Московский государственный университет
имени М. В. Ломоносова

Факультет вычислительной математики и кибернетики
Кафедра исследования операций

Практикум на ЭВМ

Отчет по основному практическому заданию

студента группы 411

Фостенко Олега Андреевича

Москва, 2025

Содержание

1 Постановка задачи

Метод возможных направлений является методом решения задачи оптимизации с ограничениями:

$$f(x) \to \min,\ x \in P$$

$$P = \{x \in \mathbb{R}^n\colon\ g_j(x) \leq 0,\ j = 1, \ldots, m\}$$

Здесь функции $f,\ g_j \in C^1(\mathbb{R}^n),\ j = 1, \ldots, m$, то есть являются скалярными функциями векторного аргумента и предполагаются дифференцируемыми с непрерывными частными производными [2].

2 Теоретическое описание метода

В методе возможных направлений выбирается начальная точка, удовлетворяющая всем ограничениям, после чего осуществляется переход к следующей точке согласно итерационной схеме:

$$x_{i+1} = x_i + \lambda s_i,$$

где x_i — начальная точка на i-й итерации, s_i — допустимое направление движения из начальной точки, λ — длина шага, а x_{i+1} — конечная точка, полученная после завершения i-й итерации.

Значение λ всегда выбирается таким образом, чтобы точка x_{i+1} оставалась в допустимой области. Направление поиска s_i определяется так, чтобы (1) небольшое перемещение в этом направлении не нарушало ни одного ограничения и (2) значение целевой функции можно было уменьшить в этом направлении.

Новая точка x_{i+1} принимается в качестве начальной точки для следующей итерации, и вся процедура повторяется до тех пор, пока не будет получена точка, для которой невозможно найти направление, удовлетворяющее обоим свойствам (1) и (2). В общем случае такая точка является условным локальным минимумом задачи. Этот локальный минимум не обязательно будет глобальным, за исключением случая задачи выпуклого программирования.

Направление, удовлетворяющее свойству (1), называется *допустимым*, а направление, удовлетворяющее обоим свойствам (1) и (2), называется *используемым допустимым.*

Направление s является *допустимым* в точке x_i, если для активных в точке x_i ограничений $(j : g_j(x_i) = 0)$ оно удовлетворяет соотношению:

$$\frac{d}{d\lambda} g_j(x_i + \lambda s)\Big|_{\lambda=0} = s^T \nabla g_j(x_i) \le 0$$

где равенство может выполняться, если ограничение, например, вогнуто (в локальном рассмотрении).

Вектор s будет *используемым допустимым* направлением, если он удовлетворяет соотношениям:

$$\frac{d}{d\lambda} f(x_i + \lambda s)\Big|_{\lambda=0} = s^T \nabla f(x_i) < 0$$
$$\frac{d}{d\lambda} g_j(x_i + \lambda s)\Big|_{\lambda=0} = s^T \nabla g_j(x_i) \le 0$$

Шагая в таком направлении хотя бы на небольшой λ, можно уменьшить целевую функцию.

Предлагается следующий алгоритм, реализующий метод возможных направлений [1]:

Алгоритм 1

1. Выбираем начальную допустимую точку x_1 и малые числа $\varepsilon_1, \varepsilon_2$ для проверки сходимости. Вычисляем $g_j(x_1)$, $j = 1, 2, \ldots, m$. Устанавливаем номер итерации $i = 1$.

2. Если $g_j(x_i) < 0$, $j = 1, 2, \ldots, m$ (т.е. x_i — внутренняя точка), задаем направление поиска:

$$s_i = -\nabla f(x_i)$$

Нормируем s_i подходящим образом и переходим к шагу 5. Если хотя бы одно $g_j(x_i) = 0$, переходим к шагу 3.

3. Находим используемое допустимое направление s, решая задачу с ограничениями:

$$\begin{aligned}
&-\alpha \to \min \\
&s^T \nabla g_j(x_i) + \theta_j \alpha \le 0, \quad j = 1, 2, \ldots, p \\
&s^T \nabla f(x_i) + \alpha \le 0 \\
&-1 \le s_k \le 1, \quad k = 1, 2, \ldots, n
\end{aligned}$$

где p ограничений считаются активными в точке x_i (остальные ограничения неактивны), θ_j — некоторые положительные константы, α — дополнительная переменная.

4. Если найденное $\alpha^* \approx 0$ ($\alpha^* \leq \varepsilon_1$), завершаем вычисления, принимая $x_{opt} \approx x_i$. Если $\alpha^* > \varepsilon_1$, переходим к шагу 5, полагая $s_i = s$.

5. Находим подходящую длину шага λ_i вдоль направления s_i и получаем новую точку:

$$x_{i+1} = x_i + \lambda_i s_i$$

6. Проверяем сходимость метода. Если

$$\|x_i - x_{i+1}\| \leq \varepsilon_2$$

завершаем итерации, принимая $x_{opt} \approx x_{i+1}$. Иначе переходим к шагу 7.

