^2 + (1.116 - \lambda * 1.008) * (1.688 - \lambda * 2.26)$$

$\frac{\partial F}{\partial \lambda} = 6(1.116 - \lambda * 1.008) * (-1.008) + 4 * 1.008 + 2(1.688 - \lambda * 2.26) * (-2.26) + 1.008 * (1.688 - \lambda * 2.26) + 2.26 * (1.116 - \lambda * 1.008)$ =0

11,45λ-6,58546=0

λ=0.57513

$$x_1^2 = 1.116 - 0.57513 * 1.008 = 0{,}536269$$

$$x_2^2 = 1.688 - 0.67499 * 2.26 = 0{,}3882$$

1.5 Методы решения задач второго порядка. Метод Ньютона и его модификации

Предположим, что функция f выпукла и дважды дифференцируема. По определению дважды дифференцируемой функции для очередной точки x^k имеем:

$$f(x) - f(x^k) = \langle f'(x^k), x - x^k \rangle + \frac{1}{2}\langle f''(x^k)(x - x^k), x - x^k \rangle +$$
$$+O\left(\|x - x^k\|^2\right). \qquad (1.36)$$

Для определения следующей точки x^{k+1} минимизируется функция $f_k(x)$, являющаяся квадратичной частью приращения $f(x)$-$f(x^k)$, т.е. решается задача:

$$f_k(x) = \langle f'(x^k), x - x^k \rangle + \frac{1}{2}\langle f''(x^k)(x - x^k), x - x^k \rangle \to min. \qquad (1.37)$$

Необходимое и достаточное условие минимума для метода Ньютона имеет вид:

$$f_k(x) = f'(x^k) + f''(x^k)(x - x^k) = 0. \qquad (1.38)$$

Решая полученную систему уравнений и принимая найденную точку минимума за x^k, получаем:

$$x^{k+1} = x^k + h^k, \qquad (1.39)$$

где

$$h^k = -\left(f''(x^k)\right)^{-1} \cdot f'(x^k). \qquad (1.40)$$

Формулы (1.39) и (1.40) и определяют метод Ньютона

Сходимость метода Ньютона и оценка скорости сходимости

Пусть функция f дважды дифференцируема, сильно выпукла с константой $\theta>0$ и удовлетворяет условию:

$$\left\|f''(x_2) - f''(x_1)\right\| \le M\|x_2 - x_1\|, \qquad (1.41)$$

где $M>0$, а начальная точка x^0 такова, что

$$\|f'(x^0)\| = \frac{8q\theta^2}{M}, 0 < q < 1. \qquad (1.42)$$

Тогда последовательность x^k сходится к точке минимума x^* с квадратичной скоростью:

$$\left\|x^k - x^*\right\| \le 4\theta q^{2^k}/M, \qquad (1.43)$$

где $0<q<1$.

Пример. Рассмотрим применение метода Ньютона на том же примере, что и при использовании градиентных методов. Необходимо найти точку минимума функции

$$f(x_1, x_2) = 3x_1^2 - 4x_1 + x_2^2 - x_1x_2 \rightarrow min.$$

Найдем первые производные:

$$\frac{\partial F}{\partial x_1} = 6x_1 - x_2 - 4;$$

$$\frac{\partial F}{\partial x_2} = -x_1 + 2x_2.$$

На их основании найдем вторые производные по каждой переменной:

$$\frac{\partial^2 F}{\partial x_1^2} = 6$$

$$\frac{\partial^2 F}{\partial x_1 \partial x_2} = -1$$

$$\frac{\partial^2 F}{\partial x_1 \partial x_2} = -1$$

$$\frac{\partial^2 F}{\partial x_2^2} = 2$$

Таким образом, матрица вторых производных имеет вид:

$$F'' = \begin{pmatrix} 6 & -1 \\ -1 & 2 \end{pmatrix}.$$

Теперь найдем обратную матрицу. Она вычисляется по формуле:

$$A^{-1} = \frac{1}{|A|} A^T. \tag{1.44}$$

Подставив в (1.44) значения производных, получим:

$$|A| = 6*2-(-1)*(-1)=11.$$

$$A^T = \begin{pmatrix} 2 & -1 \\ -1 & 6 \end{pmatrix}.$$

$$A^{-1} = \begin{pmatrix} 2/11 & 1/11 \\ 1/11 & 6/11 \end{pmatrix}.$$

Расчетные формулы метода Ньютона имеют вид:

$$X^{k+1} = X^k - \begin{pmatrix} 2/11 & 1/11 \\ 1/11 & 6/11 \end{pmatrix} * \begin{pmatrix} 6x_1 - x_2 - 4 \\ -x_1 + 2x_2 \end{pmatrix}. \tag{1.45}$$

Пусть x^0=(-2,3) .

<u>ШАГ 1</u>

$$x_1^1 = -2 - \frac{2}{11}(6 * (-2) - 3 - 4) - \frac{1}{11}(2 + 2 * 3) = 0.7272.$$
$$x_2^1 = 3 - \frac{1}{11}(6 * (-2) - 3 - 4) - \frac{6}{11}(2 + 2 * 3) = 0{,}3636.$$

Таким образом, в методе Ньютона за одну итерацию получили результат, который с помощью методов первого порядка можно получить лишь за несколько десятков шагов.

Метод Ньютона с регулировкой шага

Регулировка шага может еще ускорить получение решения методом Ньютона. Общий вид расчетных формул следующий:

$$x^{k+1} = x^k + \alpha_k h^k, \tag{1.46}$$
$$h^k = -\left(f''(x^k)\right)^{-1} \cdot f'(x^k). \tag{1.47}$$

Выбор коэффициентов α_k производится обычно или из условия минимизации функции вдоль заданного направления, или с помощью метода дробления шага, обеспечивающего выполнение условия:

$$f(x^k + \alpha_k h^k) - f(x^k) \leq \varepsilon \alpha_k \langle f'(x^k), h^k \rangle. \tag{1.48}$$

Таким образом, с увеличением порядка метода, как правило, повышается его сходимость, но увеличивается сложность.

Глава 2 Численные методы решения задачи на условный экстремум

Глава посвящена исследованию численных методов нахождения решений оптимизационной задачи в случае, если к целевой функции добавлен ряд условий, составляющих ограничения задачи. Такие задачи называются задачами на условный экстремум. В первой части главы приведена общая постановка такой задачи. Метод штрафных функций описан во второй части главы. Треться часть посвящена методу барьерных функций.

2.1. постановка задачи на условный экстремум и общая идея методов ее решения

Рассматривается решение задачи на условный экстремум:

$$\begin{cases} f(x) \to min \\ g_i(x) \le 0, i = 1, \dots, M. \end{cases} \tag{2.1}$$

Отличием от задач, рассматриваемых ранее, является наличие дополнительных ограничений, которым должна удовлетворять найденная точка минимума. Иными словами, необходимо найти минимум в области, заданной ограничениями задачи (2.1).

Очевидно, что найденная точка может не являться точкой минимума целевой функции.

Основная идея метода большинства численных методов решения задачи (2.1) заключается в переходе от задачи условной оптимизации к задаче безусловной оптимизации путем изменения исходной целевой функции за счет ограничений. В частности, к ней можно прибавлять некоторый штраф, который будет увеличиваться по мере удаления от точки минимума или барьер, не позволяющий выйти из области ограничений. Наиболее распространенными методами решения данной задачи являются методы штрафных и барьерных функций. Рассмотрим их более подробно.

2.2. Метод штрафных функций

В методе штрафных функций к целевой функции добавляется штраф, который мал (или равен нулю) если ограничения выполнены (или выполнены с точностью до ε). В случае невыполнения ограничений штраф возрастает, причем, чем дальше найденная точка от допустимой области, тем больше величина штрафа.

$$F(x, r) = f(x) + \sum_{i=1}^{M} P_i(x, r). \tag{2.2}$$

Здесь функция P_i отвечает на штраф, налагаемый в результате ограничения i, а коэффициент r регулирует размер штрафа. Таким образом, каждое из ограничений – это отдельное слагаемое штрафа.

Существуют следующие виды штрафов:

1. <u>Квадратичный штраф</u> используется для учета ограничений равенств:

$$P_i(x, r^k) = r^k\big(g_i(x)\big)^2 \quad (2.3)$$

Очевидно, что если искомая точка будет удовлетворять ограничению, то штраф будет равен нулю.

2. Штраф типа квадрата срезки:

$$P_i(x, r^k) = r^k\big(g_i^+(x)\big)^2, \quad (2.4)$$

где $g_j^+(x)$ - срезка функции:

$$g_i^+(x) = max\{0, g_i(x)\}^2 = \begin{cases} g_i(x), & \text{если } g_i(x) > 0; \\ 0, & \text{если } g_i(x) \le 0. \end{cases} \quad (2.5)$$

Из условия (2.5) непосредственно следует, что и в данном случае при выполнении ограничений штраф будет равен нулю, а в случае их нарушения будет увеличиваться пропорционально квадрату расстояний до допустимой области.

3. Бесконечный барьер является простейшим штрафом и используется в виде:

$$P_i(x, r^k) = \begin{cases} +\infty \cdot g_i(x), \text{ если } g_i(x) > 0; \\ 0, \text{ если } g_i(x) \le 0. \end{cases} \quad (2.6)$$

Алгоритм метода штрафных функций

Предварительный этап: составить штрафную функцию и новую целевую функцию F

Задать начальную точку расчета x^0, рассчитать частные производные по каждому аргументу в этой точке.

Задать начальное значение r (например, 0.01)

Этап i (i=1,2,…). Каким-либо численным методом решения задач безусловной оптимизации найти минимум x^i функции F

Если величина штрафа меньше заданной точности,

То выдать x^i в качестве ответа

Иначе увеличить r в «С» раз и перейти к этапу i+1.

Замечания.

1. Начальная точка, как правило, берется такой, чтобы ограничения нарушались (поэтому данный метод иногда называют методом внешних точек).
2. Значение С целесообразно брать от 4 до 10, постепенно, тем самым, наращивая r, и, как следствие, вклад штрафа в результирующую функцию.

Рассмотрим примеры.

1. $f(x_1, x_2) = 4x_1^2 + 4x_1 + x_2^2 - 8x_2 + 5 \to min$

$$2x_1 - x_2 = 6$$

Решим сначала аналитическим методом данную задачу без дополнительного условия.

$$f(x_1, x_2) = 4x_1^2 + 4x_1 + x_2^2 - 8x_2 + 5 \rightarrow min$$
$$\frac{\partial F}{\partial x_1} = 8x_1 + 4$$
$$\frac{\partial F}{\partial x_2} = 2x_2 - 8$$

Приравняв производные к нулю, получим: **x_1=-0.5; x_2=4.** Несложно проверить (с помощью вторых производных), что найденная точка является точкой минимума.

Теперь получим аналитическое решение с учетом дополнительного условия. Воспользуемся методом множителей Лагранжа.

$$L(x_1, x_2, \lambda) = 4x_1^2 + 4x_1 + x_2^2 - 8x_2 + 5 + \lambda(2x_1 - x_2 - 6)$$

Найдем производные по переменным x_1, x_2 и λ.

$$\frac{\partial F}{\partial x_1} = 8x_1 + 4 + 2\lambda$$
$$\frac{\partial F}{\partial x_2} = 2x_2 - 8 - \lambda$$
$$\frac{\partial F}{\partial \lambda} = 2x_1 - x_2 - 6$$

Решим систему:

$$\begin{cases} 8x_1 + 4 + 2\lambda = 0 \\ 2x_2 - 8 - \lambda = 0 \\ 2x_1 - x_2 - 6 = 0 \end{cases}$$

Получим ответ: **x_1=9/4; x_2=-3/2**

Таким образом, видно, что добавление дополнительного ограничения существенно меняет точку минимума.

Представим теперь численное решение данной задачи с помощью метода штрафных функций. В качестве штрафа выберем следующий:

$$P(r) = r(2x_1 - x_2 - 6)^2$$

Тогда новая целевая функция с учетом штрафа будет иметь вид:

$$F(x_1, x_2) = 4x_1^2 + 4x_1 + x_2^2 - 8x_2 + 5 + r(2x_1 - x_2 - 6)^2.$$

Решение задачи градиентным методом с изменяющимся шагом

<u>Этап 1.</u> r=0.01 (вклад штрафа в целевую функцию крайне мал).

$$F(x_1, x_2) = 4x_1^2 + 4x_1 + x_2^2 - 8x_2 + 5 + 0{,}01(2x_1 - x_2 - 6)^2$$

1.1 Выберем начальную точку (0,0) и будем решать задачу градиентным методом:

$$\frac{\partial F}{\partial x_1} = 8x_1 + 4 + 0{,}02(2x_1 - x_2 - 6)2 = 3{,}76$$
$$\frac{\partial F}{\partial x_2} = 2x_2 - 8 + 0{,}02(2x_1 - x_2 - 6)(-1) = -7{,}88$$
$$x_1^1 = 0 - \lambda * 3.76$$
$$x_2^1 = 0 + \lambda * 7.88$$

Возьмем начальное значение λ=1. Если значение новой функции будет меньше, чем значение исходной функции, значит, данный шаг приемлем – оставляем λ, в противном случае – уменьшаем его вдвое. Результаты представлены в следующей таблице.

