

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Тихоокеанский государственный университет»

МЕТОДЫ МНОГОМЕРНОЙ ОПТИМИЗАЦИИ

Методические указания и задания

к выполнению лабораторных работ по дисциплине «Методы оптимизации» для
студентов направления «Прикладная математика»

Хабаровск
Издательство ТОГУ
2012

Методы многомерной оптимизации : методические указания и задания к выполнению лабораторных работ по дисциплине «Методы оптимизации» для студентов направления «Прикладная математика»/ сост. Т. М. Попова. – Хабаровск : Изд-во Тихоокеан. гос. ун-та, 2012. – 44 с.

Методические указания составлены на кафедре прикладной математики и посвящены одному из важнейших направлений подготовки выпускника университета — математической теории оптимизации. Рассмотрены теоретические, вычислительные и прикладные аспекты методов конечномерной оптимизации. Много внимания уделено описанию алгоритмов численного решения задач безусловной минимизации функций нескольких переменных. Приведены задания для лабораторной работы по одномерной оптимизации.

Печатается в соответствии с решениями кафедры «Прикладная математика» и методического совета факультета фундаментальных и компьютерных наук.

© Тихоокеанский государственный университет, 2012

Общие положения

В своей жизни человек часто сталкивается с ситуацией, когда ему из некоторой совокупности возможных вариантов своего поведения или принятия решения в какой-либо области деятельности необходимо выбрать один.

Обычно человек хочет сделать „как лучше“, но, чтобы не получить плохой результат при самых хороших намерениях, для решения задачи оптимизации нужно прежде всего найти ответы на следующие вопросы:

- Что значит „лучше“?
- Что конкретно нужно улучшить?
- За счет чего можно добиться улучшения, что можно изменить?
- В каких пределах можно производить изменения?

Отвечая на первый вопрос, необходимо сформулировать критерий оптимальности, т. е. определить те признаки и предпочтения, по которым следует провести сравнительную оценку альтернатив и выбрать среди них наилучшую с точки зрения поставленной цели оптимизации.

Именно с этой точки зрения можно ответить на второй вопрос: что конкретно нужно улучшить? Это может быть повышение производительности станка или срока службы технического устройства, снижение массы конструкции летательного аппарата или затрат на его производство и т. п.

Для ответа на два последних вопроса необходимо располагать *математической моделью* объекта оптимизации. Эта модель описывает объект при помощи соотношений между величинами, характеризующими его свойства. Обычно хотя бы часть этих величин можно изменять в некоторых пределах, что и порождает множество альтернатив, среди которых и предстоит выбрать наилучшую.

Изменяемые при оптимизации величины, входящие в математическую модель объекта оптимизации, называют *параметрами оптимизации*, а соотношения, устанавливающие пределы возможного изменения этих параметров, – *ограничениями*. Эти ограничения могут быть заданы в форме равенств или неравенств.

Если множество параметров оптимизации является подмножеством конечномерного линейного пространства, то говорят о *конечномерной задаче оптимизации* в отличие от бесконечномерных задач, которые рассматривают в вариационном исчислении и оптимальном управлении. При этом критерием оптимальности может быть требование достижения наибольшего или наименьшего значения одной или несколькими действительными (скалярными) функциями параметров оптимизации, выражающими количественно меру достижения цели оптимизации рассматриваемого объекта. Каждую из таких функций принято называть *целевой*.

Если целевая функция единственная, то задачу конечномерной оптимизации называют *задачей математического программирования*, а в противном случае — *задачей многокритериальной (векторной) оптимизации*.

1. Постановка задач многомерной оптимизации

При математической формулировке *задачи условной оптимизации* целевую функцию выбирают с таким знаком, чтобы решение задачи соответствовало поиску минимума этой функции. Поэтому формулировку общей задачи математического программирования обычно записывают так:

$$f(\vec{x}) \rightarrow \min_{Q}, \quad (1.1)$$

где $Q \subset R^N$ — множество возможных альтернатив, рассматриваемых при поиске решения задачи. Любую точку $\vec{x} \in Q$ называют допустимым решением задачи математического программирования, а само множество — множеством допустимых решений или, короче, *допустимым множеством*. Точку x^* , в которой функция $f(\vec{x})$ достигает своего наименьшего значения, называют *оптимальным решением задачи*.

При отсутствии ограничений множество Q совпадает с областью определения целевой функции. Если же рассматриваемые альтернативы

должны удовлетворять некоторым ограничениям, то множество допустимых решений сужается.

Задачу (1.1) в дальнейшем будем называть *задачей минимизации целевой функции на множестве Q* . Но целевая функция может и не достигать на Q наименьшего значения. Тогда говорят о точной нижней грани функции $f(\vec{x})$ на этом множестве и вместо (1.1) используют запись

$$f(\vec{x}) \rightarrow \inf, \quad x \in Q. \quad (1.2)$$

Отличие (1.1) от (1.2) в том, что в первом случае предполагают существование точки x^* , в которой целевая функция достигает своего наименьшего значения на множестве Q , а во втором случае такая точка может и не существовать. Поэтому решение общей задачи математического программирования состоит в том, чтобы в первом случае найти точные (или с некоторой заданной точностью) значения координат точки $x^* = \{x_j\}_{j=1, \dots, N}$ и значение целевой функции $f(x^*) = \min_Q f(x)$. Во втором случае построить такую последовательность точек $\{x_n\}_{n \in \mathbb{N}}$, которой бы соответствовала последовательность $f(x_n)$, сходящаяся к значению $\inf_Q f(x)$, и вычислить это значение с заданной точностью. Отметим, что в большинстве прикладных задач имеет место первый случай, поэтому использование записи вида (1.2) будем оговаривать особо.

Сформулируем задачу *многомерной безусловной оптимизации*: найти минимум функции $f(x)$, где $x \in \mathbb{R}^N$, при отсутствии ограничений на x , при этом $f(x)$ – это скалярная целевая функция, непрерывно дифференцируемая.

При решении этого класса задача необходимо учитывать следующие факторы:

- характер целевой функции решаемой задачи (одноэкстремальная или многоэкстремальная);
- возможность получения в процессе оптимизации информации о производных целевой функции (возможность присутствует или отсутствует);

- наличие различных подходов к организации итеративной процедуры поиска оптимума (методы, основанные на итеративном движении переменных в направлении, определяемом тем или иным способом).

Все методы решения задач безусловной оптимизации состоят в том, что мы строим последовательность точек $\{x^{(n)}\}$ таким образом, чтобы последовательность функций $f(x^{(n)})$ была убывающей (т. е. спускаемся вдоль функции). На k -м шаге ($k > 0$) определяем вектор \vec{S}_k , в направлении которого функция $f(\bar{x})$ уменьшается. В этом направлении делаем шаг величиной λ_k и получаем новую точку $x^{k+1} = x^k + \lambda_k \vec{S}_k$, в которой $f(x^{k+1}) < f(x^k)$. Последовательность $\{x^{(n)}\}_{n \in \mathbb{N}}$, удовлетворяющая этому условию, называется *релаксационной последовательностью*, а соответствующие методы – *методами спуска*. Методы решения делятся на методы с использованием информации о производных функции и без использования таковой. Различные методы спуска отличаются выбором направления и величины шага. Как правило, для нахождения λ_k используется процедура одномерного поиска.

2. Методы прямого поиска

2.1. Общая характеристика методов нулевого порядка

В методах прямого поиска минимума целевой функции (или методах нулевого порядка) используют информацию только о значениях этой функции. Многие из этих методов не имеют строгого теоретического обоснования и построены на основе эвристических соображений. Поэтому вопросы сходимости методов прямого поиска еще мало изучены, а оценки скорости сходимости обычно отсутствуют. Вместе с тем эти методы идейно связаны с методами первого и второго порядков, что в ряде случаев позволяет оценивать эффективность алгоритмов прямого поиска примени-

тельно к минимизации некоторых классов функций. Распространенным способом оценки эффективности методов прямого поиска являются вычислительные эксперименты и сравнительный анализ методов по результатам таких экспериментов. Однако следует учитывать, что этот анализ не во всех случаях может приводить к однозначным выводам о преимуществах одного метода перед другим. Во-первых, это связано с тем, что сравнению обычно подвергаются не только методы, но и программные реализации соответствующих алгоритмов. Хороший метод можно „загубить“ плохим программированием, неудачным выбором параметров алгоритма. Во-вторых, методы могут вести себя по-разному на различных этапах процесса минимизации. Удовлетворительного способа преодоления указанных трудностей не существует. Единственное, что можно сделать в подобной ситуации, — привести данные о результатах вычислений в развернутой форме, позволяющей сравнивать методы по различным критериям. Кроме того, не следует забывать, что поиск решения всегда остается искусством, которому можно научиться лишь путем проб и ошибок, применяя различные методы при решении конкретных задач.

К методам нулевого порядка относятся методы, не использующие производные для выбора направления спуска: метод Гаусса, метод вращающихся направлений (Розенброка); метод деформируемого многогранника (поиска по симплексу); метод Хука – Дживса, метод Пауэлла.

2.2. Метод Гаусса – Зейделя (покоординатный спуск)

В качестве направлений поиска используются координатные векторы $\vec{e}_1, \vec{e}_2, \dots, \vec{e}_n$, где $e_j = \{0, 0, \dots, 1, \dots, 0\}$. Таким образом, в поиске по направлению меняется только переменная x_j , остальные переменные фиксируются. Задаем начальную точку x^0 . Рассматриваем направление поиска $\vec{S}_1 = \vec{e}_1$. Находим параметр λ_1 из условий одномерной минимизации $\min_{\lambda} f(\vec{x}^0 + \lambda \vec{e}_1)$. Обозначим промежуточную точку $\vec{y}^{(1)} = x^0 + \lambda_1 \vec{e}_1$. Перейдем ко второму направлению \vec{e}_2 . Находим параметр λ_2 из условия од-

номерной оптимизации $\min_{\lambda} f(y^{(1)} + \lambda \vec{e}_2)$, обозначим $y^{(2)} = y^{(1)} + \lambda_2 \vec{e}_2$, проходим по всем направлениям координатных осей, определяя $y^{(j+1)} = y^{(j)} + \lambda_{j+1} \vec{e}_{j+1}$, где λ_j находим из условия $\min_{\lambda} f(y^{(j)} + \lambda \vec{e}_{j+1})$, $j = 0, \dots, N-1$. Следующая точка $x^{(1)} = y^{(n)}$. Рассматриваем ее как стартовую, далее повторяем процесс поиска по направлениям координатных осей (рис. 1). Процесс поиска останавливаем при выполнении условия $\|x^{(k+1)} - x^{(k)}\| < \varepsilon$ или $|f(x^{(k+1)}) - f(x^{(k)})| < \varepsilon$, где ε – заданная точность. В качестве оптимального решения выбираем $x^* = x^{(k+1)}$, $f^* = f(x^*)$.