7. Увеличиваем номер итерации: $i = i + 1$ и повторяем с шага 2.

При применении данного алгоритма необходимо учитывать несколько аспектов. Они связаны с (1) нахождением подходящего используемого допустимого направления (s), (2) выбором подходящего размера шага вдоль направления s и (3) сходимостью.

3 Практические аспекты

3.1 Замечания к алгоритму

Для определения активных ограничений может быть недостаточно сравнивать ноль и $g_j(x)$, так как точность машинного представления ограничена, вследствие чего результат работы представленного алгоритма может не быть оптимумом. Поэтому вводится положительная константа ε_4: если $g_j(x) > -\varepsilon_4$, то ограничение считается активным в точке.

На шаге №3 можно выбирать направление и по-другому: «на глаз» (например, допустимые направления могут быть хорошо видны в случае функции двух переменных) или случайным образом (с соответствующей проверкой того, является полученное направление *используемым допустимым* или нет).

В случае выбора шага посредством решения задачи ЛП (как показано в алгоритме 1) θ_j на шаге №3 в общем случае можно варьировать, принимая их некоторыми положительными константами. Эти параметры позволяют задавать, насколько точка x_{i+1} будет удалена от активных в точке x_i ограничений. Для простоты берется $\theta_j = 1$ для всех j.

Выбор длины шага на шаге №5 алгоритма можно производить различными способами. В рамках практической реализации предлагается следующий способ (на каждой итерации):

Алгоритм 2

1. Выбираем начальную длину шага $\lambda = \lambda_0$, а также некоторую положительную константу ε_3 (одинаковую для всех шагов основного алгоритма (1)).

2. Вычисляем $g_j(x_i + \lambda s_i)$ для $j = 1, \ldots, m$.

3. Если не выполнено $g_j(x_i + \lambda s_i) \leq 0$, $j = 1, 2, \ldots, m$, перейти на шаг 6.

4. Если $||\lambda s_i|| \leq \varepsilon_2$ или $f(x_i) - f(x_i + \lambda s_i) \geq \varepsilon_3$, переходим на следующий шаг (5), иначе переходим на шаг 6.

5. Завершаем итерации, принимая $x_{i+1} = x_i + \lambda s_i$.

6. Присваиваем $\lambda = \frac{\lambda}{2}$. Повторяем с шага 2.

В результате работы этого алгоритма (2) новая точка будет вызывать критерий останова алгоритма 1, если функция не уменьшилась более чем на ε_3 до того, как шаг стал чрезмерно маленьким, при выполнении ограничений. В алгоритме 2 λ_0 берется достаточно большим по сравнению с константами ε_i.

В рамках основного алгоритма (1) также можно ввести максимальное число итераций, что и было сделано в практической реализации. Это позволит ограничить число шагов в случае задач без оптимума (например, допустимое множество не ограничено, а функция убывает на бесконечности достаточно быстро). Алгоритм 1 при применении алгоритма 2 и существовании глобального минимума рано или поздно завершается как минимум потому, что для продолжения итераций алгоритма 1 функция должна быть уменьшена более чем на фиксированное число при переходе от x_i к x_{i+1}.

3.2 Программная реализация

В рамках практического задания была написана программная реализация метода возможных направлений с использованием языка программирования Python и стандартных библиотек. Исходный код проекта, включая результаты экспериментов, доступен в GitHub-репозитории[1].

4 Эксперименты

Пример 1. Рассмотрим задачу минимизации функции

$$f(x, y) = x^2 + y^2$$

при ограничениях

$$g_1(x, y) = 10x - y^2 \leq 0$$

$$g_2(x, y) = -10x - y^3 \leq 0$$

$$g_3(x, y) = -10x - 1 + y^2 \leq 0$$

В данной задаче есть единственный глобальный оптимум:

$$x^* = (0, 0)$$

$$f(x^*) = 0$$

Далее представлены результаты экспериментов с перечислением значений параметров и результатов работы алгоритма.

[1] https://github.com/oscar-foxtrot/prac-io-0/tree/main/year_4_assignment_1 (дата обращения: 26.10.2025)

Параметры: $\varepsilon_1 = 10^{-6}$, $\varepsilon_2 = 10^{-6}$, $\varepsilon_3 = 10^{-6}$, $\varepsilon_4 = 10^{-3}$, $\lambda_0 = 10$, макс. итераций=500			
Начальная точка	**Итерации**	x^*	$f(x^*)$
$(0.5 - 0.001, \sqrt{5} + 0.001)$	27	$(6.48 \times 10^{-5}, 3.15 \times 10^{-2})$	9.89×10^{-4}
$(1.5 - 0.001, \sqrt{15} + 0.001)$	67	$(5.79 \times 10^{-5}, 3.80 \times 10^{-2})$	1.44×10^{-3}
$(10 - 0.001, \sqrt{100} + 0.001)$	466	$(3.00 \times 10^{-5}, 3.27 \times 10^{-2})$	1.07×10^{-3}