Таблица 2.1

Результаты первого этапа метода штрафных функций

шаг	X_1	X_2	$F(x_1, x_2)$	$\frac{\partial F}{\partial x_1}$	$\frac{\partial F}{\partial x_2}$	λ
0	0	0	0,36	3,76	-7,88	1
1.1	-3,76	7,88	45,1444			0.5
1.2	-1,88	3,94	-7,5019	-11,588	0,154	0.5
2.1	3,914	3,863	60,99377			0,25
2.2	1,017	3,9015	-7,1662			0,125
2.3	-0,4315	3,92075	-15,812	0,11665	0,0572	0,125
3.1	-0,4461	3,9136	-15,8133	-0,0009	0,0433	0,125
4.1	-0,446	3,90819	-15,8135	0,00023	0,0324	0,125

Завершен первый этап данного метода (значения x на шаге 4.1 отличаются от значений на шаге 3.1 на величину, меньше 0.01). Получили ответ на первой итерации:

$$X_1=-0,446;\ x_2=3,90819$$

Данный ответ крайне близок к аналитическому значению, полученному без учета ограничений задачи (**x_1=-0.5; x_2=4**). Проверим, чему равны ограничения:

$$2x_1 - x_2 - 6 = -10.8$$

Объясним данный результат. В сформированной функции величина штрафа пока незначительна. Поэтому был найден глобальный минимум функции безотносительно ограничений, поскольку, по-видимому, значение в данной точке существенно меньше минимума, который достигается с учетом ограничений.

Этап 2.

Поскольку этот результат крайне далек от требуемого (ограничения нарушены), поэтому увеличиваем r в 10 раз (что увеличит штраф за отклонение ограничений в целевой функции) и, начиная с найденной точки повторяем итерации метода.

r=0.1

$$F(x_1,x_2) = 4x_1^2 + 4x_1 + x_2^2 - 8x_2 + 5 + 0,1(2x_1 - x_2 - 6)^2$$

$$\frac{\partial F}{\partial x_1} = 8x_1 + 4 + 0,1 * 2 * (2x_1 - x_2 - 6) * 2$$

$$\frac{\partial F}{\partial x_2} = 2x_2 - 8 + 0,1 * 2 * (2x_1 - x_2 - 6)(-1)$$

Таблица 2.2

Результаты второго этапа метода штрафных функций

шаг	X_1	X_2	$F(x_1, x_2)$	$\frac{\partial F}{\partial x_1}$	$\frac{\partial F}{\partial x_2}$	λ
II	-0,446	3,90819	-5,3156	-3,8878	1,97640	1
1.1	3,44185	1,93179	49,540			0,5
1.2	1,49794	2,91999	3,643			0,25
2.1	0,52598	3,41409	-5,454	4,86303	0,5006	0,25
2.2	-0,68977	3,288938	-4,9687			0,125
3	-0,08189	3,351513	-6,8261	-0,46127	0,6061	0,125
4	-0,02424	3,275752	-6,8759	0,076432	0,4163	0,125
5	-0,03379	3,223708	-6,8952	0,013174	0,3057	0,125
6	-0,03544	3,185499	-6,9053	0,013966	0,2223	0,125
7	-0,03718	3,157715	-6,9106	0,009717	0,1618	0,125
8	-0,0384	3,137484	-6,9135	0,007121	0,1178	0,125
9	-0,03929	3,122756	-6,915	0,005179	0,0858	0,125
10	-0,03993	3,112034	-6,9158	0,003771	0,0624	0,125
11	-0,04041	3,104228	-6,9162	0,002745	0,0455	0,125
12	-0,04075	3,098545	-6,9164			

Завершили второй этап метода. Ответ на втором этапе

$$X_1=-0,04075;\ x_2=3,098545$$

Штраф равен -9,5626.

По сравнению с этапом 1, штраф не сильно уменьшился. Поэтому опять увеличиваем r в 10 раз и повторяем итерационную процедуру:

<u>Этап 3.</u> r=1

Выполняя аналогичные действия с r=1, получим (табли.2.3).

Таблица 2.3

Результаты второго этапа метода штрафных функций

шаг	X_1	X_2	$F(x_1, x_2)$	$\frac{\partial F}{\partial x_1}$	$\frac{\partial F}{\partial x_2}$	λ
	-0,033789	3,223708	70,80004	-33,4355	17,02999	1
1.1	33,401667	-13,8063	10463,95			0,5
1.2	16,683939	-5,29129	2317,1			0,25
1.3	8,3250751	-1,03379	456,3813			0,125
1.4	4,1456431	1,09496	79,19846			0,0625
2.1	2,0559271	2,159334	28,9012	4,257497	4,413628	
3.1	1,7898335	1,883482	26,97582	1,103407	4,374595	0,0625
4.1	1,7208706	1,61007	25,8158	1,093649	3,556798	0,0625
5.1	1,6525175	1,38777	25,0258	0,8892	2,941011	0,0625
6.1	1,5969426	1,203957	24,48721	0,735253	2,428058	0,0625
7.1	1,5509893	1,052203	24,12001	0,607015	2,004857	0,0625

Окончание табл. 2.3

8.1	1,5130509	0,9269	23,86967	0,501214	1,655396	0,0625
9.1	1,481725	0,823438	23,69899	0,413849	1,366851	0,0625
10.1	1,4558594	0,738009	23,58263	0,341713	1,1286	0,0625
11.1	1,4345024	0,667472	23,50329	0,28215	0,931878	0,0625
12.1	1,416868	0,60923	23,44921	0,23297	0,769446	0,0625
13.1	1,4023074	0,561139	23,41233	0,192362	0,635327	0,0625
14.1	1,3902848	0,521431	23,38719	0,158832	0,524586	0,0625
15.1	1,3803578	0,488645	23,37005	0,131146	0,433147	0,0625
16.1	1,3721612	0,461573	23,35837	0,108287	0,357647	0,0625
17.1	1,3653932	0,43922	23,3504	0,089412	0,295307	0,0625
18.1	1,359805	0,420763	23,34497	0,073827	0,243833	0,0625
19.1	1,3551908	0,405524	23,34127	0,060958	0,201332	0,0625
20.1	1,3513809	0,39294	23,33874	0,050333	0,166238	0,0625
21.1	1,3482351	0,382551	23,33702	0,04156	0,137262	0,0625
22.1	1,3456376	0,373972	23,33585	0,034315	0,113336	0,0625
23.1	1,3434929	0,366888	23,33505	0,028334	0,093581	0,0625
24.1	1,3417221	0,361039	23,3345	0,023395	0,077269	0,0625
25.1	1,3402598	0,35621	23,33413	0,019317	0,063801	0,0625

Определяем значение ограничений: $2x_1 - x_2 - 6 =$-3,67569

Этот результат все еще далек от необходимого значения, поэтому опять увеличиваем r в 10 раз и переходим к следующему этапу.

Этап 4. r=10

Таблица 2.4

Результаты четвертого этапа метода штрафных функций

шаг	X_1	X_2	$F(x_1, x_2)$	$\frac{\partial F}{\partial x_1}$	$\frac{\partial F}{\partial x_2}$	λ
	1,34026	0,35621	144,9304	-132,306	66,22623	1
1.1	133,646	-65,87	114719			0,5
1.2	67,493	-32,76	281434			0,25
1.3	34,4166	-16,20	67731			0,125
1.4	17,8785	-7,922	15673			0,063
1.5	9,60936	-3,783	3342,9			0,031
1.6	5,47481	-1,713	602,39			0,016
1.7	3,40753	-0,678	88,274	91,006	-39,23	0,016
2.1	1,98557	-0,066	62,785	-58,646	31,134	0,016
3.1	2,90191	-0,552	51,28	41,451	-16,222	0,016
…						
43.1	2,17329	-1,123	40,644	0,1706	0,3618	0,016
44.1	2,17062	-1,129	40,642	0,163	0,3440	0,016

Завершили четвертый этап метода. Ответ на четвертом этапе

X_1=2,170623; x_2=-1,12868

Штраф равен -0,53.

Данный штраф уже достаточно мал. Однако, добьемся того, чьтобы погрешность была меньше 0.1. Для этого увеличим еще раз значение к b выполним очередную итерацию метода. Результаты сведены в следующую таблицу.

Этап 5. r=100

Увеличивая r еще в 10 раз, осуществляем аналогичные действия. Результаты приведены в таблице 2.5.

Таблица 2.5

Результаты пятого этапа метода штрафных функций

шаг	X_1	X_2	$F(x_1, x_2)$	$\frac{\partial F}{\partial x_1}$	$\frac{\partial F}{\partial x_2}$	λ
	2,1706	-1,129	65,9298	-190,66	95,7567	1
1.1	192,83	-96,89	2287			0,5
1.2	97,502	-49,01	570615			0,25
1.3	49,836	-25,07	14209			0,125
1.4	26,004	-13,098	352429			0,0625
1.5	14,08707	-7,11348	86734,08			0,03125
1.6	8,128846	-4,12108	21021,69			0,015625
1.7	5,149735	-2,62488	4949,231			0,007813
1.8	3,660179	-1,87678	1108,936			0,003906
1.9	2,915401	-1,50273	237,7715			0,001953
1.10	2,543012	-1,31571	64,43533			0,000977
2.1	2,356818	-1,2222	43,32872	-2,81305	2,3894	0,000977
3.1	2,359565	-1,22453	43,32158	0,33998	0,81921	0,000977

На этом этапе получили решение X_1=2,35956; x_2=-1,22453

Штраф равен -0,056.

Данные значения показывают, что на данной итерации уже найдена точка, которая располагается достаточно близко к искомому значению (2.25, -1.5). Поскольку производные, в отличие от предыдущих этапов, далеки от 0, можно продолжить итерационную процедуру. Однако, сходимость к истинному значению будет осуществляться крайне медленно.

В частности, продолжая, получим:

Таблица 2.6

Результаты пятого этапа метода штрафных функций с повышенной точностью

шаг	X_1	X_2	$F(x_1, x_2)$	$\frac{\partial F}{\partial x_1}$	$\frac{\partial F}{\partial x_2}$	λ
	…					
237	2,2956735	-1,35321	43,22741	0,189596	0,381469	0,000977
238	2,2954884	-1,35359	43,22723	0,189004	0,38028	0,000977
	…					

Таким образом, видно, что итерационный процесс медленно сходится к решению (2.25, -1.5).

При необходимости, можно увеличить r и продолжить.

Сведем результаты всех этапов в следующую таблицу

Таблица 2.7

Результаты решения задачи методом штрафных функций

Этап	r	X_1	X_2	Расхождение с ограничением
1	0.01	-0,44597	3,908188	-10,8
2	0.1	-0,040748	3,098545	-9,5626
3	1	1,3402598	0,35621	-3,67569
4	10	2,170623	-1,12868	-0,53
5	100	2,3595646	-1,22453	-0,056

Сравнив полученные значения с аналитическим результатам, можно видеть, что, не смотря на определенную погрешность, значения, полученные в результате численного поиска, близки к аналитическим результатам.

3.3. Метод барьерных функций

Данный метод также сводит задачу условной оптимизации к задаче безусловной оптимизации. Кроме того, также к исходной целевой функции добавляется вспомогательная функция, сформированная из ограничений. Однако, смысл данной вспомогательной функции уже другой. Она является своего рода барьером, который не позволяет выходить за пределы допустимой области. В связи с этим, начальная точка должна выбираться таким образом, чтобы все ограничения имели бы место. Как только значение приближается к барьеру, значение целевой функции должно увеличиваться. У самого барьера оно, как правило, стремится к бесконечности.

Как и для штрафных функций, каждое ограничение представляет собой отдельное слагаемое в барьерной функции.

Виды барьеров.

1. Логарифмический штраф:

$$P_i(x, r^k) = -r^k ln\left(\left(g_i^+(x)\right)\right). \quad (2.7)$$

Логарифмическая штрафная барьерная функция неопределенна в недопустимых точках.

2. Штраф, заданный обратной функцией:

$$P_i(x, r^k) = -r^k \frac{1}{g_i(x)}. \quad (2.8)$$

Является барьером и при нарушении ограничений теряет свои свойства.