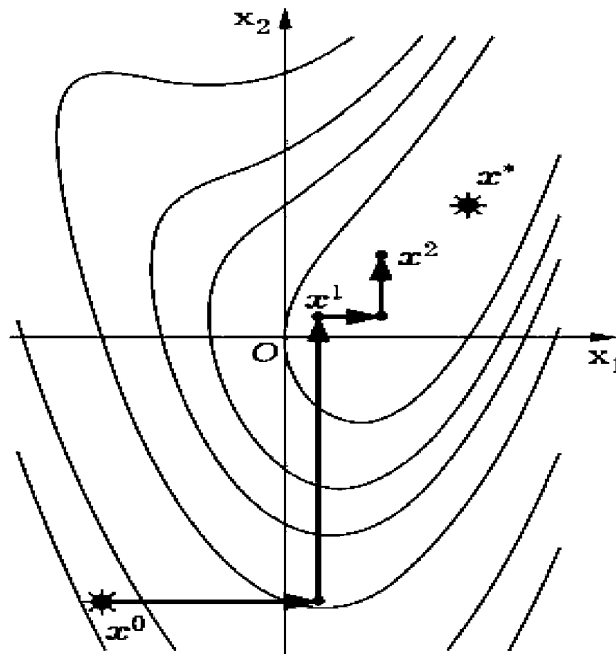


Рис. 1

Алгоритм

Шаг 0. Задать точность $\varepsilon > 0$, стартовую точку $x^{(1)}$.

Положить $y^{(1)} = x^{(1)}$; $k = j = 1$.

Шаг 1. Решить задачу одномерного поиска $\min_{\lambda} f(y^{(j)} + \lambda e_j)$. установить

λ_j . Положить $y^{(j+1)} = y^{(j)} + \lambda_j e_j$.

Шаг 2. Если $j < n$, то $j = j + 1$ и перейти на шаг 1,

если $j = n$, то перейти на шаг 3.

Шаг 3. Положить $x^{(k+1)} = y^{(n+1)}$,

если $\|x^{(k+1)} - x^{(k)}\| < \varepsilon$, то конец $x^* = x_{k+1}$,

иначе $y^{(1)} = x^{(k+1)}$; $j = 1, k = k + 1$ и перейти на шаг 1.

Достоинством метода покоординатного спуска является его простота при определении перемещения в пространстве переменных. Для гладких функций метод обеспечивает сходимость к точке локального минимума.

Недостаток. При минимизации «овражных» функций алгоритм будет делать очень мелкие шаги и может остановиться далеко от оптимума.

2.3. Метод Хука Дживса с использованием одномерной оптимизации

Эффективность прямого поиска точки минимума ограниченной снизу целевой функции можно повысить, если на каждом k -м шаге поиска соответствующим образом выбирать направление спуска. Для этого на каждом k -м шаге выделяют предварительный этап *исследующего поиска*. Целью этого этапа является выбор направления спуска путем исследования поведения целевой функции $f(\bar{x})$ в окрестности точки $x^{(k-1)}$, найденной на предыдущем шаге. В результате выполнения этапа исследующего поиска находится точка $\tilde{x}^{(k)}$, для которой $f(\tilde{x}^{(k)}) < f(x^{(k-1)})$. Направление спуска, завершающего k -й шаг поиска, определяется вектором $\tilde{x}^{(k)} - x^{(k-1)}$. Такая стратегия поиска, предложенная в 1961 г., получила название метода Хука Дживса, который состоит из двух этапов: исследующего поиска и поиска по образцу.

Исследующий поиск заключается в поиске базовой точки. Перебираем точки вдоль координатных направлений, аналогично методу Гаусса - Зейделя. Если значение целевой функции в пробной точке не превышает

значения в исходной, то шаг поиска рассматривается как успешный. В противном случае необходимо вернуться в предыдущую точку и сделать шаг в противоположном направлении. После перебора всех N координат исследующий поиск заканчивается. Полученная точка называется базовой.

Поиск по образцу: заключается в реализации единственного шага из полученной базовой точки вдоль прямой, соединяющей её с предыдущей базовой точкой. На k -м шаге переходим к этапу спуска в направлении вектора $\tilde{x}^{(k+1)} - \tilde{x}^{(k)}$, при этом $f(\tilde{x}^{(k+1)}) < f(\tilde{x}^{(k)}) \leq f(x^{(k-1)})$. Следующая точка находится по формуле

$$x^{(k)} = \tilde{x}^{(k)} + \lambda_k (\tilde{x}^{(k+1)} - \tilde{x}^{(k)}), \quad (2.1)$$

подбираем так называемый *ускоряющий множитель* λ_k из одномерной минимизации функции: $f(\tilde{x}^{(k)} + \lambda(\tilde{x}^{(k+1)} - \tilde{x}^{(k)})) \rightarrow \min_{\lambda}$, при этом $f(x^{(k)}) \leq f(\tilde{x}^{(k+1)})$, либо λ_k задаем постоянным, обычно полагаем $\lambda_k = 2$. На рис. 2 иллюстрируются этапы исследующего поиска и спуска для первых двух шагов поиска точки x^* минимума целевой функции двух переменных при $\lambda_k = 2$ из начальной точки $x^{(0)}$.

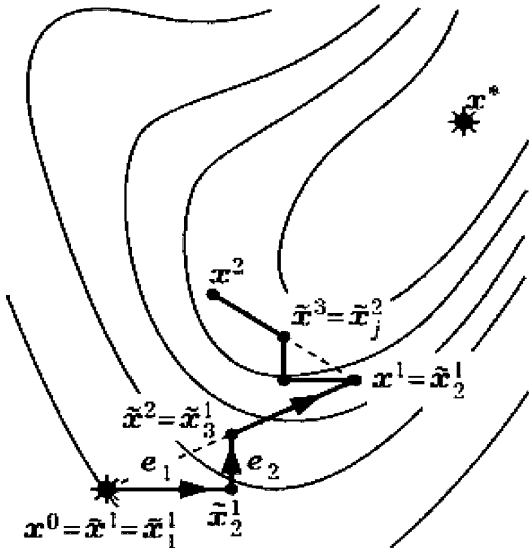


Рис. 2

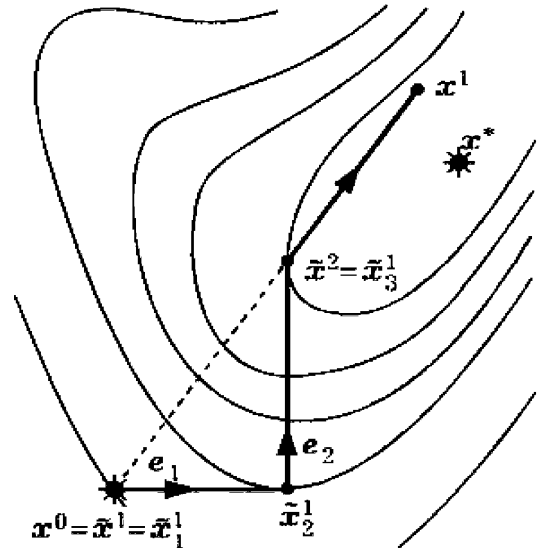


Рис. 3

На рис. 3 иллюстрируется первый шаг поиска оптимальной точки минимума целевой функции двух переменных с применением на этапе исследующего поиска модифицированного метода покоординатного спуска, а на этапе спуска — процедуры подбора ускоряющего множителя $\lambda_k > 0$ в формуле (2.1), исходя из условия минимума целевой функции в направлении вектора $\tilde{x}^{(2)} - \tilde{x}^{(1)}$.

Алгоритм

Шаг 0. Задать $\varepsilon > 0$, $x^{(1)}$ — начальная точка. Положить $y^{(1)} = x^{(1)}$, $k = j = 1$ (j — номер шага на итерации).

Шаг 1. Решить задачу одномерной оптимизации $\min_{\lambda} f(y^{(j)} + \lambda e_j) \rightarrow \lambda_j$,
положить $y^{(j+1)} = y^{(j)} + \lambda_j e_j$.

Шаг 2. Если $j < n$, то положить $j = j + 1$ и перейти на шаг 1,
если $j = n$, то перейти на шаг 3.

Шаг 3. Положить $x^{(k+1)} = y^{(n+1)}$.

Шаг 4. Если $\|x^{(k+1)} - x^{(k)}\| < \varepsilon$, то конец $x^* = x^{(k+1)}$,
иначе перейти на шаг 5.

Шаг 5. Положить $\vec{S} = x^{(k+1)} - x^{(k)}$.

Шаг 6. Решить задачу $\min_{\lambda} f(y^{(j)} + \lambda \vec{S}) \rightarrow \bar{\lambda}$. Положить $y^{(1)} = x^{(k+1)} + \bar{\lambda} \vec{S}$.

Шаг 7. Положить $j = 1$, $k = k + 1$. Перейти на шаг 1.

2.4. Метод вращающихся направлений (Розенброка)

Существует еще один альтернативный подход к решению задач минимизации овражных функций, основанный на итеративной перестройке системы ортогональных направлений таким образом, чтобы они определяли движение вдоль дна оврага. Такие методы получили название методов вращающихся направлений. Метод, реализующий эту стратегию поиска, также предусматривает проведение исследующего поиска на каждом k -м

шаге. Целью исследующего поиска является выбор текущего направления спуска с учетом информации о поведении целевой функции в окрестности точки $x^{(k-1)}$, найденной на предыдущем шаге.

Отличие этого метода от метода Хука — Дживса состоит в способе выбора направлений исследующего поиска. Если в методе Хука — Дживса они фиксированы и коллинеарны направлениям векторов стандартного базиса, то в рассматриваемом методе выбор этих направлений проводят в процессе минимизации целевой функции путем построения на каждом k -м шаге поиска нового ортонормированного базиса методом Грамма — Шмидта.

Итогом выполнения этого этапа является нахождение точки $\tilde{x}^{(k)}$, для которой $f(\tilde{x}^{(k)}) < f(x^{(k-1)})$. Тогда вектор $\tilde{x}^{(k)} - x^{(k-1)}$ определит направление спуска на k -м шаге.