Таблица 1: Результаты оптимизации серии 1

Параметры: $\varepsilon_1 = 10^{-9}$, $\varepsilon_2 = 10^{-9}$, $\varepsilon_3 = 10^{-9}$, $\varepsilon_4 = 10^{-6}$, $\lambda_0 = 10$, макс. итераций=5000			
Начальная точка	**Итерации**	x^*	$f(x^*)$
$(0.5 - 0.001, \sqrt{5} + 0.001)$	63	$(7.72 \times 10^{-8}, 1.03 \times 10^{-3})$	1.07×10^{-6}
$(1.5 - 0.001, \sqrt{15} + 0.001)$	133	$(5.01 \times 10^{-8}, 1.08 \times 10^{-3})$	1.16×10^{-6}
$(10 - 0.001, \sqrt{100} + 0.001)$	818	$(1.31 \times 10^{-7}, 1.50 \times 10^{-3})$	2.24×10^{-6}

Таблица 2: Результаты оптимизации серии 2

Параметры: $\varepsilon_1 = 10^{-9}$, $\varepsilon_2 = 10^{-9}$, $\varepsilon_3 = 10^{-9}$, $\varepsilon_4 = 10^{-6}$, $\lambda_0 = 0.01$, макс. итераций=5000			
Начальная точка	**Итерации**	x^*	$f(x^*)$
$(0.5 - 0.001, \sqrt{5} + 0.001)$	5001	$(8.65 \times 10^{-7}, 3.72 \times 10^{-3})$	1.38×10^{-5}
$(1.5 - 0.001, \sqrt{15} + 0.001)$	5001	$(0.471, 2.171)$	4.936
$(10 - 0.001, \sqrt{100} + 0.001)$	5001	$(8.592, 9.269)$	159.749

Таблица 3: Результаты оптимизации серии 3

Пример 2. Рассмотрим задачу минимизации функции

$$f(x, y) = 20 + (x^2 - 10\cos(2\pi x)) + (y^2 - 10\cos(2\pi y))$$

при ограничениях

$$g_1(x, y) = (x - 1)^2 + (y - 1)^2 - 0.0625 \le 0$$

$$g_2(x, y) = y - x + 0.25 \le 0$$

Глобальный оптимум:

$$x^* \approx (1.11788, 0.867884)$$

$$f(x^*) \approx 7.87488$$

Параметры: $\varepsilon_1 = 10^{-6}$, $\varepsilon_2 = 10^{-6}$, $\varepsilon_3 = 10^{-6}$, $\varepsilon_4 = 10^{-3}$, $\lambda_0 = 10$, макс. итераций=500			
Начальная точка	**Итерации**	x^*	$f(x^*)$
$(1.25 - 0.001, 1 - 0.001)$	7	$(1.119, 0.868)$	7.910
$(1.25 - 0.1, 1 - 0.2)$	70	$(1.115, 0.865)$	7.905
$(1.25 - 0.05, 1 - 0.05)$	153	$(1.121, 0.870)$	7.919

Таблица 4: Результаты оптимизации серии 1

Параметры: $\varepsilon_1 = 10^{-9}$, $\varepsilon_2 = 10^{-9}$, $\varepsilon_3 = 10^{-9}$, $\varepsilon_4 = 10^{-6}$, $\lambda_0 = 10$, макс. итераций=5000			
Начальная точка	**Итерации**	x^*	$f(x^*)$
$(1.25 - 0.001, 1 - 0.001)$	867	$(1.118, 0.868)$	7.875
$(1.25 - 0.1, 1 - 0.2)$	384	$(1.117797, 0.867797)$	7.874924
$(1.25 - 0.05, 1 - 0.05)$	707	$(1.117971, 0.867971)$	7.874899

Таблица 5: Результаты оптимизации серии 2

Как видно по результатам расчетов, представленных в виде таблиц 1–5, метод успешно справляется с численным решением приведенных задач, находя точку, близкую к оптимальному решению.

Параметры точности метода существенно влияют на точность, а также на число итераций (как правило, выше точность - больше итераций требуется). Выбор длины шага также существенен - слишком маленький шаг приводит к проблемам со сходимостью, а слишком большой - к лишним итерациям в методе поиска шага, что влияет на время работы программы.

Выбор начальной точки также влияет на скорость сходимости метода, однако при существовании оптимума метод находит приближенное решение при различных начальных приближениях.

Заключение

В ходе работы был успешно реализован и протестирован метод возможных направлений. Эксперименты подтвердили, что метод демонстрирует надежную сходимость на различных задачах нелинейного программирования при корректном выборе параметров.

Эксперименты показали, что метод не является самым эффективным. Он, однако, имеет вариации, очень простые в реализации, в чем состоит одно из основных его преимуществ.

Список литературы

1. *Rao S.* Engineering Optimization: Theory and Practice. — 4th edition. — John Wiley & Sons, Inc., 2009. — P. 393–403.

2. *Zoutendijk G.* Methods of Feasible Directions: A Study in Linear and Non-Linear Programming. — Elsevier Publishing Company, 1960. — P. 78–79.