При использовании барьерных функций необходимо предусматривать вычислительные процедуры, в которых в случае нарушения ограничений происходит уменьшение шага поиска, поскольку при решении именно этой подзадачи возникают затруднения. Кроме того, необходимо учитывать условие итерационных процедур:

$$F^{(k+1)}(x, r^k) < F^{(k)}(x, r^k). \quad (2.9)$$

Замечание.

При использовании барьерных функций выбор начального значения r^k может оказаться важным с точки зрения сокращения числа итераций. Если начальное значение r^k выбрано очень малым, то функция F будет мало отличаться от f(x) и метод будет сходиться быстро. Однако могут возникнуть осложнения из-за малой значимости штрафа. Слишком большое значение параметра r^k может привести к тому, что вспомогательная функция P может стать доминирующей. Это может привести к большим вычислительным затратам. Для многих задач оптимальное значение параметра r^k=1 (если r^k изменяется в сторону уменьшения).

Глава 3 Точные методы решения задач дискретной оптимизации

Эта и следующая глава посвящена исследованию задач с дискретным аргументом целевой функции. Данная задача, как правило, сложнее, чем задача непрерывной оптимизации. В большинстве случаев ее размерность не позволяет использовать полный перебор для ее решения. В связи с этим весьма актуальными являются методы, позволяющие тем или иным способом сократить количество перебираемых значений. Первая часть данной главы посвящена особенностям задач дискретной оптимизации. Метод ветвей и границ, как один из основных точных методов ее решения приведен во второй части. Третья часть содержит не менее распространенный метод динамического программирования и специфику его применения для различных задач.

3.1 Особенности задач дискретной оптимизации. Обзор методов решения задач дискретной оптимизации

Задачу дискретной оптимизации можно сформулировать следующим образом. На некотором пространстве W задано конечное или счетное множество альтернатив $X \subseteq W$. На множестве X задана функция $F: X \to R$. Задача: найти оптимальную допустимую альтернативу (максимизирующую или минимизирующую целевую функцию).

Принципиальное отличие данной задачи от задач оптимизации с непрерывной функцией в специфике множества W.

Пример – задача о ранце.

Имеется множество предметов N={1,…,n} с весами ci, i=1,..,n и ценностями pi, i=1,..,n и рюкзак емкостью R. Необходимо выбрать подмножество предметов N*, суммарный вес которых не превосходит R, имеющий максимальную ценность.

Математическая запись задачи:

Введем в рассмотрение вектор $x=(x_1,\dots,x_n)$:

$x_i=0$ означает, что предмет i не будет взят; $x_i=1$ – i-й предмет будет взят.

$$\begin{cases} \sum_{j=1}^{n} p_j x_j \to min, \\ \sum_{j=1}^{n} c_j x_j \le R, \\ x_j \in \{0,1\}, j = 1, \dots, n. \end{cases} \qquad (3.1)$$

Все задачи дискретной оптимизации в той или иной степени сводятся к перебору. Очевидно, что полный перебор является неоптимальным с вычислительной точки зрения, большинство методов основаны на идее тем или иным способом сократить число перебираемых значений. Условно все методы можно разделить на следующие категории:

- аналитические;
- методы релаксации (основаны на сведении задачи дискретной оптимизации к непрерывному случаю, поиска соответствующего решения, а затем – к оценке решения исходной задачи);
- численные или алгоритмические.

Аналитические методы позволяют записать решение задачи в виде формулы. Фактически оно представляют собой компактную запись алгоритма. Получить аналитическое решение предпочтительнее (по сравнению с остальными способами), поскольку знание некоторого закона или зависимостей, которые представляет собой формула, позволяет анализировать данное решение, изучать его свойства (сужать множество альтернатив, избавляться от неудобных ограничений, выявлять легко решаемые частные случаи) и т.д. Однако, следует отметить, что в силу сложности задач, это происходит достаточно редко.

Для разных задач дискретной оптимизации необходимо использовать принципиально разные аналитические подходы. Однако, существует одна общая основная идея, которая используется в большинстве из них. Эта идея заключается в использовании **монотонности критерия относительно частичного порядка.**

Проиллюстрируем данную идею на примере решения задачи отбраковки.

Пример. Пусть некоторые готовые изделия попадают на контроль, заключающийся в последовательном прохождении n тестов. Порядок прохождения тестов может быть произвольным. Тест i, 1,…,n длится t_i. В результате прохождения данного теста деталь с вероятностью p_i будет считаться бракованной и вернется на доработку, с оставшейся вероятностью будет проходить дальнейшее тестирование.

Необходимо выбрать такую последовательность тестов, чтобы выявить брак как можно раньше.

Фактически решение данной задачи позволит максимизировать пропускную способность системы и уменьшить очередь.

Решение.

Рассмотрим произвольный порядок прохождения тестов. Без ограничения общности, пусть деталь сначала проходит тест 1, потом тест 2 и т.д.

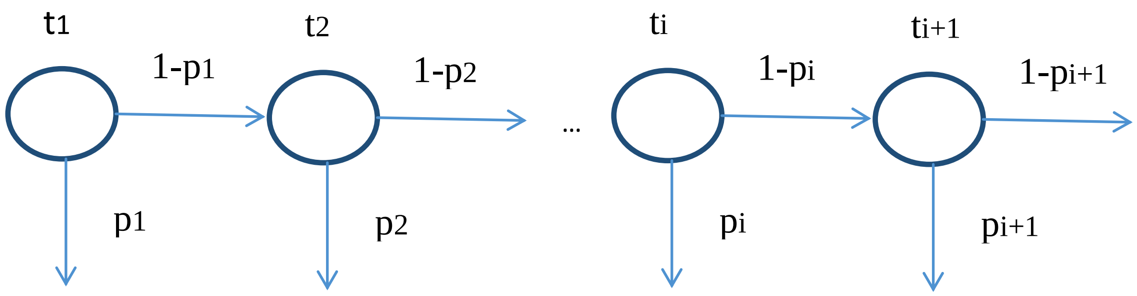

Рис. 3.1. Графическая иллюстрация задачи

Фактически данная задача требует минимизировать общее время, которое заявка проводит на участке тестирования, варьируя порядок выполнения тестов.

Выберем два произвольных рядом стоящих теста i и i+1. Фактически задача может быть сведена к тому, требуется ли два этих теста поменять местами или нет.

Вычислим время, которое затратит заявка на прохождение этих двух тестов при последовательности i->i+1. Оно будет определяться следующим образом:

$$T_1 = t_i + (1 - p_i)t_{i+1}. \quad (3.2)$$

Поменяем порядок следования тестов. Тогда время будет следующим:

$$T_2 = t_{i+1} + (1 - p_{i+1})t_i. \quad (3.3)$$

Тесты нужно оставить в исходном порядке, если:

$$T_1 \le T_2. \quad (3.4)$$

Подставив в формулу (3.4) значения T_1 и T_2, получим:

$$\frac{t_i}{p_i} \le \frac{t_{i+1}}{p_{i+1}}. \quad (3.5)$$

Таким образом, для получения оптимального порядка прохождения тестов необходимо их упорядочить по возрастанию значения $\frac{t_i}{p_i}$. Упорядочив эти значения, получим, что исходная целевая функция будет убывать (не возрастать).

Таким образом, если удается доказать, что функция монотонна относительно некоторого частичного порядка, то можно доказать, что решение достигается в одной из граничных точек.

отличие от задач непрерывной оптимизации, численные методы решения которых имеют, в целом, очень схожие схемы, методы дискретной оптимизации существенным образом отличаются друг от друга. В целом, точное решение задачи дискретной оптимизации даст метод полного перебора, когда исследуются абсолютно все значения. Однако, это требует существенных временных затрат. В большинстве своем идея алгоритмических методов решения оптимизационных задач заключается в сужении числа перебираемых значений, что позволяет снизить трудоемкость методов, возможно, чуть уступив в точности. Таким образом, для алгоритмических методов решения задач дискретной оптимизации принципиальным являются два момента:

- трудоемкость или вычислительная сложность метода;
- точность метода.

С точки трудоемкости или вычислительной сложности алгоритмы можно разделить на:

- полиномиальные;
- псевдополиномиальные;
- экспоненциальные.

Подробнее об этих классах задач можно ознакомиться в [].

С точки зрения точности все алгоритмы делятся на точные и приближенные.

Далее в данной главе будут рассматриваться точные методы решения задач дискретной оптимизации.

3.2 Метод ветвей и границ.

3.2.1.Общее описание метода ветвей и границ

Задачи дискретной оптимизации имеют конечное множество допустимых решений, которые теоретически можно перебрать и выбрать наилучшее (дающее минимум или максимум целевой функции). Практически же зачастую это бывает неосуществимо даже для задач небольшой размерности. В методах неявного перебора делается попытка так организовать перебор, используя свойства рассматриваемой задачи, чтобы отбросить часть допустимых решений. Наибольшее распространение среди схем неявного перебора получил метод ветвей и границ, в основе которого лежит идея последовательного разбиения множества допустимых решений. На каждом шаге метода элементы разбиения (подмножества) подвергаются анализу – содержит ли данное подмножество оптимальное решение или нет. Если рассматривается задача на минимум, то проверка осуществляется путем сравнения нижней оценки значения целевой функции на данном подмножестве с верхней оценкой функционала. В качестве оценки сверху используется значение целевой функции на некотором допустимом решении. Допустимое решение, дающее наименьшую верхнюю оценку, называют рекордом. Если оценка снизу целевой функции на данном подмножестве не меньше оценки сверху, то рассматриваемое подмножество не содержит решения лучше рекорда и может быть отброшено. Если значение целевой функции на очередном решении меньше рекордного, то происходит смена рекорда. Будем говорить, что подмножество решений просмотрено, если установлено, что оно не содержит решения лучше рекорда. Если просмотрены все элементы разбиения, алгоритм завершает работу, а текущий рекорд является оптимальным решением. В противном случае среди непросмотренных элементов разбиения выбирается множество, являющееся в определенном смысле перспективным. Оно подвергается разбиению (ветвлению). Новые подмножества анализируются по описанной выше схеме. Процесс продолжается до тех пор, пока не будут просмотрены все элементы разбиения.

Алгоритм метода ветвей и границ

Алгоритм имеет общий вид и может применяться к различным классам задач. Чтобы применить его к конкретной задаче необходимо определить следующие процедуры. Пусть имеется конечное множество допустимых решений G.

1. Требуется вычислить нижнюю оценку H(G) для значения целевой функции на множестве G. Для каждой конкретной задачи она будет вычисляться своим способом.
2. Требуется определить способ ветвления (разбиения исходного множества допустимых решений на подмножества G_1, G_2,…).

В общем случае алгоритм следующий.

Итерация 0. Полагаем $G^0 = G$. Далее некоторым способом (зависящим от задачи) разбиваем исходное множество на систему непересекающихся подмножеств:

$$G^0 = G_1^1 \cup G_2^1 \cup \ldots \cup G_p^1. \tag{3.6}$$

$$G_1^1 \cap G_2^1 \cap \ldots \cap G_p^1 = \emptyset. \tag{3.7}$$

Итерация k+1. Пусть на шаге k были получены множества G_1^k, G_2^k,…, G_r^k. Пусть по некоторому правилу (которое, опять же, зависит от специфики задачи) выбрано множество $G_{s(k)}^k$ и это множество разбивается на подмножества:

$$G_{s(k)}^k = G_{s(k),1}^k \cup G_{s(k),2}^k \cup \ldots \cup G_{s(k),m}^k \tag{3.8}$$

$$G_{s(k),1}^k \cap G_{s(k),2}^k \cap \ldots \cap G_{s(k),m}^k = \emptyset \tag{3.9}$$

3. Границы. В процессе ветвления появляются множества, которые не могут быть подвергнуты дальнейшему ветвлению (граничные). Эти множества называются концевыми вершинами дерева подмножеств.
4. Нахождение решения. Для граничных множеств нужно уметь находить допустимые решения, способ зависит от специфики рассматриваемой задачи.
5. Признак оптимальности. Если значение целевой функции для найденного решения меньше, чем оценки для альтернативных множеств, которые не подвергались ветвлению, то решение оптимально. В противном случае, необходимо рассмотреть все альтернативные множества с оценками, значения которых меньше целевой функции для найденного решения и продолжить их деление.

Таким образом, общая схема алгоритма метода ветвей и границ следующая:

1. Согласно формуле (3.6) и учетом ограничения (3.7) осуществили разбиение исходного множества всех вариантов G на ряд подмножеств
2. Выбрали подмножество для дальнейшего ветвления.
3. Для всех остальных подмножеств нашли нижние оценки H_i.
4. Если ветвление данного множества возможно, то осуществили процесс ветвления и перешли к пункту 2 иначе перешли к пункту 5.
5. Невозможность дальнейшего деления выбранного множества означает получение однозначного кандидата в решения x^k.