Рассмотрим $\bar{s}_1, \bar{s}_2, \dots, \bar{s}_n$ — линейно независимые, взаимно ортогональные векторы, т. е. $(\bar{s}_j, \bar{s}_k) = \begin{cases} 0, & j \neq k, \\ 1, & j = k \end{cases}$,

После решения задач $\min_{\lambda} f(y^{(j)} + \lambda \bar{s}) \rightarrow \lambda_j$ определяется новая точка $x^{(k+1)} = x^{(k)} + \sum_{j=1}^n \lambda_j \bar{s}_j$. Выбор новых направлений $\bar{s}_1, \bar{s}_2, \dots, \bar{s}_n$ осуществляется на основе следующих соотношений (метод Грамма — Шмидта):

$$a_j = \begin{cases} s_j, & \text{если } \lambda_j = 0, \\ \sum_{i=1}^n \lambda_i s_i, & \text{если } \lambda_j \neq 0; \end{cases} \quad b_j = \begin{cases} a_j, & \text{при } j = 1, \\ a_j - \sum_{i=1}^{j-1} (a_i, \bar{s}_i) \bar{s}_i, & \text{при } j \geq 2; \end{cases} \quad (2.2)$$

$$\bar{s}_j = \frac{b_j}{\|b_j\|}.$$

На рис. 4 иллюстрируются этапы одного шага поиска точки минимума целевой функции двух переменных из начальной точки.

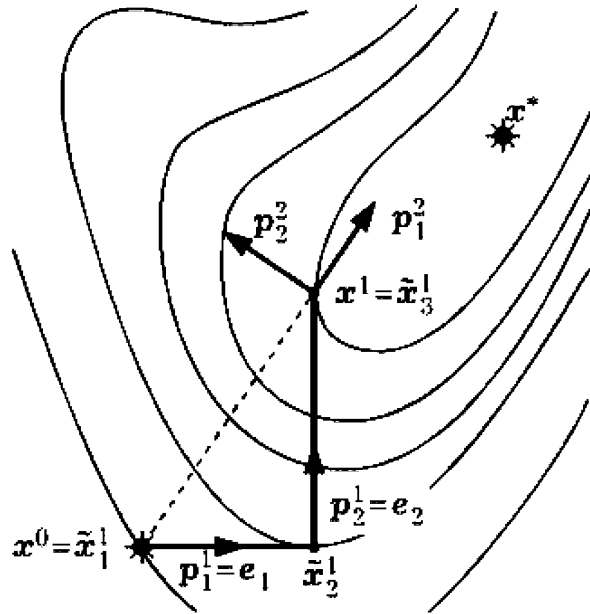


Рис. 4

Алгоритм

Шаг 0. Задать $\varepsilon > 0$, $S_j = e_j$, $j = 1, \dots, n$, x^1 – начальная точка.

Положить $y^{(1)} = x^{(1)}$, $k = j = 1$ (j – номер шага на итерации).

Шаг 1. Решить задачу одномерной оптимизации $\min_{\lambda} f(y^{(j)} + \lambda S_j) \rightarrow \lambda_j$,

положить $y^{(j+1)} = y^{(j)} + \lambda_j S_j$.

Шаг 2. Если $j < n$, то положить $j = j + 1$ и перейти на шаг 1.

Шаг 3. Положить $x^{(k+1)} = y^{(n+1)}$.

Шаг 4. Если $\|x^{(k+1)} - x^{(k)}\| < \varepsilon$, то конец $x^* = x^{(k+1)}$.

Шаг 5. Положить $j = 1$, $k = k + 1$

Шаг 6. Определить новое множество направлений по формуле (2.2).

Перейти на шаг 1.

Экспериментальное сравнение алгоритмов Хука Дживса и Розенброка по числу вычислений целевой функции в процессе оптимизации говорит в пользу алгоритмов вращающихся направлений, это выигрыш растет при

увеличении размерности. Но необходимо учитывать более существенные вычислительные затраты на пересчет системы ортогональных направлений.

Для минимизации овражных функций с извилистыми оврагами, у которых крутизна склонов намного больше, чем крутизна дна оврага, возможно применение комбинированных алгоритмов. На первом этапе – простые алгоритмы (покоординатного спуска), на втором более сложные. Условием перехода на второй этап может служить соотношение $f(x_k) \leq \Theta f(x_1)$, где x_1 – начальная точка поиска, $0 \leq \Theta \leq 1$ подбирается в зависимости от характера и размерности вектора оптимизационных переменных.

Например, для $n = 2$, $\Theta \in [0,6; 0,8]$, при $n = 4$, $\Theta \in [0,65; 0,75]$, при $n = 6$, $\Theta \in [0,1; 0,2]$

2.5. Метод поиска по симплексу. Метод Нелдера Мида

Это наиболее известный метод среди методов, не использующих стратегию движения по направлениям. Метод основан на том, что экспериментальным образом, содержащим наименьшее количество точек, является *симплекс*.

Регулярный (правильный) симплекс в N -мерном пространстве – это многогранник, образованный $N+1$ равноотстоящими точками – вершинами симплекса.

В пространстве R^2 правильный симплекс – равносторонний треугольник, а в R^3 правильный тетраэдр.

Если определена базовая вершина симплекса, то координаты всех остальных вершин симплекса $\{x^{(i)}, i = 1, \dots, N\}$ могут быть рассчитаны по формулам

$$x_j^{(i)} = \begin{cases} x_j^{(0)} + d_1, & i \neq j, \\ x_j^{(0)} + d_2, & i = j, \end{cases} \quad (2.3)$$

где $d_1 = \frac{a(\sqrt{N+1}-1)}{N\sqrt{2}}$; $d_2 = \frac{a(\sqrt{N+1}+N-1)}{N\sqrt{2}}$, a – длина ребра симплекса.

Важное свойство симплекса это то, что новый симплекс можно построить на любой грани исходного путём отражения какой-либо вершины относительно центра тяжести всех остальных вершин симплекса:

$\tilde{x}^{(k)} = 2x_c - x^{(k)}$, где $x_c = \frac{1}{N} \sum_{i \neq k} x^{(i)}$ центр тяжести (рис. 5).

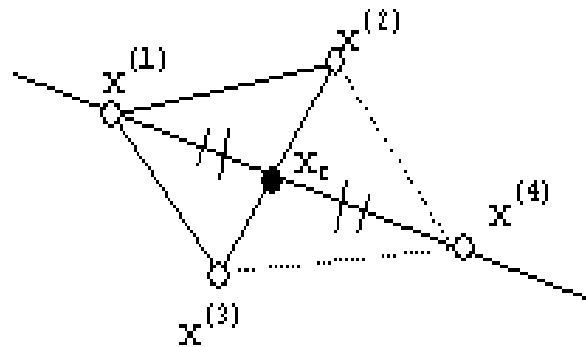


Рис. 5

Вычисляя значения целевой функции в вершинах симплекса, получаем информацию о характере изменения этой функции в области расположения симплекса.

Поиск точки минимума целевой функции с помощью правильных симплексов производится следующим образом:

- На каждой итерации происходит вычисление целевой функции во всех точках симплекса и их упорядочивание по возрастанию значений.
- Затем осуществляется последовательная попытка построить новые симплексы с лучшими значениями целевых функций, путем отражения точек с худшими значениями.
- Если последовательная попытка отражения двух худших вершин оканчивается неудачей, то производится сжатие симплекса к точке с наименьшим значением и осуществляется новая итерация.
- Поиск завершается, как только разность между значениями функции в точках симплекса становится достаточно малой.

При реализации алгоритма используются следующие процедуры:

- Процедура отражения (рис. 6, а):

$$\tilde{x}^{(k)} = 2x_c - x^{(k)}. \quad (2.4)$$

- Процедура сжатия вовнутрь ($0 < \alpha < 1$) (рис. 6, б):

$$\tilde{x}^{(k)} = x_c - \alpha(x_c - x^{(k)}). \quad (2.5)$$

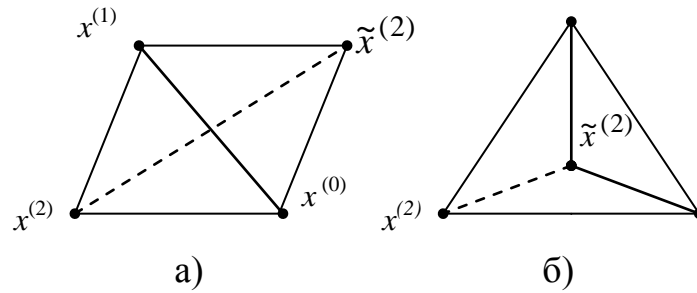


Рис 6.

- Процедура отражения со сжатием ($0 < \alpha < 1$) (рис 7а)

$$\tilde{x}^{(k)} = x_c + \alpha(x_c - x^{(k)}). \quad (2.6)$$

- Процедура отражения с растяжением ($\beta > 1$) (рис.7б)

$$\tilde{x}^{(k)} = x_c + \beta(x_c - x^{(k)}). \quad (2.7)$$

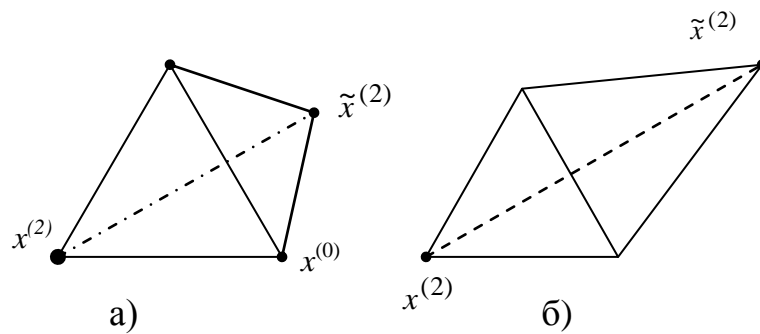


Рис. 7

Алгоритм

Шаг 0. Задать $\varepsilon > 0$, параметры алгоритма: α, β, a , базовую точку,

построить начальный симплекс (2.3). Вычислить $f(x^0)$.

Шаг 1. Вычислить значения целевой функции в точках симплекса

$$f(x^{(i)}), i = 1, \dots, N.$$

Шаг 2. Упорядочить точки по возрастанию значений целевой функции.

$$\text{Базовая точка } x^{(0)} : f(x^{(0)}) = \min_i f(x^{(i)}).$$

Шаг 3. Если выполнено условие достижимости заданной точности

$$\frac{1}{N} \sum_{i=1}^N \left(f(x^{(i)}) - f(x^{(0)}) \right)^2 < \varepsilon^2, \text{ то конец и } x^* \cong x^{(0)}, \text{ иначе на шаг 4.}$$

Шаг 4. Определить центр тяжести $x_c = \frac{1}{N} \sum_{i \neq k} x^{(i)}$, пробные точки $\tilde{x}^{(k)}$

по формулам (2.4) (2.7), $k = 1, \dots, 4$.

Найти $f_{\min} = \min_k f(\tilde{x}^{(k)})$ и соответствующую точку $\tilde{x}^{(0)}$.

Шаг 5. Если $f_{\min} < f(x^{(N)})$, то положить $x^{(N)} = \tilde{x}^{(0)}$. Перейти на шаг 2.

Шаг 6. Сжать симплекс в 2 раза: $x^{(i)} = \frac{x^{(i)} + x^{(0)}}{2}$, $i = 1, \dots, N$, перейти на шаг 1.

С теоретической точки зрения методы поиска при помощи нерегулярного симплекса изучены недостаточно полно, однако практические вычисления указывают на их работоспособность. В задачах безусловной минимизации рекомендуют выбирать $\alpha = 1/2$, $\beta = 2$, или $\alpha = 1/4$, $\beta = 5/2$. Вопрос выбора оптимальных параметров для каждой конкретной целевой функции может быть решен после дополнительного исследования свойств метода с помощью вычислительного эксперимента.

Графическая иллюстрация процесса поиска по симплексу приведена на рис. 8.

При работе с деформируемым (нерегулярным) симплексом возможно его вырождение: когда вершины попадают в пространство, размерность которого меньше размерности задачи (например, при $N=2$ вершины сим-

плекса попадают на одну прямую). Формально это $\tilde{x}^{(k)} = x^{(k)}$, при этом симплекс уже не выйдет из этого пространства и не может продолжать поиск минимума во всем пространстве. Поэтому необходимо осуществить процедуру *обновления* симплекса в соответствии с (2.3).

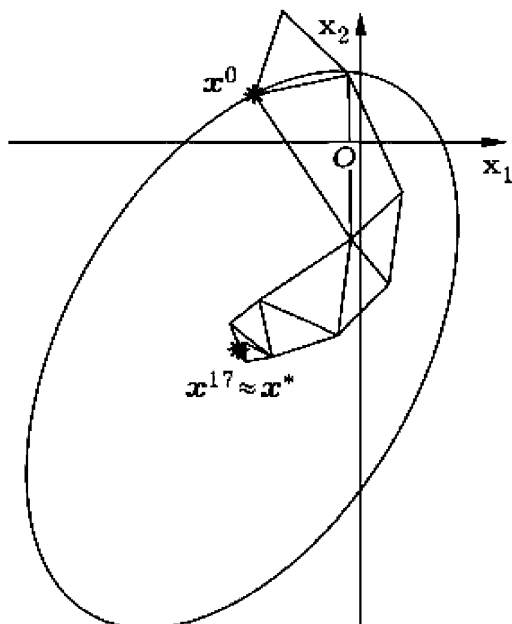


Рис. 8

Достоинства метода: простота; малое количество заранее установленных параметров; простая стратегия поиска, вычисление только значений функции, небольшой объём требуемой памяти.

Недостатки метода: метод работает эффективно при $N \leq 6$, алгоритм основан на циклическом движении по координатам. Это может привести к вырождению алгоритма в бесконечную последовательность исследующих поисков без поиска по образцу.

2.6. Методы сопряженных направлений (Пауэлла)

Разработан целый ряд методов безусловной оптимизации, использующих понятие *сопряженности векторов*, которые определяют направление поиска на смежных итерациях. Важность рассмотрения этого вида ме-

тодов заключается в том, что для функций, имеющих квадратичный вид, они обеспечивают сходимость к точке минимума за число шагов, не более размерности задачи. Данный метод подходит также и для других функций после разложения в ряд Тейлора в окрестности точки оптимума.

Основная идея: если квадратичную функцию n переменных привести к виду суммы полных квадратов, то её оптимум может быть найден в результате n одномерных поисков по преобразованным координатным направлениям. Процедура преобразования квадратичной функции $q(x) = a + bx + \frac{1}{2}(Qx, x)$ к виду суммы полных квадратов эквивалентна нахождению такой матрицы преобразования U , которая приводит матрицу квадратичной формы (Qx, x) к диагональному виду. Квадратичная форма (Qx, x) путём преобразования $x = U y$ приводится к виду: (Dy, y) , где D диагональная матрица. Вместо координат вектора x в стандартной координатной системе, используются его координаты в новой системе, задаваемой векторами y_j . Поскольку y_j совпадают с главными осями квадратичной формы, то матрица D диагональная. Таким образом, с помощью преобразования переменных квадратичной функции строится новая система координат, совпадающая с главными осями квадратичной функции, следовательно, одномерный поиск точки оптимума в преобразованных координатах эквивалентен поиску вдоль каждой из осей квадратичной функции (рис.9).

Таким образом, для нахождения оптимума достаточно провести n одномерных поисков вдоль векторов y_j . Ясно, что такой способ нахождения точки минимума нетрудно обобщить на n -мерный случай, хотя для реализации этого способа сначала нужно будет решить громоздкую задачу на собственные значения матрицы Q порядка n . Естествен вопрос: нельзя ли избежать нахождения собственных векторов матрицы Q , но все же сократить количество итераций при поиске точки минимума?

Важным понятием, используемым методами этого вида, является *сопряженность* векторов. Векторы s_1, s_2, \dots, s_k называют *H -сопряженными*,

если они линейно независимы и $(Hs_j, s_i) = 0$, при $j \neq i$, где H – симметрическая матрица порядка $n \times n$.

Рассмотрим этот вопрос на примере двумерной задачи для квадратичной формы $f(x) = \frac{1}{2}(Qx, x)$. Выберем начальную точку x^0 и произвольное направление спуска, определяемое вектором p_1 , не обязательно сонаправленным антиградиенту $w_1 = -Qx^0$ функции $f(x)$ этой точке, но составляющее с ним острый угол, т.е. $(Qx^0, p_1) < 0$. Проведя из точки x^0 в этом направлении исчерпывающий спуск, получим точку $x^{(1)}$ на линии уровня $f(x) = f(x^{(1)})$ (рис. 10).

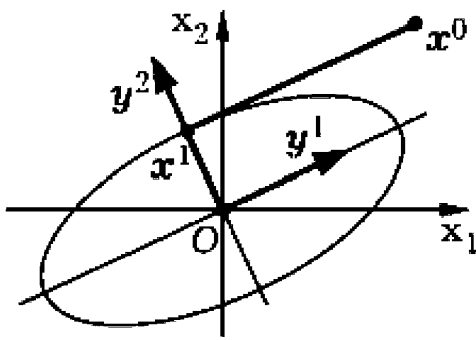


Рис. 9

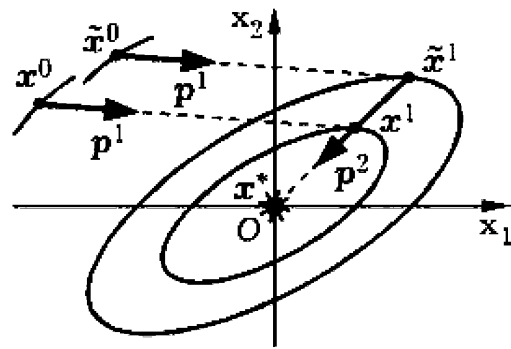


Рис. 10

Затем выберем точку \tilde{x}^0 , не лежащую на прямой, проходящей через x^0 и $x^{(1)}$, но такую, что $(Q\tilde{x}^0, p_1) < 0$, и проведем исчерпывающий спуск из точки \tilde{x}^0 в том же направлении p_1 . В результате получим точку касания $\tilde{x}^{(1)}$ на линии уровня $f(x) = f(\tilde{x}^{(1)})$. Оказывается, что для получения искомой точки минимума достаточно провести исчерпывающий спуск из точки $x^{(1)}$ (или из точки $\tilde{x}^{(1)}$) в направлении p_2 , коллинеарном вектору $x^{(1)} - \tilde{x}^{(1)}$.

Векторы p_1 и p_2 , называют сопряженными относительно матрицы Q (или Q -ортогональными), а направления, определяемые такими векторами, *сопряженными направлениями*. Понятие сопряженности векторов является обобщением понятия их ортогональности.

Алгоритм метода сопряженных направлений

Шаг 0. Выбрать начальную точку x^0 , задать $\varepsilon > 0$.

Шаг 1. Положить $s_1 = e_1$ 1-я орта, $k = 1$.

Найти λ_1 из решения задачи $\min_{\lambda} f(x^0 + \lambda s_1)$ и точку $x^1 = x^0 + \lambda_1 s_1$.

Шаг 2. Положить $y^{(k-1)} = x^{(k)} + e_{k+1}$.

Шаг 3. Найти точку y_k путем последовательной минимизации функции $f(x)$ по направлениям $s_i, i = 1, \dots, k$.

Шаг 4. Положить $k = k + 1$, направление $s_k = x^{(k-1)} - y^{(n-1)}$.

Найти λ_k из решения задачи $\min_{\lambda} f(x^{(k-1)} + \lambda s_k)$,

точку $x^{(k)} = x^{(k-1)} + \lambda_0 s_k$.

Шаг 5. Если $k < n$, то перейти к шагу 1, иначе шаг 6.

Шаг 6. Если $f(x)$ является квадратичной, то $x^* = x_n$ и закончить.

Шаг 7. Если $\|x^0 - x_n\| \leq \varepsilon$, то $x^* = x_n$ и закончить, иначе $x^0 = x_n$ и перейти к шагу 1.

3. Методы первого порядка

Методы 1-го порядка используют информацию о производной функции. Если ограниченная снизу целевая функция $f(x)$, $x \in \mathbb{R}^n$, является дифференцируемой на множестве R^n , то алгоритм поиска точки x^* ее минимума можно построить, используя информацию, по крайней мере, о градиенте этой функции. Такие методы называются градиентными.

Градиентные методы безусловной оптимизации используют только первые производные целевой функции и являются методами линейной аппроксимации на каждом шаге, т. е. целевая функция на каждом шаге заменяется касательной гиперплоскостью к ее графику в текущей точке.

Во всех этих методах предполагается, что $f(x)$, ∇f существуют и непрерывны. Все эти методы основаны на итерационной процедуре, определяемой формулой $x^{(k+1)} = x^{(k)} + \alpha_k S^{(k)}$, где α_k — величина шага, $S^{(k)}$ — вектор в направлении $x^{(k+1)} - x^{(k)}$.