Если

$$f(x^k) < H(G_i), \tag{3.10}$$

где G_i – все множества, не подвергшиеся делению, то алгоритм заканчивает свою работу: найдено оптимальное решение x^k.

6. Если найдено хотя бы одно из подмножеств G_j, для которого неравенство (3.10) не выполняется, то необходимо перейти к данному подмножеству и начать осуществлять его деление до тех пор, пока не будет получено другого кандидата в решение. После этого перейти к пункту 5.

7. Если найдены несколько подмножеств, для которых неравенство (5) не выполняется, то дальнейшее деление необходимо произвести для каждого из этих подмножеств.

3.2.2 Пример использования метода ветвей и границ для решения задачи о коммивояжере

Пример. С помощью метода ветвей и границ решить задачу о коммивояжере.

Данную задачу можно сформулировать следующим образом. Пусть есть N городов. Известна стоимость переезда c_{ij} из города i в город j (в общем случае $c_{ij} \neq c_{ji}$). Коммивояжер должен? Выехав из некоторого города, объехать все города и вернуться в исходных город, побывав в каждом городе один раз и затратив на переезды минимальную стоимость.

Пусть задана следующая матрица стоимости переезда из города i (строки) в город j (столбцы).

Таблица 3.1

Матрица расстояний/стоимости для задачи о коммивояжере

	1	2	3	4
1	-	27	44	28
2	47	-	43	36
3	29	26	-	29
4	34	35	34	-

Предварительный этап.

Нижняя оценка – это оценка, меньше которой коммивояжер не сможет заплатить. Поскольку очевидно, что коммивояжер должен обязательно проделать процедуру «выезда из каждого города», то определим минимальные значения по строкам.

В частности, поскольку ему требуется выехать из города 1, то меньше 27 единиц он заплатить не сможет; т.к. надо выезжать из города 2, то требуется заплатить еще как минимум 36 единиц и т.д. Найдя минимумы по строкам и вычтя их из соответствующих строк, получим следующую матрицу.

Таблица 3.2

Матрица с приведенными строками

	1	2	3	4
1	-	0	17	1
2	11	-	7	0
3	3	0	-	3
4	0	1	1	-

Предварительная оценка суммы проезда будет равна 27+36+26+34=123

Проанализируем данную матрицу. Мы видим, что в первом, втором и четвертом столбцах есть нулевые элементы, а в третьем – нет. Это означает, что из каждого города есть возможность выезда в более предпочтительный по стоимости город, чем в третий (например, из первого – по второй, из второго – в четвертый, а из четвертого – в первый). Но в третий город необходимо въехать. Чтобы оценить минимальное число затрат, надо найти минимальный элемент в третьем столбце и добавить его к нижней оценке:

H(G)=123+1=124.

Вычтя из каждого элемента третьего столбца 1, получим следующую матрицу

Таблица 3.3

Приведенная матрица

	1	2	3	4
1	-	0	16	1
2	11	-	6	0
3	3	0	-	3
4	0	1	0	-

Определим способ ветвления множеств и оценки целевой функции на множествах (в данном случае, множестве), которые не будут делиться. Итак, пусть имеется множество маршрутов. Маршрут содержит множество отдельных путей (под путем будем понимать выбор города, из которого в данный момент выезжаем, и города, в который въезжаем). Например, маршрут:

1->3->4->2->1. Пути этого маршрута 1->3; 3->4; 4->2 и 2->1.

Этап 1.

На каждом шаге k будем фактически выбирать определенный путь. Будем работать с двумя множествами: основное множество – это множество ВСЕХ маршрутов, содержащих выбранный путь, альтернативное множество – множество маршрутов, которые не содержат выбранный путь.

Очевидно, что на каждом шаге логично выбирать путь с «приведенной» нулевой стоимостью (т.е. со стоимостью, не превышающей минимальное значение).Но в матрице есть несколько нулевых элементов. Из них необходимо выбрать тот путь, не поехав в который заплатим наибольшую альтернативную стоимость. Поясним это на данном примере. Если, например, из города 4 мы не сможем выехать в город 1, то из данного города можно выехать в город 3 не переплатив лишних денег. Однако, если мы из второго города не поедем в город 4, то надо ехать или в город 1 или в город 3, что повлечет как минимум дополнительных 6 единиц стоимости. Аналогичная ситуация будет со въездом в каждый город.

Таким образом, найдем для каждого нулевого элемента матрицы метку – сумму минимальных значений в соответствующих столбце и строке.

Таблица 3.4

Определение меток

	1	2	3	4
1	-	0^0	16	1
2	11	-	6	0^7
3	3	0^3	-	3
4	0^3	1	0^6	-

Итак, элемент с наибольшей меткой – это элемент (2,4). Следовательно, все исходное множество потенциальных маршрутов мы разделим на два подмножества:

- основное множество G_1^1– это множество, содержащее пусть (2,4);

- альтернативное множество G_2^1– множество маршрутов, не содержащее путь (2,4).

Оценим функцию стоимости на множестве G_2^1. Найденная метка как раз показывает минимальную необходимую доплату за отказ от данного маршрута (наименьшая альтернативная стоимость выезда из города 2 – это 6 единиц; а въезда в город 4 – 1 единица).

Для того, чтобы выполнить условие, заключающееся в однократном пребывании коммивояжере в каждом городе, запретив путь (4,2) (в противном случае будет цикл 2 -4-2).

Вычеркнем вторую строку и четвертый столбец из матрицы. Получим следующий результат (табл. 3.5).

Таблица 3.5

Результат после первого этапа

	1	2	3
1	-	0	16
3	3	0	-
4	0	-	0

Проверим матрицу на наличие нулей в каждой строке и каждом столбце. Каждая строка и каждый столбец содержат нули, значит, матрицу приводить не придется и исходная стоимость маршрута к следующему шагу не уменьшится.

Множество G_1^1 можно разделить, поэтому переходим к следующему этапу.

Этап 2.

Найдем метки каждого нулевого элемента

Таблица 3.6

Определение меток на втором этапе

	1	2	3
1	-	0^{16}	16
3	3	0^{3}	-
4	0^{3}	-	0^{16}

В данном случае для этого алгоритма пути (1,2) и (4,3) равнозначны; можно выбрать любой. Выберем первый встречающийся – (1,2).

Альтернативное множество для этого выбора – это множество всех маршрутов, содержащих путь (2,4), но не содержащих путь (1,2). Стоимость альтернативного множества увеличится на 16. Вычеркнем перую строку и второй столбец.

Выбран путь 1-2. Шагом раньше выбран путь 2-4. Следовательно, чтобы не было циклов, надо запретить путь 4-1.

Таблица 3.7

Выбор маршрута на втором этапе

	1	3
3	3	-
4	-	0^{16}

Третья строка и первый столбец содержат ненулевой элемент. Следовательно, стоимость основного маршрута увеличится на 3 единицы.

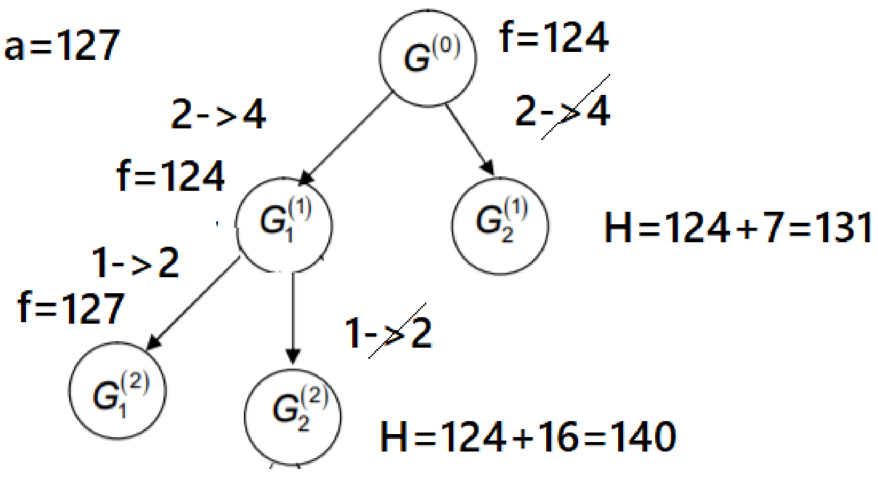

Рис. 3.2. Ветвление после второго этапа

Множество G_1^2 можно разделить, поэтому переходим к следующему шагу.

Этап 3.

После приведения, получим следующую матрицу.

Таблица 3.8

Полученная матрица к третьему этапу

	1	3
3	0	-
4	-	0

В данном случае, нет разницы, какой путь выбирать. Выберем первый – это путь (3,1).

Больше основное множество делить нельзя, поэтому следующий выбор – это фрагмент маршрута 4-3. Они уже не влияют на стоимость ни основных, ни альтернативных множеств.

Заключительный этап.

Получили итоговый маршрут: 1->2->4->3->1.

Получили конечную стоимость маршрута: 127. Имеем две оценки альтернативных маршрутов: 131 и 140. Поскольку каждая из этих оценок больше значения целевой функции на найденном решении (127), то метод завершает свою работу. В противном случае необходимо было бы осуществить возврат на данный шаг, поставить в данной ячейке запрет (например, если было бы не 131, а 126, то запрет был бы в исходной матрице 4×4 в ячейке (1,2)) и продолжать дальнейшую работу с данной матрицей.

Таким образом, получили решение задачи о коммивояжере методом ветвей и границ.

3.3 Метод динамического программирования

Метод динамического программирования является одним из наиболее распространенных способов решения сложных оптимизационных задач путем разбиения их на более простые подзадачи.

Ключевая идея в методе динамического программирования заключается в разбиении (как правило, рекурсивном) данной задачи на более мелкие подзадачи, а затем, после решения этих частных подзадач, - в объединении полученных решений в одно целое. В связи с этим, возникают ограничения на задачи, для решения которых может применяться метод динамического программирования. Они должны обладать так называемой оптимальной подструктурой. Это означает, что в оптимальном решении задачи содержится оптимальное решение ее частных подзадач.

Исходя из этого, в общем виде метод динамического программирования состоит из следующих этапов:

- разбиение задачи на подзадачи меньшего размера;
- нахождение оптимального решения подзадач (данная процедура осуществляется рекурсивно по данным трем этапам);
- использование полученного решения всех частных подзадач для формирования общего решения задачи.

Метод динамического программирования представляет собой рекурсивный процесс, позволяющий на каждом шаге с помощью некоторой рекуррентной формулы перейти от задачи некоторой размерности N к одной или нескольким задачам меньшей размерности.

Рассмотрим решение задачи для некоторых классов задач.

3.3.1 Метод динамического программирования для решения задачи о ранце.

Задача о ранце (о рюкзаке) заключается в выборе из некоторого множества вещей, которые характеризуются весом и ценностью, такого подмножества, которое, при выполнении ограничений на размер ранца (рюкзака) обладало бы максимальной ценностью.

Выведем формулу, позволяющую методом динамического программирования решать задачу о ранце. Пусть получено решение об i-1 вещи и решается задача о вещи i. Пусть к настоящему моменту рюкзак имеет вместимость r. На данном этапе есть два варианта:

- вещь i положим в рюкзак;
- вещь i брать не будем.

Рассмотрим суммарную ценность P(i,r) на данном этапе. Она зависит от двух переменных:

- номера вещи i, которая в данный момент рассматривается;
- суммарного объема всех вещей r, которые к настоящему моменту положены в рюкзак.

Если вещь i не брать, то ценность не изменится. Вес вещей также не изменится. Поэтому ценность от вещей 1,2,…,i будет равна ценности вещей 1,2,…,i-1 при сохранении объема r, то есть:

$$P(i,r) = P(i-1,r). \qquad (3.11)$$

Если вещь i в рюкзак положить, то суммарная ценность увеличится на ценность i-й вещи - p_i . При этом, если на шаге i суммарный вес вещей равен r, то на предыдущем шаге он должен равняться r-c_k, где c_k – вес вещи i. Поскольку ценность должна быть максимальной, получим рекурсивную формулу:

$$P(i,r) = max\{P(i-1,r); p_i + P(i-1,r-c_i)\}. \qquad (3.12)$$

Приведем пример решения задачи о ранце для следующей задачи. Пусть вес и ценность вещей задана таблицей.

Таблица 3.9

Вес и ценность вещей

Номер	1	2	3	4
Вес	2	1	3	4
ценность	3	2	4	5

Пусть рюкзак ограничен весом 5 единиц.