Градиентные методы различаются только способом определения α_k , и $S^{(k)}$ обычно определяется путём решения задачи оптимизации $f(x)$ в направлении $S^{(k)}$. Направление $S^{(k)}$ зависит от того, как аппроксимируется функция $f(x)$.

3.1. Метод наискорейшего спуска (Коши)

Впервые такой метод рассмотрел и применил еще О. Коши в XVIII в. Идея его проста: градиент целевой функции $f(x)$ в любой точке есть вектор в направлении наибольшего возрастания значения функции. Следовательно, антиградиент будет направлен в сторону наибольшего убывания функции и является направлением наискорейшего спуска. Антиградиент (и градиент) ортогонален поверхности уровня $f(x)$ в точке x .

Пусть в точке x требуется определить направление наискорейшего спуска (то есть направление наибольшего локального уменьшения $f(x)$). Разложим $f(x)$ в ряд Тейлора в окрестности точки x и отбросим члены второго порядка по Δx и выше $f(x) = f(x^{(k)}) + (\nabla f(x^{(k)}), \Delta x) + \dots$

Локальное уменьшение $f(x)$ определяется вторым слагаемым, т. е. наибольшее уменьшение $f(x)$ будет тогда, когда $(\nabla f(x^{(k)}), \Delta x)$ будет иметь наибольшую отрицательную величину. Этого можно добиться вы-

бором $S^{(k)} = -\nabla f(x^{(k)})$. Этот случай соответствует наискорейшему локальному спуску $x^{(k+1)} = x^{(k)} - \alpha_k \nabla f(x^{(k)})$.

На рис.11 показана релаксационная последовательность. Построенная по методу наискорейшего спуска.

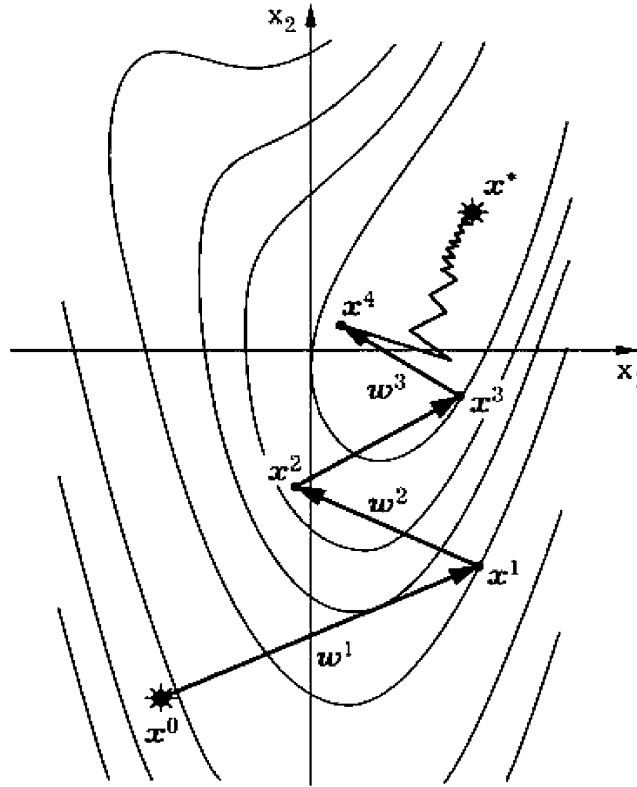


Рис.11

Алгоритм

Шаг 0. Задать $\varepsilon > 0$, x^1 начальную точку. Положить $k = 1$.

Шаг 1. Если $\|\nabla f(x^{(k)})\| < \varepsilon$, то конец и оптимум $x^* = x^{(k)}$,

$$\text{иначе вычислить } S^{(k)} = -\frac{\nabla f(x^{(k)})}{\|\nabla f(x^{(k)})\|}.$$

Шаг 2. Решить задачу одномерной оптимизации $\min_{\lambda} f(x^{(k)} + \lambda S^{(k)}) \rightarrow \lambda_k$.

Положить $x^{(k+1)} = x^{(k)} + \lambda_k S^{(k)}$.

Шаг 3. Положить $k = k + 1$ перейти на шаг 1.

Недостатки: одним из недостатков этого метода является то, что он сходится к любой стационарной точке, в том числе и седловой, которая не может быть решением. Также отмечена очень медленная сходимость наискорейшего спуска в общем случае. Дело в том, что спуск является "наискорейшим" в локальном смысле. Если гиперпространство поиска сильно вытянуто ("овраг"), то антиградиент направлен почти ортогонально дну "оврага", т. е. наилучшему направлению достижения минимума. В этом смысле прямой перевод английского термина "steepest descent", т. е. спуск по наиболее крутому склону, более соответствует положению дел, чем термин "наискорейший", принятый в русскоязычной специальной литературе. Метод обладает большой надёжностью, но медленную сходимость вблизи точки минимума устранить нельзя. Поэтому метод самостоятельно обычно не используется, а используется как предварительная процедура для более сложных методов.

Достоинство: на каждой k -ой итерации $f(\bar{x}^{k+1}) < f(\bar{x}^k)$ выполняется свойство убывания функции.

3.2. Методы сопряженных градиентов

В методе сопряженных направлений происходит отклонение направления наискорейшего спуска путем добавления к нему направления, используемого на предыдущем шаге. В методе сопряженного градиента строится последовательность направлений поиска $S^{(k)}$, являющихся линейными комбинациями градиента текущего направления наискорейшего спуска, и предыдущих направлений поиска, т. е.

$$S^{(k+1)} = -\nabla f(x^{(k)}) + \sum_{i=1}^k \alpha_i S^{(i)}.$$

Метод Флетчера Ривса. Величины α_j выбираются так, чтобы новое направление $S^{(k)}$ было сопряжено со всеми предыдущими направлениями.

При этом критерием окончания поиска является выполнение условия:
 $(\nabla^2 f(x^{(k)}), S^{(k)}) = 0$.

Доказано, что

$$\alpha_1 = \alpha_2 = \dots = \alpha_{k-1} = 0, \quad \alpha_k = \frac{\|\nabla f(x^{(k)})\|^2}{\|\nabla f(x^{(k-1)})\|^2},$$

и это очень ценный результат, позволяющий строить быстрый и эффективный алгоритм оптимизации.

Алгоритм Флетчера Ривса

Шаг 0. Задать $\varepsilon > 0$, x_1 начальную точку, $S^{(1)} = -\nabla f(x^{(1)})$.

Положить $y^{(1)} = x^{(1)}$, $k = j = 1$.

Шаг 1. Если $\|\nabla f(x^{(k)})\| < \varepsilon$, то конец и оптимум $x^* = y^{(j)}$,

иначе одномерная оптимизации $\min_{\lambda} f(y^{(j)} + \lambda S^{(j)}) \rightarrow \lambda_j$,

положить $y^{(j+1)} = y^{(j)} + \lambda_j S^{(j)}$.

Шаг 2. Если $j < n$ перейти к шагу 3, иначе к шагу 4.

Шаг 3. Положить $S^{(j+1)} = -\nabla f(y^{(j+1)}) + \frac{\|\nabla f(y^{(j+1)})\|^2}{\|\nabla f(y^{(j)})\|^2} S^{(j)}$, $j = j + 1$,

перейти к шагу 1.

Шаг 4. Положить $x^{(k+1)} = y^{(n+1)}$, $y^{(1)} = y^{(n+1)}$,

$S^{(1)} = -\nabla f(y^{(1)})$, $k = k + 1$, $j = 1$, перейти к шагу 1.

Метод обладает квадратичной сходимостью. Преимуществом алгоритма Флетчера Ривса является то, что он не требует обращения матрицы и экономит память ЭВМ, так как ему не нужны матрицы, используемые в Ньютоновских методах, но в то же время почти столь же эффективен как

квази-Ньютоновские алгоритмы. Так как направления поиска взаимно сопряжены, то квадратичная функция будет минимизирована не более, чем за n шагов. В общем случае используется рестарт, который позволяет получать результат.

Алгоритм Флетчера-Ривса чувствителен к точности одномерного поиска, поэтому при его использовании необходимо устранять любые ошибки округления, которые могут возникнуть. Кроме того, алгоритм может отказать в ситуациях, где Гессиан становится плохо обусловленным. Гарантии сходимости всегда и везде у алгоритма нет, хотя практика показывает, что почти всегда алгоритм приводит к приближенному оптимуму.

Метод Поллака Райвера. Это алгоритм, аналогичный методу Флетчера - Ривза, с модификацией направлений. Строим последовательность точек $x^{(j+1)} = x^{(j)} + \lambda_j S^{(j)}$, где направления $S^{(j)}$ строим с учетом предыдущих направлений: $S^{(j+1)} = -\nabla f(x^{(j+1)}) + \gamma_j S^{(j)}$,

$$\gamma_j = \frac{(\nabla f(x^{(j)}) - \nabla f(x^{(j-1)}), \nabla f(x^{(j)}))}{\|\nabla f(x^{(j)})\|^2}.$$

4. Методы 2-го порядка (Ньютоновские методы)

4.1. Методы, использующие производные второго порядка

Если целевая функция $f(x)$ является дважды дифференцируемой в R^n , то эффективность процесса поиска точки x^* ее минимума можно повысить, используя информацию не только о градиенте этой функции, но и о ее матрице Гессе $H(x)$. Направление поиска, соответствующее наискорейшему спуску (методы, рассмотренные выше), связано с линейной аппроксимацией целевой функции. Методы, использующие вторые производные, возникли из квадратичной аппроксимации целевой функции: $f(x)$,

которую можно получить при разложении функции в ряд Тейлора 2-го порядка,

$$\varphi(x) = f(x^k) + \nabla f(x^k)(x - x^k) + \frac{1}{2}(x - x^k)^T H(x^k)(x - x^k),$$

где $H(x^k)$ матрица Гессе (вторых производных). Минимум $f(x)$ (если он существует) достигается там же, где и минимум квадратичной формы $\varphi(x)$.

Если матрица Гессе $H(x^{(k)})$ целевой функции, вычисленная в точке $x^{(k)}$, является положительно определенной, то точка $x^{(k)}$ минимума функции $\varphi(x)$ единственна и может быть найдена из условия, что ее градиент равен нулевому вектору: $\nabla \varphi = \nabla f(x^{(k)}) + H(x^{(k)})(x - x^{(k)}) = 0$, следовательно, $x^{(k)} = x^{(k-1)} - H^{-1}(x^{(k-1)})\nabla f(x^{(k-1)})$.