Опишем следующую таблицу. По столбцам будем откладывать возможный вес рюкзака (он будет изменяться от 0 до 5), а по строкам – количество предметов (для данной задачи количество варьируется от 0 до 4). Нулевые значения введены для того, чтобы можно было воспользоваться нерекурсивным выходом их рекуррентных формул динамического программирования. Таким образом, получим следующую заготовку для таблицы решения (табл. 3.10)

Таблица 3.10

Основа для решения задачи о ранце методом динамического программирования

Кол. предм. /R	0	1	2	3	4	5
P(0,r)						
1 P(1,r)						
2 P(2,r)						
3 P(3,r)						
4 P(4,r)						

Очевидно, что первая строка будет нулевой (если ни одну вещь не положили, то и стоимости никакой нет). Она в дальнейшем понадобится для определения ценности на последующих этапах с помощью формулы (3.11).

Таблица 3.11

Нулевая строка таблицы динамического программирования

Кол. предм. /R	0	1	2	3	4	5
P(0,r)	0	0	0	0	0	0

Рассмотрим формирование первой ненулевой строки (1 P(1,r)). Она означает, что у нас есть только первый предмет. С учетом того, что вес ранца не может быть отрицательным, при r=0 и r=1 выбора нет – ничего не кладем. В остальных случаях кладем первый предмет и получаем суммарную ценность равную 3.

Таблица 3.12

Нулевая и первая строки решения задачи

Кол. предм. /R	0	1	2	3	4	5
P(0,r)	0	0	0	0	0	0
1 P(1,r)	0	0	3+P(0,0)=3	3	3	3

Кроме данной информации, необходимо еще запоминать два момента. Первый: кладем или не кладем вещь при данном размере рюкзака, второй – каким образом было получено данное решение.

Сведения о помещении или «не помещении» вещи в рюкзак отмечаем знаками «+» или «-». На данном этапе (поскольку вещь под номером 0 нас не интересует, можно не отмечать, каким образом получено данное решение). Поэтому перепишем фрагмент следующим образом.

Таблица 3.13

Определение пути принятия решения

Кол. предм. /R	0	1	2	3	4	5
P(0,r)	0	0	0	0	0	0
1 P(1,r)	0 («-»)	0 («-»)	3 («+»)	3 («+»)	3 («+»)	3 («+»)

Рассмотрим формирование ячеек во второй строке. Она (после завершения прямого хода) должна ответить на вопрос о том, класть ли вещь под номером 2 в рюкзак или нет. Столбец 1 означает, что размер рюкзака равен 1. Если ничего не положим, то в результате получим 0. Если положим вещь 2 (ее вес равен 1, следовательно, мы можем это сделать), получим ценность 2. Следовательно, выбирая наибольший из этих элементов, получим число 2 (оно записано в строке 2, в столбце 1). Столбец 2 (объем рюкзака равен 2). Рассмотрим формулу (1). На данном этапе можем либо не положить вещь 2, либо нет. Если вещь 2 не кладем, тогда ценность P(2,r) будет совпадать с ценностью P(1,r), т.е. ценностью, когда положили вещь 1 и объем не увеличился. Это значение равно 3. Во втором случае вещь 2 можно положить. Тогда, с учетом того, что вес вещи 2 составляет 1 единицу, общая стоимость рассчитывается как стоимость вещи 2 плюс стоимость ценности для одной вещи и объема рюкзака, равным 2-1=1. Из фрагмента таблицы, полученной на втором этапе, видим, что на пересечении строки 1 и столбца 1 стоит значение 0. Значит:

$$P(2,2) = max\{P(1,2); 2 + P(1,1)\} = \max\{3,2 + 0\} = 3$$

елаем выводы, что:

1) Вещь два не берем (тогда, взяв вещь 1, получим ценность выше);
2) Данное решение получено с помощью решения P(1,2) (т.е. вещь не взяли, следовательно, размер не изменился).

Рассмотрим столбец 3 (емкость равна 3).

Если вещь 2 не положим, то ценность будет определяться ценностью помещения в рюкзак вещи 1 (т.е. значением 3) и не изменившейся емкостью рюкзака r. Вещь два можно положить в рюкзак (поскольку емкость 3 позволяет положить обе вещи). В этом случае, согласно формуле (1). Ценность P(2,3) будет определяться как ценность вещи 2 (т.е. 2) плюс значение P(1,3-1)=P(1,2). Это значение равно 2. Иными словами, если вещь два будет положена в рюкзак, то, с учетом того, емкость позволяет положить и первую вещь, суммарная ценность равна 5. Итак:

$$P(2,3) = max\{P(1,3); 2 + P(1,2)\} = \max\{3,2 + 3\} = 5$$

Делаем выводы, что:

1) Вещь под номером 2 берем;
2) Данное решение получено с помощью решения P(1,2).

Аналогичным образом получаются значения для остальных ячеек (поскольку новых вещей для данной строки нет). В итоге получим таблицу.

Таблица 3.14

Принятие решения при двух предметах

Кол. предм. /R	0	1	2	3	4	5
P(0,r)	0	0	0	0	0	0
1 P(1,r)	0 («-»)	0 («-»)	3 («+»)	3 («+»)	3 («+»)	3 («+»)
2	0(«-»)	2(«+»)	3(«-»)	5(«+»)	5(«+»)	5(«+»)

Здесь стрелками показано, на основании какой ветки в формуле (3.11) получено данное решение.

Формирование третьей строки (делаем вывод о том, кладем или нет третью вещь).

Столбец 0 – 0.

Столбец 1 (емкость рюкзака равна 1, следовательно, вещь 3 положить не можем). Поэтому

P(3,1)=P(2,1)=3.

Выводы:

1) Вещь 3 не кладем
2) Пришли из ячейки P(2,1)

Столбец 2.

Вещь 3 не кладем (как и в предыдущем случае, не позволяет емкость). Следовательно,

P(3,2)=P(2,2)=3.

Выводы:

1) Вещь 3 не кладем
2) Пришли из ячейки P(2,2)

Столбец 3. Вещь 3 положить можем. Ищем максимум.

Если вещь 3 не кладем, то P(3,3)=P(2,3)=5 (т.е. оставим вещи 1 и 2 и получим ценность 5).

Если вещь 3 кладем то,

P(3,3)=4+P(2,0)=4.

Вещь 3 выгоднее в данном случае не класть.

Выводы:

1) Вещь 3 не кладем
2) Пришли из ячейки P(2,3)

Столбец 4. Ищем максимум.

Если вещь 3 не кладем, то P(3,4)=P(2,4)=5

Если вещь 3 кладем то, с учетом веса можно положить еще и вещь 1, т.е.

P(3,4)=4+P(2,1)=6.

В денном случае вещь выгоднее положить.

Выводы:

1) Вещь 3 кладем

2) Пришли из ячейки P(2,1)

Столбец 5.

В данном случае, если вещь 3 кладем, то в оставшуюся емкость (5-3=2) можно положить вещь 1, которая более ценна. То же самое показывает формула (1):

P(3,5)=4+P(2,2)=4+3=7

Выводы:

1) Вещь 3 кладем

2) Пришли из ячейки P(2,2)

Таблица к данному этапу (табл. 3.15).

Таблица 3.15

Принятие решения при трех предметах

Кол. предм. /R	0	1	2	3	4	5
P(0,r)	**0**	**0**	**0**	**0**	**0**	**0**
1 P(1,r)	**0 («-»)**	**0 («-»)**	**3 («+»)**	**3 («+»)**	**3 («+»)**	**3 («+»)**
2	**0(«-»)**	**2(«+»)**	**3(«-»)**	**5(«+»)**	**5(«+»)**	**5(«+»)**
3	**0(«-»)**	**2(«-»)**	**3 («-»)**	**5 («-»)**	**6(«+»)**	**7 («+»)**

Последняя четвертая строка. Вещь 4 имеет вес 4, поэтому в столбцы 0,1,2,3 мы ее не кладем из за отсутствия необходимой емкости. В этом случае ценность P(k,r) будет определяться ценностью P(k-1,r).

Последний столбец 5.

Рассмотрим два варианта:

1) Вещь 4 кладем. Тогда P(4,5) будет определяться как 5+P(3,1) (ценность вещи 5, из текущей емкости 5 вычли вес вещи 4, получили 1). Итого, 5+ P(3,1)=5+2=7.

2) Вещь 4 не кладем. Тогда P(4,5)=P(3,5) -7.

Таблица 3.16

Принятие решения при четырех предметах

Кол. предм. /R	0	1	2	3	4	5
P(0,r)	**0**	**0**	**0**	**0**	**0**	**0**

Окончание табл. 3.16

1 P(1,r)	**0 («-»)**	**0 («-»)**	**3 («+»)**	**3 («+»)**	**3 («+»)**	**3 («+»)**
2	**0(«-»)**	**2(«+»)**	**3(«-»)**	**5(«+»)**	**5(«+»)**	**5(«+»)**
3	**0(«-»)**	**2(«-»)**	**3 («-»)**	**5 («-»)**	**6(«+»)**	**7 («+»)**
4	**0(«-»)**	**2(«-»)**	**3 («-»)**	**5 («-»)**	**6(«+»)**	**7 («+»)**

Таким образом, суммарная ценность всех вещей, которые можно положить в рюкзак, равна 7. Теперь на данном этапе необходимо принять решение о выборе самих вещей. Исходя из полученного значения в последней ячейке, существует 2 варианта. Рассмотрим каждый из них отдельно.

Вариант 1. Вещь 4 не кладем. Тогда, двигаясь по строкам, получаем следующие ответы:

Вещь 4 – «-» (перейти к ячейке (3,5) и, исходя из этого, выписать ответ для вещи 3)

Вещь 3 – «+» (по стрелке перейти к ячейке (2,2) и записать ответ)

Вещь 2 – «-» (переходим по стрелке к ячейке (1,2))

Вещь 1 – «+».

Вариант 2.

Вещь 4 кладем.

Вещь 4 – «+» (перейти к ячейке (3,1))

Вещь 3 – «-» (перейти к ячейке (2,1))

Вещь 2 – «+» (перейти к ячейке (1,0))

Вещь 1 – «-».

Замечание.

Решение данной задачи можно было начать с ячейки (4,5), последовательно находя значения только в тех ячейках матрицы, которые необходимы для расчета. В данном случае на каждом шаге получали бы рекуррентное соотношение, которое можно было бы окончательно посчитать, дойдя до строки 1.

3.3.2. Метод динамического программирования для решения задачи о конвейере [8]

Рассмотрим применение метода динамического программирования для решения задачи о конвейере. Цех оснащен двумя конвейерами. На оба конвейера поступают для сборки детали, после чего на каждом рабочем месте выполняется некоторое обслуживание (например, частичная сборка). На каждом конвейере имеется n рабочих мест, пронумерованных от 1 до N.

На обоих конвейерах на рабочих местах с одинаковыми номерами выполняются одни и те же операции. Однако время выполнения одних и тех же операций на разных конвейерах отличается.

- S_{ij} – рабочее место j на конвейере i;
- a_{ij} –время обслуживания на рабочем месте S_{ij};
- e_i – постановка детали на конвейер i;
- x_i – снятие готовой детали с конвейера i;
- t_{ij} - время, требуемое на перемещение детали с одного конвейера на другой после выполнения операции S_{ij}.

Без ограничения общности, иллюстрация работы такого конвейера с числовыми значениями приведена на рис. 3.3.

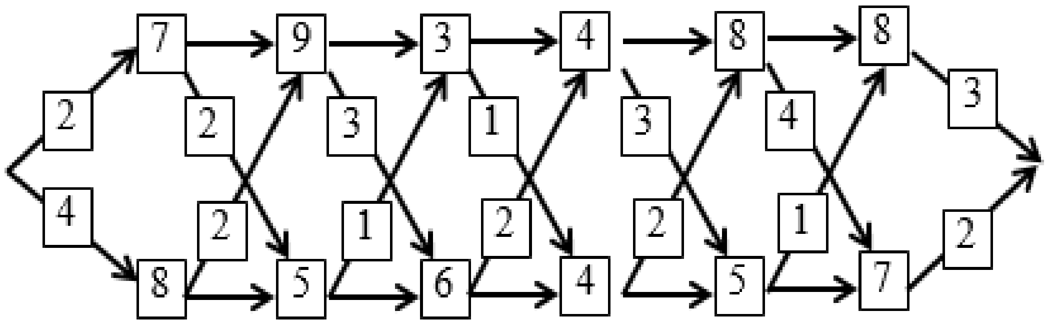

Рис. 3.3. Иллюстрация задачи о конвейере

Необходимо вывести рекуррентную формулу, Выведем рекуррентное соотношение, позволяющее оценивать время для любой стадии.

Введем в рассмотрение функцию f(n) -

Найдем оптимальное время завершения обслуживания детали. Обозначим через $f_i(j)$ – минимально возможное время, которое деталь проходит от начала до рабочего места s_{ij}. Конечная цель заключается в определении кратчайшего времени f^*, в течение которого происходит обслуживание детали на всех этапах. Оно будет определяться следующим образом:

$$f^* = \min\left(f_1(n) + x_1, f_2(n) + x_2\right) \qquad (3.13)$$

Здесь $f_1(n)$ - минимально возможное время, которое деталь проходит от начала до последнего рабочего места первого конвейере, а $f_2(n)$ – второго. Таким образом, необходимо определить величины $f_1(n)$ и $f_2(n)$. Выведем рекурсивную и нерекурсивную формулу их определения.