Алгоритм оптимизации, в котором направление поиска определяется из этого соотношения, называется *методом Ньютона*, а направление $p_k = -H^{-1}(x^{(k-1)})\nabla f(x^{(k-1)})$ — *ньютонovskyм направлением*, составляет с вектором градиента тупой угол (рис. 12).

В задачах поиска минимума произвольной квадратичной функции с положительной матрицей вторых производных метод Ньютона дает решение за одну итерацию независимо от выбора начальной точки.

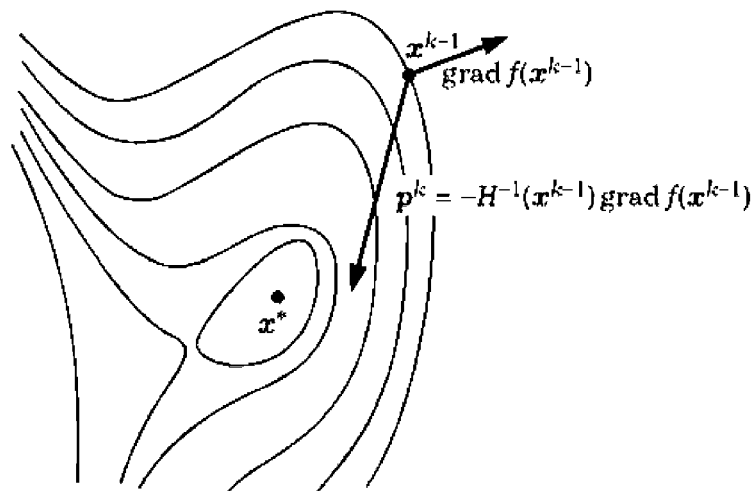


Рис. 12

4.2. Модификации метода Ньютона

Метод Ньютона считается эталонным, с ним сравнивают все разрабатываемые оптимизационные процедуры. Для применения метода Ньютона матрица Гессе должна быть положительно определенной и хорошо обусловленной (определитель ее должен быть существенно больше нуля, т. е. отношение наибольшего и наименьшего собственных чисел должно быть близко к единице). Но матрица Гессе может быть вырожденной и не иметь обратной матрицы.

Эту проблему можно решить, если направление спуска задавать вектором $p_k = (\beta_k I - H(x^{(k-1)}))^{-1} \nabla f(x^{(k-1)})$, где I — единичная матрица порядка n , а β_k — параметр, выбираемый так, чтобы в точке $x^{(k-1)}$ матрица $H_k = \beta_k I - H(x^{(k-1)})$ была положительно определена.

В связи с этим более серьезной проблемой является необходимость вычисления и обращения на каждой итерации матрицы порядка n , что в случае большой размерности пространства \mathbf{R}^n является достаточно трудоемкой операцией. На практике обычно не вычисляют матрицу, обратную к положительно определенной матрице H_k , а вектор p_k находят из решения системы линейных алгебраических уравнений (СЛАУ) $H_k p_k = -\nabla f(x^{(k-1)})$. Эту СЛАУ можно решить различными численными методами, например прямыми и итерационными.

Эти приемы называются *модификацией метода Ньютона*, они используют ньютоновские направления по мере возможности и уклоняющиеся от них только тогда, когда это необходимо.

Общий принцип модификаций метода Ньютона состоит в следующем: на каждой итерации сначала строится некоторая "связанная" с $H(x^k)$, положительно определенная матрица H_k , а затем направление спуска S_k вычисляется по формуле $H_k S_k = -\nabla f(x^k)$. Так как H_k положительно определена, то направление S_k обязательно будет направлением спуска. Процедуру построения H_k организуют так, чтобы она совпадала с матрицей Гессе,

если она является положительно определенной. Эти процедуры строятся на основе некоторых матричных разложений.

Опишем алгоритм варианта метода Ньютона поиска точки x^* минимума дважды дифференцируемой целевой функции $f(x)$, в котором направление спуска определяется путем решения СЛАУ.

Шаг 0. Зададим начальную точку $x^{(0)}$, начальную величину шага $\lambda_0 = 1$, коэф.дробления шага $0 < \nu \leq 1$, точность ε , номер итерации $k=0$.

Шаг 1. Вычислим $\nabla f_k = \nabla f(x^k)$, $H_k = H(x^k)$, если выполнено условие $\|\nabla f_k\| \leq \varepsilon_{\nabla}$ и H_k – положительно определенная, то конец и $x^* = x^k$, иначе если H_k положительно определенная, то на шаг 3, иначе шаг 2.

Шаг 2. Подбираем β_k , так чтобы $H_k = \beta_k I - H(x^k)$ была положительно определенной, переходим на шаг 3.

Шаг 3. Находим направление спуска p_k из решения СЛАУ

$$H_k p_k = -\nabla f(x^k),$$

Шаг 4. Вычисляем $x^{k+1} = x^k + \lambda_k p_k$ и $f(x^{k+1})$.

Шаг 5. Если $f(x^{k+1}) \geq f(x^k)$, то полагаем $\lambda_k = \nu \lambda_k$ и переходим к шагу 4, иначе к шагу 6.

Шаг 6. Проверка любого условия окончания поиска (4.1).

Если оно выполнено, то $x^* = x^{k+1}$,
иначе $k = k + 1$ и вернемся к шагу 1.

Условия останова поиска:

$$\|x^{k+1} - x^k\| \leq \varepsilon_x \text{ или } \|f(x^{k+1}) - f(x^k)\| \leq \varepsilon_f. \quad (4.1)$$

При произвольном выборе начальной точки x^0 метод Ньютона обладает квадратичной скоростью сходимости, и имеет место оценка

$$\|x^{(k)} - x^*\| \leq \frac{L}{\lambda_1} \|x^{(k-1)} - x^*\|^2, \quad k \in N.$$

Здесь λ_1 – оценка наименьшего собственного числа матрицы Гессе в окрестности точки x^* и $\|H(x) - H(y)\| \leq L\|x - y\|$.

4.3. Метод Ньютона-Рафсона

Данный метод состоит в многократном использовании Ньютоновского направления при оптимизации функций, не являющихся квадратичными.

Основная итерационная формула многомерной оптимизации $x^{k+1} = x^k + \lambda_{k+1} S^{k+1}$ используется в этом методе при выборе направления оптимизации из соотношения $S^{k+1} = -H^{-1}(x^k) \nabla f(x^{k+1})$ $\lambda_{k+1} = 1$.

Реальная длина шага скрыта в ненормализованном Ньютоновском направлении S^{k+1} . Так как этот метод не требует значения целевой функции в текущей точке, то его иногда называют *непрямым или аналитическим методом оптимизации*. Его способность определять минимум квадратичной функции за одно вычисление выглядит на первый взгляд исключительно привлекательно. Однако это "одно вычисление" требует значительных затрат. Прежде всего, необходимо вычислить n частных производных первого порядка и $n(n+1)/2$ второго. Кроме того, матрица Гессе должна быть обратимой. Это требует уже порядка n^3 вычислительных операций. С теми же самыми затратами методы сопряженных направлений или методы сопряженного градиента могут сделать порядка n шагов, т. е. достичь практически того же результата. Таким образом, итерация метода Ньютона-Рафсона не дает преимуществ в случае квадратичной функции.

Если же функция не квадратичная, то:

- начальное направление уже, вообще говоря, не указывает действительную точку минимума, а значит, итерации должны повторяться неоднократно;

- шаг единичной длины может привести в точку с худшим значением целевой функции, а поиск может выдать неправильное направление, если, например, гессиан не является положительно определенным;
- гессиан может стать плохо обусловленным, что сделает невозможным его инвертирование, т. е. определение направления для следующей итерации.

Сама по себе стратегия не различает, к какой именно стационарной точке (минимума, максимума, седловой) приближается поиск, а вычисления значений целевой функции, по которым можно было бы отследить, не возрастает ли функция, не делаются. Значит, все зависит от того, в зоне притяжения какой стационарной точки оказывается стартовая точка поиска. Стратегия Ньютона-Рафсона редко используется сама по себе без модификации того или иного рода.

4.4. Метод Марквардта

Это комбинация методов Ньютона и Коши. Вдали от точки минимума направление определяется по методу Коши, а в окрестности точки минимума – по методу Ньютона.

Строим релаксационную последовательность:

$$x^{k+1} = x^k + \lambda_k p_k, \quad \lambda_k = 1, \quad (4.2)$$

$$p_k = \left(\beta_k I - H(x^{(k-1)}) \right)^{-1} \nabla f(x^{(k-1)}). \quad (4.3)$$

На начальном этапе $\beta_k \approx 10^4$, при этом второй член в (4.3) много больше первого, поэтому поиск осуществляется по методу Коши. По мере приближения к точке оптимума β_k уменьшается и стремится к нулю. Таким образом, вблизи точки оптимума первый член в (4.3) много больше второго и поиск осуществляется по методу Ньютона.

Если после первого шага $f(x^{(1)}) < f(x^0)$, то следует выбрать $\beta_1 < \beta_0$ и реализовать следующий шаг, в противном случае $\beta_0 = \gamma \beta_0$, где $\gamma > 1$, и повторить предыдущий шаг.

Вычислительные эксперименты показали, что метод наиболее эффективен для функций вида суммы квадратов

5. Методы переменной метрики

5.1. Квазиньютоновские методы: общая характеристика

Среди алгоритмов многомерной минимизации следует выделить группу алгоритмов, которые объединяют достоинства метода наискорейшего спуска и метода Ньютона. Такие алгоритмы принято относить к так называемым квазиньютоновским методам. Особенность этих алгоритмов состоит в том, что при их применении нет необходимости вычислять и обрабатывать матрицу Гессе целевой функции $f(x)$ и в то же время удается сохранить высокую скорость сходимости алгоритмов, присущую методу Ньютона и его модификациям.

В этих методах обратная матрица Гессе аппроксимируется другой матрицей – метрикой. Метрика изменяется на каждой итерации, и поэтому методы так же называются методами с *переменной метрикой*.

Элементы релаксационной последовательности $\{x_k\}$ в алгоритмах квазиньютоновских методов минимизации непрерывно дифференцируемой в \mathbf{R}^n целевой функции $f(x)$ строят в соответствии с рекуррентным соотношением $x^{k+1} = x^k + \lambda_k p_k$, направление спуска на каждой k -й итерации задают в виде $p_k = -\tilde{H}_k \nabla f(x^k)$, где \tilde{H}_k – метрика, а $\tilde{H}_{k+1} = \tilde{H}_k + C_k$, где C_k – корректирующая матрица.