Для первого рабочего места первого и второго конвейеров будут справедливы следующие формулы:

$$f_1(1) = e_1 + a_{11} \qquad (3.14)$$

и

$$f_2(1) = e_2 + a_{21} \qquad (3.15)$$

соответственно.

Для остальных рабочих мест будут справедливы следующие формулы:

$$f_1(j) = \min\left(f_1(j-1) + a_{1j}, f_2(j-1) + t_{2j-1} + a_{1j}\right) \qquad (3.16)$$

и

$$f_2(j) = \min\left(f_2(j-1) + a_{2j}, f_1(j-1) + t_{1j-1} + a_{2j}\right) \qquad (3.17)$$

Проиллюстрируем работу данных формул для задачи, иллюстрация которой представлена на рис.6.1.

Прямой ход будет заключаться в определении минимального времени для каждого из конвейеров.

Для первого рабочего места для каждого конвейера будут использоваться формулы (3.14) и (3.15). Получим:

$$f_1(1) = 2 + 7 = 9.$$
$$f_2(1) = 4 + 8 = 12.$$

Определим время для обработки деталей на каждом из конвейеров на втором этапе (второе рабочее место). Кроме того, будем определять, на каком конвейере деталь была обработана на предыдущей стадии.

$f_1(2)$=min(9+9,12+2+9)=18 (на предыдущей стадией деталь была обработана на конвейере 1)

$f_2(2)$=min(9+9,12+2+9)=18 (на предыдущей стадией деталь была обработана на конвейере 1)

На третьем этапе на каждом из конвейеров время обработки будет следующим:

$f_1(3)$=min(18+3,16+1+3)=20 (на предыдущей стадией деталь была обработана на конвейере 2)

$f_2(3)$=min(16+6,18+3+6)=22 на предыдущей стадией деталь была обработана на конвейере 2)

На четвертом этапе на каждом из конвейеров время обработки будет следующим:

$f_1(4)$=min(20+4,22+2+4)=24 (на предыдущей стадией деталь была обработана на конвейере 1)

$f_2(4)$=min(22+4,20+1+4)=25 (на предыдущей стадией деталь была обработана на конвейере 1)

На пятом этапе получим следующее время обработки на каждом из конвейеров:

f1(5)=min(24+8,25+2+8)=32 (на предыдущей стадией деталь была обработана на конвейере 1)

f2(5)=min(25+5,24+3+5)=30 (на предыдущей стадией деталь была обработана на конвейере 2)

На пятом этапе получим следующее время обработки на каждом из конвейеров:

f1(6)=min(32+4,30+1+4)=35 (на предыдущей стадией деталь была обработана на конвейере 2) f1(6)=min(30+7,32+4+7)=37 (на предыдущей стадией деталь была обработана на конвейере 2).

С помощью формулы (3.13) определим наименьшее время обслуживания:

Сведем результаты в следующую таблицу.

Таблица 3.17

Решение задачи о конвейере

f1(1)=2+7=9	f2(1)=4+8=12
f1(2)=min(9+9,12+2+9)=18 (к.1)	f2(2)=min(12+5,9+2+5)=16 (к.1)
f1(3)=min(18+3,16+1+3)=20 (к.2)	f2(3)=min(16+6,18+3+6)=22 (к.2)
f1(4)=min(20+4,22+2+4)=24 (к.1)	f2(4)=min(22+4,20+1+4)=25 (к.1)
f1(5)=min(24+8,25+2+8)=32 (к.1)	f2(5)=min(25+5,24+3+5)=30 (к.2)
f1(6)=min(32+4,30+1+4)=35 (к.2)	f1(3)=min(30+7,32+4+7)=37 (к.2)

f*=min(35+3,37+2)=38 (минимум достигнут в случае, если на последней стадии деталь была обработана на конвейере 1)

Обратный ход.

Определим путь прохождения детали, исходя из того, на каком этапе на каком из конвейеров она была обработана. Поскольку минимум был достигнут на конвейере 1, то именно на этом этапе будет использован конвейер 1. Далее переходим к соответствующей ячейке таблицы (первый столбец, шестая строка). На данное рабочее место деталь попала с конвейера 2. Переходим к пятому этапу (второй столбец, пятая строка). Видим, что на четвертом этапе деталь также должна быть обработана на конвейере 2. Переходим к четвертому этапу (второй столбец, четвертая строка). Из таблицы следует, что на третьем этапе деталь была обработана на конвейере 1 (первый столбец, третья строка). Здесь видно, что на втором этапе деталь необходимо обработать на конвейере 2 (переходим ко второму столбцу второй ячейки). Исследуя соответствующую ячейку видно, что на первой стадии деталь должна быть обработана на первом конвейере. Подытожим полученные результаты, получим:

Таблица 3.18

Выбор рабочих мест для задачи о конвейере

Рабочее место (этап)	Конвейер
1	К.1
2	К.2
3	К.1
4	К.2
5	К.2
6	К.1

4.3.3. Метод динамического программирования для оптимального разбиения отрезка

Пример. Рассмотрим проблему оптимального разбиения некоторого отрезка [A,B] на участки, которая называется задачей о ближайшем соседе. Она может быть сформулирована следующим образом. Дано целое положительное число M и неотрицательная функция f(x,y), которая отражает затраты, связанные с «обслуживанием» отрезка [x,y]⊆[0,M]. Требуется разбить отрезок [0,M] на n частей таким образом, чтобы суммарные затраты, соответствующие этому разбиению были минимальны. Математическую постановку задачи можно описать следующим образом:

$$\begin{cases} \sum_{k=1}^{n} f(x_{k-1}, x_k) \to min; \\ 0 = x_0 \le x_1 \le \dots \le x_n = M \end{cases} \qquad (3.18)$$

Пусть Sn(M) – оптимальное значение целевой функции задачи (3.18). Обозначим эту задачу через <n,M>. Пусть $S_k(y)$ – оптимальное значение целевой функции задачи <k,y>. Справедливы рекуррентные соотношения:

$$S_k(y) = \begin{cases} f(0, y), k = 1, y = 1, \dots, M. \\ \min_{0 \le x \le y}\left(S_{k-1}(x) + f(x, y)\right), k = 2, \dots, n, y = 1, \dots, M. \end{cases} \qquad (3.19)$$

Пример. Пусть требуется разбить отрезок [0,8] на 4 части таким образом, чтобы минимизировать суммарные затраты. Пусть целевая функция f(x,y) задана следующей таблицей (табл. 3.19).

Таблица 3.19

Целевая функция для задачи о разбиении

	1	2	3	4	5	6	7	8
0	3	19	24	41	42	63	66	83
1	0	6	18	26	39	48	56	77
2	-	0	11	19	35	44	55	56
3	-	-	0	13	25	27	45	53
4	-	-	-	0	3	15	24	37
5	-	-	-	-	0	3	16	27
6	-	-	-	-	-	0	12	21
7	-	-	-	-	-	-	0	16
8	-	-	-	-	-	-	-	0

Найдем сначала $S_1(y)$, пользуясь прямой формулой (3.19) (без рекурсии).

$S_1(0)=0$; $S_1(1)=3$; $S_1(2)=19$; $S_1(3)=24$; $S_1(4)=41$; $S_1(5)=42$; $S_1(6)=63$; $S_1(7)=66$;

$S_1(8)=83$.

Выпишем значения S_1 в следующую таблицу.

Таблица 3.20

Результаты после первого этапа

y	0	1	2	3	4	5	6	7	8
S_1(y)	0	3	19	24	41	42	63	66	83

Далее будем искать S_2(y) через S_1(y).

$$S_2(y) = \min_{0 \le x \le y} \left(S_1(x) + f(x,y)\right), y = 1, \dots, M. \tag{3.20}$$

Будем последовательно увеличивать y от 0 до 8 и применять формулу (3.20). Получим следующие результаты. Для каждого значения y будем также запоминать, каким образом получен соответствующий минимум. Эта информаций будет необходима для заключительно этапа при определении оптимального решения.

S_2(1)=min(S_1(0)+f(0,1))=3 (нельзя сформировать другие отрезки, заканчивающиеся 1);

S_2(2)=min(S_1(0)+f(0,2); S_1(1)+f(1,2))=min(0+19; 3+6)=9 (запомним, что минимум достигается через точку 1);

S_2(3)=min(S_1(0)+f(0,3); S_1(1)+f(1,3); S_1(2)+f(2,3))=min(0+24; 3+18; 19+11)=21 (запомним, что минимум достигается через точку 1);

S_2(4)=min(S_1(0)+f(0,4); S_1(1)+f(1,4); S_1(2)+f(2,4); S_1(3)+f(3,4))=min(0+41; 3+26; 19+19; 24+13)=29 (запомним, что минимум достигается через точку 1);

S_2(5)=min(S_1(0)+f(0,5); S_1(1)+f(1,5); S_1(2)+f(2,5); S_1(3)+f(3,5); S_1(4)+f(4,5))=min(0+42; 3+26; 19+19; 24+13; 41+3)=42 (запомним, что минимум достигается через точку 0);

S_2(6)=min(S_1(0)+f(0,6); S_1(1)+f(1,6); S_1(2)+f(2,6); S_1(3)+f(3,6); S_1(4)+f(4,6); S_1(5)+f(5,6))=min(0+63; 3+48; 19+44; 24+27; 41+15; 42+3)=45 (запомним, что минимум достигается через точку 5);

S_2(7)=min(S_1(0)+f(0,7); S_1(1)+f(1,7); S_1(2)+f(2,7); S_1(3)+f(3,7); S_1(4)+f(4,7); S_1(5)+f(5,7), S_1(6)+f(6,7))=min(0+66; 3+56; 19+55; 24+45; 41+24; 42+16; 66+12)=58 (запомним, что минимум достигается через точку 5);

S_2(8)=min(S_1(0)+f(0,8); S_1(1)+f(1,8); S_1(2)+f(2,8); S_1(3)+f(3,8); S_1(4)+f(4,8); S_1(5)+f(5,8), S_1(6)+f(6,8), S_1(7)+f(7,8))=min(0+83; 3+77; 19+56; 24+53; 41+37; 42+27; 66+21; 83+16)=69 (запомним, что минимум достигается через точку 5).

Дополним предыдущую таблицу столбцом, содержащим значения S_2.

Таблица 3.21

Результаты после второго этапа

y	0	1	2	3	4	5	6	7	8
S_1(y)	0	3	19	24	41	42	63	66	83
S_2(y)	0	3(0)	9(1)	21(1)	29(1)	42(0)	45(5)	58(5)	69(5)

В скобках указано, через какое значение достигается данный минимум.

Перейдем к третьему отрезку.

$$S_3(y) = \min_{0 \le x \le y} \left(S_2(x) + f(x, y) \right), y = 1, \dots, M.$$

$S_3(1)=\min(S_2(0)+f(0,1))=3$ (нельзя сформировать другие отрезки, заканчивающиеся 1);

$S_3(2)=\min(S_2(0)+f(0,2);\ S_2(1)+f(1,2))=\min(0+19;\ 3+6)=9$ (запомним, что минимум достигается через точку 1);

$S_3(3)=\min(S_2(0)+f(0,3); S_2(1)+f(1,3); S_2(2)+f(2,3))$

И т.д.

Таблица 3.22

Результаты после третьего этапа

y	0	1	2	3	4	5	6	7	8
$S_1(y)$	0	3	19	24	41	42	63	66	83
$S_2(y)$		3(0)	9(1)	21(1)	29(1)	42(0)	45(5)	58(5)	69(5)
$S_3(y)$		3(0)	9(1)	20(2)	28(2)	32(4)	44(4)	53(4)	65(2)

Последний шаг прямого хода метода динамического программирования выполняется аналогично. Сведем результаты в следующую таблицу.

Таблица 3.23

Результаты после четвертого этапа

y	0	1	2	3	4	5	6	7	8
$S_1(y)$	0	**3**	19	24	41	42	63	66	83
$S_2(y)$		3(0)	9(1)	21(1)	**29(1)**	42(0)	45(5)	58(5)	69(5)
$S_3(y)$		3(0)	9(1)	20(2)	28(2)	**32(4)**	44(4)	53(4)	65(2)
$S_4(y)$		3(0)	9(1)	20(2)	28(2)	31(4)	35(5)	48(5)	**59(5)**

Обратный ход метода динамического программирования.

На последнем шаге прямого хода найдены минимальные затраты S* =$S_4(8)$=59, соответствующие оптимальному разбиению исходного отрезка [0,8], и оптимальное значение последней точки разбиения $x_3^* = 5$. Следовательно, теперь отрезок [0,5] необходимо оптимально разбить на три части. Значение x_2^* получим как условно оптимальное решение, соответствующее величине $S_3(5)$ из таблицы. Это будет точка $x_2^* = 4$. Теперь необходимо найти последнюю точку, которая будет разбивать отрезок [0,4]. Согласно значению, находящемуся в строке $S_2(y)$ и в строке 4, видим, что следующий столбец будет 1. Следовательно, отрезок [0,4] должен быть разбит точкой $x_1^* = 1$.

Таким образом, ответ у задачи будет следующий. Отрезок [0,8] сможет быть разбит на 4 отрезка следующим образом: [0,1], [1,4]; [4,5]; [5,8]. При этом суммарные затраты будут равны 59 единиц.

Глава 4 Эвристические методы решения задач дискретной оптимизации

4.1. Генетические алгоритмы

4.1.1 Основные теоретические сведения

Под эвристическим алгоритмом понимают алгоритм решения задачи, включающий практический метод, не являющийся гарантированно точным или оптимальным, но достаточный для решения поставленной задачи. Использование такого алгоритма позволяет ускорить решение задачи в тех случаях, когда точное решение не может быть найдено.

Эвристические алгоритмы широко применяются для решения задач, обладающих высокой вычислительной сложностью. В таких случаях вместо полного перебора вариантов, занимающего существенное время, а иногда технически невозможного, применяется значительно более быстрый, но недостаточно обоснованный теоретически алгоритм, позволяющий получить решение, которое, как правило, близко к оптимальному.

Тем не менее, при использовании эвристических алгоритмов необходимо понимать, что он не гарантирует наилучшего решения и в некоторых случаях может дать результат, далекий от оптимального. Тем не менее, в большинстве случаев эвристические алгоритмы дают приемлемое и по времени и по «качеству» решение.

Метаэвристика – это метод оптимизации, многократно использующий простые правила или эвристики для достижения решения, близкого к оптимальному.

Метаэвристики пытаются объединить основные эвристические методы в рамках алгоритмических схем более высокого уровня, направленных на эффективное изучение пространства поиска.

Рассмотрим следующие метаэвристические алгоритмы.

Генетический алгоритм это метаэвристический алгоритм, используемый для решения оптимизационных задач путём случайного подбора, комбинирования и вариации искомых параметров с использованием механизмов, аналогичных естественному отбору в природе. Является разновидностью эволюционных вычислений, с помощью которых решаются оптимизационные задачи с использованием методов естественной эволюции, таких как наследование, мутации, отбор и скрещивание.

При инициализации алгоритма задается множество особей, представляющих текущую популяцию. Как правило, и количество особей и сами особи генерируются случайным образом. Каждая из них оценивается с помощью целевой функции, которая в данном случае называется функцией приспособленности.

Из текущей популяции выбираются особи, к которым применяются так называемые генетические операторы, такие как скрещивание и мутация. В результате этого получается новое поколение особей, которое также оценивается в помощью функции приспособленности. Наилучшее решение функции для данной популяции запоминается как эталонное, которое на следующих стадиях сравнивается с наилучшим. Алгоритм заканчивает свою работу в случае, если в течение нескольких стадий не удалось улучшить решение или если проведено заданное число итераций.

Таким образом, можно выделить следующие этапы генетического алгоритма:

Задать целевую функцию (приспособленности) для особей популяции

Создать начальную популяцию

(Начало цикла)

Размножение (скрещивание)

Мутирование

Вычислить значение целевой функции для всех особей

Формирование нового поколения (селекция)

Если выполняются условия остановки, то (конец цикла), иначе (начало цикла).

Рассмотрим стадии генетического алгоритма более подробно. Содержание хромосомы (размер, тип и т.д.) будет зависеть от специфики задании. Например, для задач оранце хромосома представляет собой бинарный вектор, каждый элемент которого равен 0, если вещь не кладем в рюкзак, и 1 в противном случае. Для задачах на графах (поиск кратчайшего пути, задача о коммивояжере) каждая хромосома представляет собой один их путем (т.е. один из вариантов решения задачи).

После создания популяции осуществляется процедура скрещивания. Примером простейшего скрещивания является одноточечное скрещивания. В этом случае случайным (или заданным) образом определяется некотороая точка в массиве каждой хромосомы, и та часть, которая левее данной точки, берется, например, от родителя А, а та, которая правее – от родителя В. Для двухточечного скрещивания в массиве потомка определяются уже две точки. В результате получим три части массива, две из которых будут наследоваться, например, от родителя а, а одна – от родителя В. Пример одноточечного и двухточечного скрещиваний приведен на *рис. 4.2*.

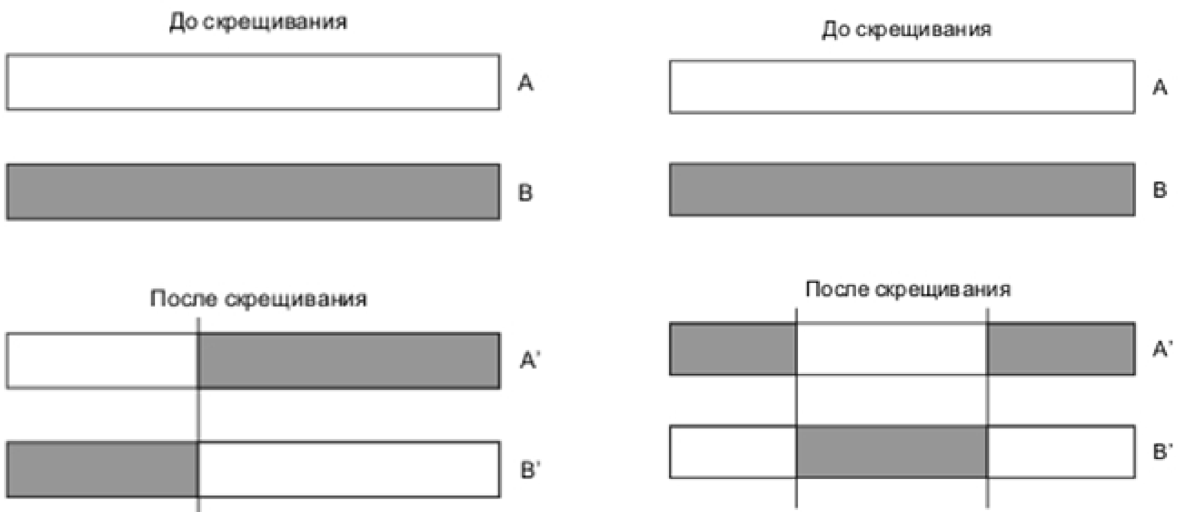

Рис. 4.1. Иллюстрация одноточечного и двухточечного скрещивания

Существует ещё два способа формирования родительской пары - инбридинг и аутбридинг. Оба эти метода основаны на понятии схожести особей. Под инбридингом понимается такой метод, когда первый родитель выбирается случайно, а вторым с большей вероятностью будет максимально близкая к нему особь. Аутбридинг же, наоборот, формирует брачные пары из максимально далеких особей. К преимуществам данных подходов можно отнести возможность их применения для решения многоэкстремальных задач. Однако их влияние на поведение генетического алгоритма не одинаково. Инбридинг характеризуется тем, что при данном способе поиск концентрируется в локальных узлах. Это приводит к разбиению популяции на отдельные локальные группы вокруг подозрительных на экстремум участков. Аутбридинг предупреждает сходимость алгоритма к уже найденным решениям, работая с новыми, неисследованными областями.

С помощью мутации создаются такие цепочки генов, которые не входили в первое поколение, но были необходимы для получения правильного решения. В тоже время она создает бесполезные и «вредные» цепочки, что затрудняет поиск решения. Необходимо правильно пообобрать настройки мутации, чтобы она и помогала и мало мешала поиску решения.

Существует множество разных алгоритмов мутаций. Среди них наиболее распространенными являются следующие:

- инверсия гена;
- замена гена на случайный;
- замена гена на соседний.

В начале, каждой из трёх процедур, реализующих перечисленные алгоритмы, хромосомам присваиваются произвольные вероятности мутации. И только если эта вероятность больше 80%, хромосома мутирует. Рассмотрим отдельно каждый алгоритм мутации.

Формирование нового поколения может осуществляться разными способами, но наиболее распространенными являются следующие два:

- элитный;
- отбор с вытеснением.

При элитном отборе в новое поколение попадают наилучшие с точки зрения целевой функции как предков, таки потомков. Для такого алгоритма характерна, как правило, достаточно быстрая сходимость.

Отбор с вытеснением подразумевает, что новую популяцию будут составлять лишь потомки (отобранные из множества потомков с точки зрения целевой функции).

4.1.2 Примеры

Рассмотрим пример решения задачи о ранце с помощью генетических алгоритмов. Задано

4.2 Метод муравьиных колоний

4.2.1 Теоретические сведения

Метод муравьиных колоний представляет представляет собой вероятностную жадную эвристику, где вероятности устанавливаются, исходя из информации о качестве решения, полученной из предыдущих решений.

Идея муравьиного алгоритма заключается в имитации поведения муравьёв, связанного с их способностью быстро находить кратчайший путь от муравейника к источнику пищи и адаптироваться к изменяющимся условиям, находя новый кратчайший путь. При своём движении муравей метит путь феромоном, и эта информация используется другими муравьями для выбора пути. Это элементарное правило поведения и определяет способность муравьёв находить новый путь, если старый оказывается недоступным.

Муравьиные алгоритмы в общем виде состоят из следующих этапов.

Первый этап – это, фактически инициализация алгоритма. Он заключается в создании множества муравьев, количество которых, как правило, случайное и зависит от специфики задачи. Кроме того, на данном этапе необходимо инициализировать все параметры, на основании которых в дальнейшем будут рассчитаны вероятности выбора каждого отрезка пути. На данном этапе эти параметры задаются случайным образом.

Второй этап – это поиск решения, заключающийся в определении наилучшего пути к цели, который на каждом этапе определяется вероятностью выбора того или иного отрезка пути. Данная вероятность рассчитывается по формуле:

$$P_{ij,k}(t) = \frac{(\tau_{ij})^{\alpha} \cdot (\eta_{ij})^{\beta}}{\sum_{l \in J_{i,k}} (\tau_{il})^{\alpha} \cdot (\eta_{il})^{\beta}}. \tag{4.1}$$

Где τ_{ij} - уровень феромона; η_{ij} – величина, обратная расстоянию между узлами i и j; α и β - параметры (константы).

При α=0 муравьиный алгоритм превращается в обычный жадный алгоритм, поскольку в формуле (4.1) будет учитываться лишь расстояние, и не будет учитываться опыт других муравьев. При β=0 получается другой крайний случай, когда во внимание берется лишь предыдущий опыт, а расстояние между узлами неважно.

Оба этих крайних случая, как правило, дают результаты, которые зачастую далеки от оптимальных.

Уровень феромона обновляется в соответствии с формулой:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \frac{Q}{L_k(t)}. \tag{4.2}$$

Здесь ρ - интенсивность испарения; $L_k(t)$ – цена текущего решения для муравья k; Q – параметр, представляющий собой значение порядка цены оптимального решения. Таким образом, $\frac{Q}{L_k(t)}$ - феромон, откладываемый муравьем k.

4.2.2 Примеры

Решим муравьиным алгоритмом задачу о коммивояжере. Без ограничения общности, возьмем α=1; β=1. Пусть исходная матрица расстояний между городами имеет вид, представленный в таблице 4.1.

Таблица 4.1

Матрица расстояний

	1	2	3	4	5
1	×	38	74	59	45
2	38	×	46	61	72
3	74	46	×	49	85
4	59	61	49	×	42
5	45	72	85	42	×

Пусть в начале работы алгоритма сгенерированы случайный числа, показывающие уровень феромона τ (в данном примере он для разных ребер будет различным, можно при инициализации взять все начальные значения τ одинаковыми). Результаты работы датчика случайных чисел занесены в следующую таблицу.