Нужно построить последовательность метрик $\{\tilde{H}_k\}$, которая при $k \rightarrow \infty$ имела бы предел, равный $H^{-1}(x^*)$, где $H(x^*)$ – матрица Гессе в точке минимума x^* функции $f(x)$.

5.2. Методы Пирсона

Пирсон предложил несколько методов с аппроксимацией обратного гессиана без явного вычисления вторых производных, т. е. путем наблю-

дения за изменениями направления антиградиента. При этом получаются сопряженные направления. Эти алгоритмы отличаются только деталями. Приведем те из них, которые получили наиболее широкое распространение в прикладных областях.

Они аппроксимируют матрицу Гессе или обратную к ней, но используют для этого только первые производные. В большинстве из методов применяются сопряжённые направления.

На каждом шаге $x^{k+1} = x^k + \lambda_k S^k$, где $S^k = -\tilde{H}^k \nabla f(x^k)$ – направление поиска. В этом алгоритме обратный гессиан аппроксимируется матрицей, вычисляемой на каждом шаге по формуле

$$\tilde{H}^{k+1} = \tilde{H}^k + \frac{\left((x^{k+1} - x^k) - \tilde{H}^k (\nabla f(x^{k+1}) - \nabla f(x^k))\right)(x^{k+1} - x^k)^T}{(x^{k+1} - x^k)(\nabla f(x^k) - \nabla f(x^{k+1}))}$$

или в другой модификации

$$\tilde{H}^{k+1} = \tilde{H}^k + \left(x^{(k+1)} - x^{(k)} - \tilde{H}^k \cdot (\nabla f(x^{(k+1)}) - \nabla f(x^{(k)}))\right) \cdot A_k,$$

$$\text{где } A_k = \frac{\tilde{H}^k \cdot (\nabla f(x^{(k+1)}) - \nabla f(x^{(k)}))^T}{(\nabla f(x^{(k+1)}) - \nabla f(x^{(k)}))^T \cdot \tilde{H}^k \cdot (\nabla f(x^{(k+1)}) - \nabla f(x^{(k)}))}.$$

В качестве начальной матрицы H^0 выбирается произвольная положительно определенная, симметрическая матрица (обычно единичная). Данный алгоритм Пирсона часто приводит к ситуациям, когда матрица H^k становится плохо обусловленной, а именно – она начинает осциллировать, колеблясь между положительно определенной и не положительно определенной, при этом определитель матрицы близок к нулю. Для того чтобы избежать этой ситуации, необходимо через каждые n шагов перезадавать матрицу, приравнивая ее к H^0 .

5.2. Метод Дэвидона Флетчера Пауэлла

Метод Дэвидона Флетчера Пауэлла (ДФП) основан на использовании ньютоновских направлений, но не требует вычисления обратного гессиана на каждом шаге. Направление поиска на k -м шаге определяется

направлением $p_k = -\tilde{H}^k \nabla f(x^{(k)})$, где \tilde{H}^k – положительно определенная симметричная матрица, которая обновляется на каждом шаге и в пределе становится равной обратному гессиану. Матрица \tilde{H}^k строится следующим образом: в качестве начальной выбираем единичную матрицу, далее на k -м шаге: $\tilde{H}^{k+1} = \tilde{H}^k + A^k + B^k$,

$$A_k = -\frac{u^{(k)} \left(u^{(k)} \right)^T}{\left(u^{(k)}, v_k \right)}, \quad B_k = -\frac{\tilde{H}^k v_k v_k^T \left(\tilde{H}^k \right)^T}{\left(\tilde{H}^k v_k, v_k \right)}, \quad (5.1)$$

где $u^{(k)} = x^{(k)} - x^{(k-1)}$, $v_k = \nabla f(x^{(k)}) - \nabla f(x^{(k-1)})$.

Алгоритм Дэвидона Флетчера Пауэлла

Шаг 0. Выбираем стартовую точку x^0 , начальную матрицу \tilde{H}^0 – единичную, точность $\varepsilon > 0$, $k = 1$, вычислим $f(x^{(k-1)})$, $\nabla f(x^{(k-1)})$.

Шаг 1. Определяем направление спуска $p_k = -\tilde{H}^{k-1} \nabla f(x^{(k-1)})$.

Шаг 2. Найдем λ_k , минимизируя функцию $\min f(x^{(k-1)} + \lambda p_k)$,

Определяем $x^{(k)} = x^{(k-1)} + \lambda_k p_k$.

Шаг 3. Вычисляем $f(x^{(k)})$, $\nabla f(x^{(k)})$, если выполнены условия останова (4.1), или $\|\nabla f(x^{(k)})\| < \varepsilon$, то поиск завершаем и $x^* = x^{(k)}$, иначе на шаг 4.

Шаг 4. Обновляем \tilde{H}^k , на k -м шаге $\tilde{H}^{k+1} = \tilde{H}^k + A^k + B^k$ по формулам (5.1), $k = k + 1$, переходим на шаг 1.

Метод эффективен на практике, если ошибка вычислений градиента $\nabla f(x^{(k)})$ невелика и матрица \tilde{H}^k не становится плохо обусловленной. Матрица A_k обеспечивает сходимость \tilde{H}^k к $H^{-1}(x^*)$, матрица B_k обеспечивает положительную определенность \tilde{H}^{k+1} на всех этапах.

$$\tilde{H}^{(k+1)} = I + \sum_{i=1}^k A_i + \sum_{i=1}^k B_i.$$

В случае квадратичной функции $\sum_{i=1}^{n-1} A_i = H^{-1}$, $\sum_{i=1}^{n-1} B_i = -I$, т. е. алгоритм ДФП использует сопряженные направления.

Таким образом, метод ДФП использует как идеи ньютоновского подхода, так и свойства сопряженных направлений, и при минимизации квадратичной функции сходится не более чем за n итераций. Если оптимизируемая функция имеет вид, близкий к квадратичной функции, то метод ДФП эффективен за счет хорошей аппроксимации H^{-1} (метод Ньютона). Если же целевая функция имеет общий вид, то метод ДФП эффективен за счет использования сопряженных направлений.

На практике оказалось, что метод ДФП может давать отрицательные шаги или окончиться в нестационарной точке. Это возможно, когда \tilde{H}^k становится плохо обусловленной. Этого можно избежать путем увеличения числа получаемых значимых цифр или перезадаванием матрицы \tilde{H}^k в виде специальной диагональной матрицы, где $h_{jj} = \frac{x_j^{(k)} - x_j^{(k-1)}}{\nabla f(x^{(k-1)})}$.

Ошибки округления, и особенно неточность линейного поиска, могут послужить причиной потери устойчивости метода ДФП и даже привести к ошибочным шагам, когда значение целевой функции на некоторой итерации возрастает, вместо того, чтобы уменьшаться.

Практическая проверка эффективности алгоритма показала, что он столь же эффективен, как и метод Флетчера Ривса.

5.3 Метод Бroyдена Флетчера Шенно

Возможны и другие способы построения последовательности \tilde{H}^k матриц с применением $\tilde{H}_{k+1} = \tilde{H}_k + C_k$. Так, например, в алгоритме квазиньютоновского метода, называемого в литературе методом Бройдена Флетчера Шенно, или БФШ-методом, который аналогичен методу ДФП, обновление матрицы \tilde{H}^k строится по формуле $\tilde{H}^{k+1} = \tilde{H}^k + A_k + B_k + C_k$,

где A_k, B_k находим по формулам (5.1), а $C_k = (\tilde{H}^k v^k, v^k) r^k (r^k)^T$, при этом $r^k = \frac{\tilde{H}^k v^k}{(\tilde{H}^k v^k, v^k)} - \frac{u^k}{(u^k, v^k)}$. В качестве начальной матрицы выбирается единичная матрица.

5.4. Методы Пауэлла и Мак Кормика

В алгоритме Пауэлла обновление матрицы \tilde{H}^k оценивающей матрицу Гессе производится по формуле: $\tilde{H}^{k+1} = \tilde{H}^k - \frac{(w^k - w^{k-1})(w^k - w^{k-1})^T}{(v^k, w^k)}$, где $w^k = u^{(k)} + \tilde{H}^k v^k$.

В алгоритме Мак Кормика $\tilde{H}^{k+1} = \tilde{H}^k - \frac{w^k (u^{(k)})^T}{(v^k, u^{(k)})}$.

Все рассмотренные методы обновления матриц \tilde{H}^k сохраняют свойство положительной определенности, и последовательность $\{\tilde{H}^k\}_{k \in N}$ сходится при $k \rightarrow \infty$ к матрице $H^{-1}(x^*)$.

6. Методы случайного поиска.

Методы случайного поиска реализуют итеративный процесс движения оптимизационных переменных в пространстве с использованием случайных направлений. Одно из преимуществ этих методов – достаточная простота, методы обладают большим спектром возможных направлений движения.

Рассмотрим алгоритм поиска *с линейной стратегией*: определив направление, в котором целевая функция уменьшается, не меняем направление до тех пор, пока оно не приведет к увеличению целевой функции.

Алгоритм поиска с ограничением на число неудачных попыток

- Шаг 0.** Задать $\varepsilon > 0$ параметр точности, $\lambda > 0$ начальный шаг,
 $\gamma > 1$ - коэффициент уменьшения шага, N – предельное число неудачных попыток (чаще всего $N=3n$), x – начальная точка поиска.
Вычислить $f(x)$.
- Шаг 1.** Положить счетчик неудачных попыток $j = 1$.
- Шаг 2.** Получить реализацию единичного вектора ξ , имеющего равномерное распределение.
- Шаг 3.** Найти пробную точку $y = x + \lambda \xi$, вычислить $f(y)$.
- Шаг 4.** Если $f(y) < f(x)$, то положить $x = y$, $f(x) = f(y)$, положить $j = 1$ и перейти к шагу 3.
- Шаг 5.** Положить $j = j + 1$, если $j \leq N$, то перейти к шагу 2.
- Шаг 6.** Если $\lambda < \varepsilon$, то поиск завершить, полагая $x^* = x$, иначе уменьшить шаг поиска ($\lambda = \lambda / \gamma$) и перейти к шагу 1.

Стратегия линейного поиска хороша вдали от оптимума. Вблизи оптимума более целесообразна нелинейная стратегия, при которой случайная смена направлений не зависима от результата. Изменение алгоритма следующее: на шаге 4 вместо перехода к шагу 3, необходимо перейти к шагу 2.

Алгоритм случайного поиска наилучшей пробы.