Таблица 4.2

Начальные значения τ

	1	2	3	4	5
1	×	3	2	2	2
2	3	×	1	1	1
3	2	1	×	2	2
4	2	1	2	×	1
5	2	1	2	1	×

Пусть некоторый муравей находится в первом городе. Определим вероятности его перемещения в другие города. Для этого воспользуемся формулой (4.1).

$$P_{12} = \frac{3 \cdot \left(\frac{1}{38}\right)}{3 \cdot \left(\frac{1}{38}\right) + 2 \cdot \left(\frac{1}{74}\right) + 2 \cdot \left(\frac{1}{59}\right) + 2 \cdot \left(\frac{1}{45}\right)} = 0.4283$$

$$P_{13} = \frac{2 \cdot \left(\frac{1}{74}\right)}{3 \cdot \left(\frac{1}{38}\right) + 2 \cdot \left(\frac{1}{74}\right) + 2 \cdot \left(\frac{1}{59}\right) + 2 \cdot \left(\frac{1}{45}\right)} = 0.1466$$

$$P_{14} = \frac{2 \cdot \left(\frac{1}{59}\right)}{3 \cdot \left(\frac{1}{38}\right) + 2 \cdot \left(\frac{1}{74}\right) + 2 \cdot \left(\frac{1}{59}\right) + 2 \cdot \left(\frac{1}{45}\right)} = 0.184$$

$$P_{15} = \frac{2 \cdot \left(\frac{1}{45}\right)}{3 \cdot \left(\frac{1}{38}\right) + 2 \cdot \left(\frac{1}{74}\right) + 2 \cdot \left(\frac{1}{59}\right) + 2 \cdot \left(\frac{1}{45}\right)} = 0.2411$$

Далее необходимо с полученными вероятностями выбрать тот или иной путь. Для этого:

- сгенерируем случайное число в пределах от 0 до 1;
- если оно меньше или равно 0.4283, то выбираем маршрут 1->2;
- если оно больше, чем 0.4283, но меньше, чем 0.4283+0.1466=0.5749, то выбираем маршрут 1->3;
- если значение больше, чем 0.5749, но меньше, чем 0.5749+0.184=0.7589, то выбираем маршрут 1->4;
- в противном случае выбираем маршрут 1->5.

Фактически, отрезки полученных вероятностей уложили в отрезке [0,1] для того, чтобы определить вероятность выпадения каждого из значений.

Аналогичным образом рассчитаем вероятности выхода из остальных точек и определим вероятности попадания по все пункты из пункта 2,3,4 и 5.

Пусть, без ограничения общности, генератор случайных чисел выдал число 0.68. Это означает, что на данном этапе данный маршрут будет включать путь 1->4. Выделим данный путь на графе (*рис. 4.2*).

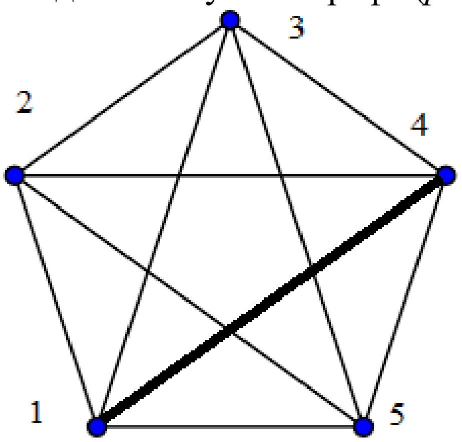

Рис. 4.2. Формирование маршрута. Итерация 1

Вычеркнем город 1 из списка городов, которые необходимо посетить. Рассчитаем вероятности переезда в любые города из четвертого (кроме первого).

$$P_{42} = \frac{1 \cdot \left(\frac{1}{61}\right)}{1 \cdot \left(\frac{1}{61}\right) + 2 \cdot \left(\frac{1}{49}\right) + 1 \cdot \left(\frac{1}{42}\right)} = 0.2023$$

$$P_{43} = \frac{2 \cdot \left(\frac{1}{49}\right)}{1 \cdot \left(\frac{1}{61}\right) + 2 \cdot \left(\frac{1}{49}\right) + 1 \cdot \left(\frac{1}{42}\right)} = 0.5038$$

$$P_{45} = \frac{1 \cdot \left(\frac{1}{42}\right)}{1 \cdot \left(\frac{1}{61}\right) + 2 \cdot \left(\frac{1}{49}\right) + 1 \cdot \left(\frac{1}{42}\right)} = 0.2938$$

Аналогичным образом осуществляем выбор следующего пути маршрута:

- сгенерируем случайное число в пределах от 0 до 1;

- если оно меньше или равно 0.2023, то выбираем маршрут 4->2;

- если оно больше, чем 0.2023, но меньше, чем 0.2023+0.5038=0.7061, то выбираем маршрут 4->3;

- если значение больше, чем 0.5749, но меньше, чем 0.5749+0.184=0.7589, то выбираем маршрут 4->5;

- в противном случае выбираем маршрут 1->5.

Пусть, без ограничения общности, выпало число 0.33. Это означает, что на данном этапе будет выбран путь 4->3 (*рис. 4.6*).

Вычеркиваем город 4 из списка городов, в которые необходимо посетить следовательно, из города 3 можно переехать лишь в города 2 и 5. Рассчитаем вероятности, согласно формуле 4.1.

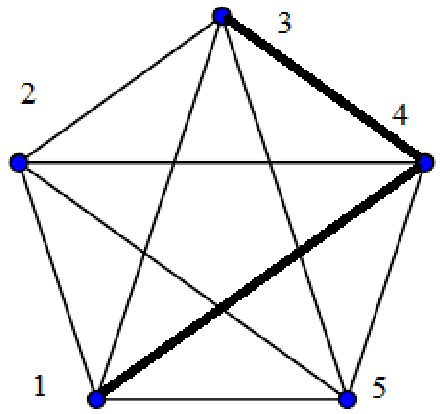

Рис. 4.3. Формирование маршрута. Итерация 2

$$P_{32} = \frac{1 \cdot \left(\frac{1}{46}\right)}{1 \cdot \left(\frac{1}{46}\right) + 2 \cdot \left(\frac{1}{85}\right)} = 0.4802$$

$$P_{35} = \frac{2 \cdot \left(\frac{1}{85}\right)}{1 \cdot \left(\frac{1}{46}\right) + 2 \cdot \left(\frac{1}{85}\right)} = 0.5198$$

Генерируем случайное число. Если оно меньше, чем 0.4802, то едем из третьего города во второй, если больше, то - в пятый.

Пусть сгенерировали число 0.24. Следовательно, переходим из города 3 в город 2. После этого остается единственный маршрут: из 2 в 5. После этого возвращаемся из пятого города в первый.

Таким образом, сгенерирован случайный маршрут: 1 ->4->3->2->5->1

Далее рассчитаем длину полученного маршрута. Она будет равна:

L_0=59+49+46+72+45=271.

Теперь рассчитаем новый феромон, который будем использовать на следующем шаге по формуле (4.2). Пусть ρ=0.1 (интенсивность испарения).

Таким образом, после данного этапа у ребер, по которым муравей прошел, уровень феромона должен увеличиться (это будет зависеть от ρ и Q), а у остальных, наоборот, уменьшиться.

Рассчитаем сначала τ_{14}, τ_{43}, τ_{32}, τ_{25}, τ_{51}.

Выберем для данного примера значение Q=100. Получим:

$$\tau_{14}(2) = (1-\rho)\,\tau_{14}(1) + \frac{100}{271} = 0.9 \cdot 2 + 0{,}369 = 2.169$$

$$\tau_{43}(2) = (1-\rho)\,\tau_{43}(1) + \frac{100}{271} = 0.9 \cdot 2 + 0.369 = 2.169$$

$$\tau_{32}(2) = (1-\rho)\,\tau_{32}(1) + \frac{100}{271} = 0.9 \cdot 1 + 0.369 = 1.269$$

$$\tau_{25}(2) = (1-\rho)\,\tau_{25}(1) + \frac{100}{271} = 0.9 \cdot 1 + 0.369 = 1.269$$

$$\tau_{51}(2) = (1-\rho)\,\tau_{51}(1) + \frac{100}{271} = 0.9 \cdot 2 + 0.369 = 2.169$$

Для остальных ребер значения уменьшатся следующим образом:

$$\tau_{ij}(2) = (1-\rho)\,\tau_{ij}(1) = 0.9 \cdot \tau_{ij}(1)$$

Поскольку изначально было лишь 3 значения, покажем данные изменения.

Если начальный уровень феромона был равен 1, то на данном этапе он станет 0.9

Если начальный уровень феромона был равен 2, то на данном этапе он станет 1.8

Если начальный уровень феромона был равен 3, то на данном этапе он станет 2.7.

Таким образом, получим следующую новую матрицу:

Таблица 4.3.

Новые значения τ

	1	2	3	4	5
1	×	2.7	1.8	2.169	1.8
2	2.7	×	0.9	0.9	1.269
3	1.8	1.269	×	1.8	1.8
4	1.8	0.9	2.169	×	0.9
5	2.169	0.9	1.8	0.9	×

На этом текущий шаг цикла завершен, можно перейти к следующему шагу.

Как и в генетических алгоритмах, в качестве условия останова можно рассмотреть отсутствие улучшения эталонного решения в течение нескольких шагов.

Заключение

Предлагаемое учебное пособие содержит необходимый теоретический материал для курса «Методы оптимизации», предназначенного для бакалавров направления «Информатика и вычислительная техника».

Необходимость практической реализации методов на ЭВМ обусловила включение в пособие большого количества примеров, иллюстрирующих каждый шаг выполнения того или иного численного метода. Это также способствует улучшению освоения теоретической базы данного курса.

Ограниченное содержание курса не позволяет включить в пособие материал, который в некоторых случаях может быть полезен при решении оптимизационных задач. В частности, достаточно кратко изложен материал, посвященный аналитическим методам решения задач на безусловный и условный экстремум. При необходимости изучения дополнительного материала по данной тематике, целесообразно воспользоваться источниками [9-11]. Также в сокращенном виде (без теории двойственности и ее приложений, что бывает весьма полезно при решении соответствующих задач) описано линейное программирование и симплекс-метод решения линейных задач. В [12] данная теория раскрыта более подробно.

Студентам, начинающим исследования в области методов оптимизации и желающих изучить возможные направления, в которых данная теория может найти свое применение, можно порекомендовать источник [5], где в простой и понятной форме изложено большое количество практических задач.

Список использованных источников

1. Васильев Ф.П. Методы оптимизации. Книга 1/ Ф.П. Васильев. М.: МЦНМО, 2011. – 620 с.
2. Васильев Ф.П. Методы оптимизации. Книга 2/ Ф.П. Васильев. М.: МЦНМО, 2011. – 434 с.
3. Васильев Ф.П. Численные методы решения экстремальных задач. – М.: Наука, 1988. – 552 с.
4. Сухарев А.Г. Курс методов оптимизации/ А.Г. Сухарев, А.В. Тимохов, В.В. Федоров: учеб. Пособие. – М.: ФИЗМАТЛИТ, 2005. – 368 с.
5. Таха Х. Введение в исследование операций, 7-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2005. – 912 с.
6. Прокопенко Н. Ю. Методы оптимизации [Текст]: учеб. пособие /Н. Ю. Прокопенко; Нижегор. гос. архитектур. - строит. ун-т. – Н. Новгород: ННГАСУ, 2018. – 118 с. ISBN 978-5-528-00287-3.
7. Сигал И Х. Задача о рюкзаке: теория и вычислительные алгоритмы Учебное пособие - М.: МИИТ. 1999 - 74с.
8. Кормен Томас Х. Алгоритмы: построение и анализ, 2-е издание: Пер. с англ./ Т.Х. Кормен, Ч.И. Лейзерсон, Р.Л. Ривест, К. Штайн. – М.: Издательский дом «Вильямс», 2005. – 1296 с.
9. Б. Т. Поляк, Методы минимизации при наличии ограничений, Итоги науки и техн. Сер. Мат. анал., 1974, том 12, 147–197.
10. Эрроу К. Дж., Гурви ц Л., Удзав а Х., Исследования по линейному и нелинейному программированию. Перев. с англ., Изд-во ин. лит., ,1962, 334 с.
11. А. В. Фиакко, Г. П. МакКормик. Нелинейное программирование. Методы последовательной безусловной минимизации. М., «Мир», 1972. – 240 с.
12. Юдин Д. Б,., Гольштей н Е. Г., Линейное программирование. Теория и конечные методы. М., Физматгиз, 1963, 775 с.

Оглавление

Учебное издание

ОЛЕЙНИКОВА Светлана Александровна
НЕДИКОВА Татьяна Николаевна

МАТЕМАТИЧЕСКИЕ МОДЕЛИ И ЧИСЛЕННЫЕ МЕТОДЫ РЕШЕНИЯ ОПТИМИЗАЦИОННЫХ ЗАДАЧ

Издание публикуется в авторской редакции
Дизайн обложки С.А. Кравец

Подписано в печать 21.09.2022. Формат 60х84 1/16
Усл. печ. л. 5,0. Заказ 000. Тираж 500 экз.

ООО Издательство «Научная книга»
394077, Россия, г. Воронеж, ул. 60-й Армии, 25-120
http://www.sbook.ru/

Отпечатано с готового оригинал-макета
в ООО «Цифровая полиграфия»
394036, Россия, г. Воронеж, ул. Куколкина, 6
Тел. (473) 261-03-61