- Шаг 0.** Задать $\varepsilon > 0$ параметр точности, $\lambda > 0$ начальный шаг,
 $\gamma > 1$ коэффициент уменьшения шага, N – предельное число неудачных попыток (чаще всего $N=3n$), x – начальная точка поиска.
Вычислить $f(x)$.
- Шаг 1.** Положить счетчик неудачных попыток $j = 1$.
- Шаг 2.** Получить m реализаций единичного вектора ξ , имеющих равномерное распределение $\xi_1, \xi_2, \dots, \xi_m$, $m < n$.
- Шаг 3.** Найти пробные точки $y^i = x + \lambda \xi^i$, $i = 1, \dots, m$,

вычислить $f(y) = \min_i f(y^i)$.

Шаг 4. Если $f(y) < f(x)$, то положить $x = y$, $f(x) = f(y)$,
положить $j = 1$ и перейти к шагу 3.

Шаг 5. Положить $j = j + 1$, если $j \leq N$, то перейти к шагу 2.

Шаг 6. Если $\lambda < \varepsilon$, то поиск завершить, полагая $x^* = x$, иначе уменьшить шаг поиска ($\lambda = \lambda / \gamma$) и перейти к шагу 1.

Задание для лабораторной работы

Цель: ознакомиться с методами поиска минимума функции двух переменных в оптимизационных задачах без ограничений.

План работы: найти минимум функции $u = f(x, y)$ на области определения функции:

1. С использованием программного обеспечения исследовать каждый из алгоритмов на заданной тестовой функции, осуществляя спуск из различных исходных точек (не менее трех). Исследовать сходимость алгоритма, фиксируя точность определения минимума, количество итераций метода и количество вычислений функции в зависимости от задаваемой точности поиска. Результатом выполнения данного пункта должны быть выводы об объёме вычислений в зависимости от задаваемой точности и начального приближения.
2. Построить траекторию спуска различных алгоритмов из одной и той же исходной точки с одинаковой точностью. В отчете наложить эту траекторию на рисунок с линиями равного уровня заданной функции.

Варианты 1-5

Найти минимум функции $f(x) = A - (x - a)e^{-(x-a)} - (y - b)e^{-(x-b)}$. Параметры функции заданы в табл. 1.

Варианты 5-10

Найти минимум функции

$$f(x, y) = A - \exp \left\{ - \frac{1}{(10 - r^2)} \left(\frac{(x - a)^2}{c^2} - \frac{2r(x - a)(x - b)}{cd} + \frac{(x - b)^2}{d^2} \right) \right\}.$$

Параметры функции заданы в табл. 2.

Вариант 11-15

Найти минимум функции

$$f(x, y) = 100 - \frac{A_1}{1 + \left(\frac{x - a_1}{b} \right)^2 + \left(\frac{y - c_1}{d} \right)^2} - \frac{A_2}{1 + \left(\frac{x - a_2}{b} \right)^2 + \left(\frac{y - c_2}{d} \right)^2}$$

Параметры функции заданы в табл. 3.

Таблица 1

Номер варианта	A	a	b	Методы поиска для самостоятельной реализации
1	20	1	2	Методы Гаусса-Зейделя, Дэвидона – Флетчера – Пауэлла
2	10	2	3	Методы Хука и Дживса, наискорейшего спуска (Коши)
3	10	3	1	Методы Дэвидона Флетчера Пауэлла, случайного поиска наилучшей пробы
4	20	3	2	Методы поиска по симплексу, сопряженных градиентов Поллака Райвера
5	30	2	1	Методы наискорейшего спуска, Розенброка

Таблица 2

Номер варианта	A	a	b	c	d	r	Методы поиска для самостоятельной реализации
6	20	1	2	3	3	1	Методы Ньютона, случ. поиска с ограничением на число неудачных попыток
7	20	3	1	1	2	3	Методы Пирсона, Хука Дживса
8	30	2	2	2	1	2	Методы вращающихся координат (Розенброка), Бройдена Флетчера Шенно
9	20	3	2	1	1	-1	Методы сопряженных градиентов (Полака - Райвера); Гаусса Зейделя (покоординатный спуск)
10	1	2	3	2	1	-	Методы сопряженных градиентов Флетчера

Номер варианта	A	a	b	c	d	r	Методы поиска для самостоятельной реализации
	0					2	Ривса, Марквардта

Таблица 3

Номер варианта	A_1	A_2	a_1	a_2	b	c_1	c_2	d	Методы поиска для самостоятельной реализации
11	2	1	1	2	2	1	1	3	Методы Девидона Флетчера Пауэлла, Нелдера Мида
12	3	1	3	3	2	2	1	1	Методы Бroyдена Флетчера Шенно, случ. поиска с ограничением на число неудачных попыток
13	2	1	3	2	2	1	1	2	Методы Ньютона, наискорейшего спуска
14	3	1	3	3	2	2	2	1	Методы Бroyдена Флетчера Шенно, Розенброка с минимизацией по направлению
15	3	2	4	2	1	4	2	3	Методы Гаусса Зейделя, Мак Кормика,

Содержание отчета

Отчет должен содержать:

- титульный лист;
- цель работы;
- задание;
- таблицы с результатами проведенных исследований, где должны быть отражены начальное приближение \bar{x}_0 (не менее трех), задаваемая точность, количество итераций, число вычислений целевой функции, найденная точка и значение функции в ней;
- траекторию спуска различных алгоритмов из одной и той же исходной точки с одинаковой точностью (наложить эту траекторию на рисунок с линиями равного уровня заданной функции);
- а также выводы о сходимости алгоритмов в зависимости от точности и начального приближения с указанием преимуществ и недостатков.

Контрольные вопросы

1. Что такое математическая модель объекта оптимизации?

2. Сформулируйте математическую постановку задачи оптимизации.
3. Дайте определение оптимального решения задачи оптимизации.
4. Какая последовательность называется релаксационной?
5. Сформулируйте идею методов прямого поиска нулевого порядка.
6. Каким образом выбирают направления и параметр шага в методе Гаусса Зейделя?
7. В чем заключается этап исследующего поиска в методе Хука Дживса?
8. Как выбирается ускоряющий множитель в этапе поиска по образцу в методе Хука Дживса?
9. В чем отличия метода Розенброка и метода Хука Дживса? Каковы условия применения метода Розенброка?
10. Дайте определение регулярного симплекса.
11. Как строится новый симплекс на основе базового?
12. В чем отличие процедур отражения и сжатия вовнутрь?
13. Какие векторы называются H сопряженными?
14. Сформулируйте основную идею метода Пауэлла.
15. Как выбирается направление и параметр шага в методе наискорейшего спуска?
16. В чем отличие выбора направлений метода сопряженных градиентов и метода сопряженных направлений? Как выбирается параметр шага в методе Флетчера Ривса?
17. Как выбирается параметр шага в методе Поллака Райвера?
18. Какие направления поиска называются ньютоновскими?
19. В чем состоит основной принцип модификаций метода Ньютона?
20. Как строится релаксационная последовательность в методе Марквардта?
21. Какие методы называются квазиньютоновскими? Как строится релаксационная последовательность в этих методах?
22. Как аппроксимируют обратную матрицу Гессе в методах Пирсона? Какая матрица выбирается в качестве начальной?
23. В чем отличие методов Пирсона и метода Дэвидона Флетчера Пауэлла?

24. В чем отличие стратегий линейного и нелинейного поиска в методах случайного поиска?

Библиографический список

1. *Алексеев В. М.* Сборник задач по оптимизации / В. М. Алексеев, Э. М. Галлеев, В. М. Тихомиров. – М. : Наука, 1984. – 287 с.
2. *Аттетков А. В.* Методы оптимизации : учеб. для вузов / А. В. Аттетков, С. В. Галкин, В. С. Зарубин ; под ред. В. С. Зарубина, А. П. Крищенко. – 2-е изд., стереотип. – М. : Изд-во МГТУ им. Н. Э. Баумана, 2003. – 440 с. (Сер. Математика в техническом университете; Вып. XIV).
3. *Ашманов С. А.* Теория оптимизации в задачах и упражнениях / С. А. Ашманов, А. В. Тимохов. – М. : Наука, 1991. – 448 с.
4. *Глебов Н. И.* Методы оптимизации : учеб. пособие / Н. И. Глебов, Ю. А. Кочетов, А. В. Плясунов. – Новосибирск : Из-во Новосибирского ун-та, 2000. – 105 с.
5. *Лесин В.В.*, Основы методов оптимизации: учебник / В. В. Лесин, Ю.П. Лисовец. – М. : Изд-во МАИ, 1995. – 341 с.

Оглавление

1. Постановка задач многомерной оптимизации	4
2. Методы прямого поиска	6
2.1. Общая характеристика методов нулевого порядка	6
2.2. Метод Гаусса Зейделя (покоординатный спуск).....	7
2.3. Метод Хука Дживса с использованием одномерной оптимизации	9
2.4. Метод вращающихся направлений (Розенброка)	11
2.5. Метод поиска по симплексу. Метод Нелдера Мида	14
2.6. Методы сопряженных направлений (Пауэлла)	19
3. Методы первого порядка	22
3.1. Метод наискорейшего спуска (Коши)	22
3.2. Методы сопряженных градиентов	24
4. Методы 2-го порядка (Ньютоновские методы)	26
4.1 Методы, использующие производные второго порядка	26
4.2. Модификации метода Ньютона	27
4.3. Метод Ньютона Рафсона	30
4.4. Метод Марквардта	31
5. Методы переменной метрики	32
5.1. Квазиньютоновские методы: общая характеристика	32
5.2. Методы Пирсона	33
5.3. Метод Дэвидона Флетчера Пауэлла	34
5.4 Метод Бройдена Флетчера Шенно	36
5.5. Методы Пауэлла и Мак Кормика	36
6. Методы случайного поиска	36
Задание для лабораторной работы	38
Контрольные вопросы	40
Библиографический список	42

МЕТОДЫ МНОГОМЕРНОЙ ОПТИМИЗАЦИИ

Методические указания и задания к выполнению лабораторных работ по дисциплине
«Методы оптимизации» для студентов направления «Прикладная математика»

Попова Татьяна Михайловна

Главный редактор *Л. А. Суевалова*

Редактор *Т. Ф. Шейкина*

Подписано в печать 08.02.2012 Формат 60х84 1/16.

Бумага писчая. Гарнитура «Таймс». Печать цифровая.

Усл. печ. л. 2,56. Тираж 200 экз. Заказ 36

Издательство Тихоокеанского государственного университета.

680035, Хабаровск, ул. Тихоокеанская, 136.

Отдел оперативной полиграфии

Издательства Тихоокеанского государственного университета.

680035, Хабаровск, ул. Тихоокеанская, 136.