1 Постановка задачи

Пусть дана задача двумерной минимизации $\phi_0(x) = 4x_1 + x_2 + 4\sqrt{1 + 3x_1^2 + x_2^2}$. Необходимо:

1. Ввести ограничения вида $\phi_i(x) \le 0$, где $i = \overline{1, m}, m = 4$, которые могут иметь линейный вид, таким образом, чтобы оптимальная точка решения данной задачи находилась:

   - внутри введенной области

$$\begin{cases} x_1 \le 0 \\ x_2 \le 0 \\ x_1 - 2x_2 - 1 \le 0 \\ -x_1 - x_2 - 1 \le 0 \end{cases} \tag{1}$$

   - на границе введенной области

$$\begin{cases} x_1 \le 0 \\ x_2 \le 0 \\ x_1 - 2x_2 - 1 \le 0 \\ -x_1 - x_2 - 0.7504 \le 0 \end{cases} \tag{2}$$

2. Решить задачу условной минимизации методом всевозможных направлений Зойтендейка

3. Проиллюстрировать поведение алгоритма

2 Исследование применимости метода

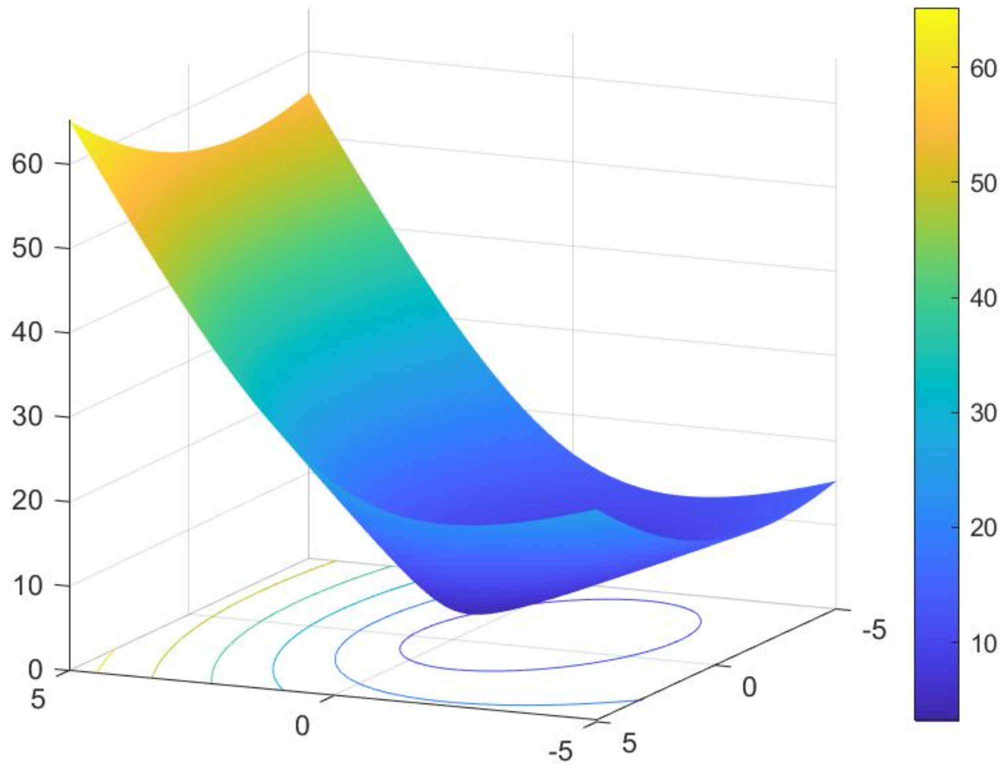


Рис. 1: График исходной функции

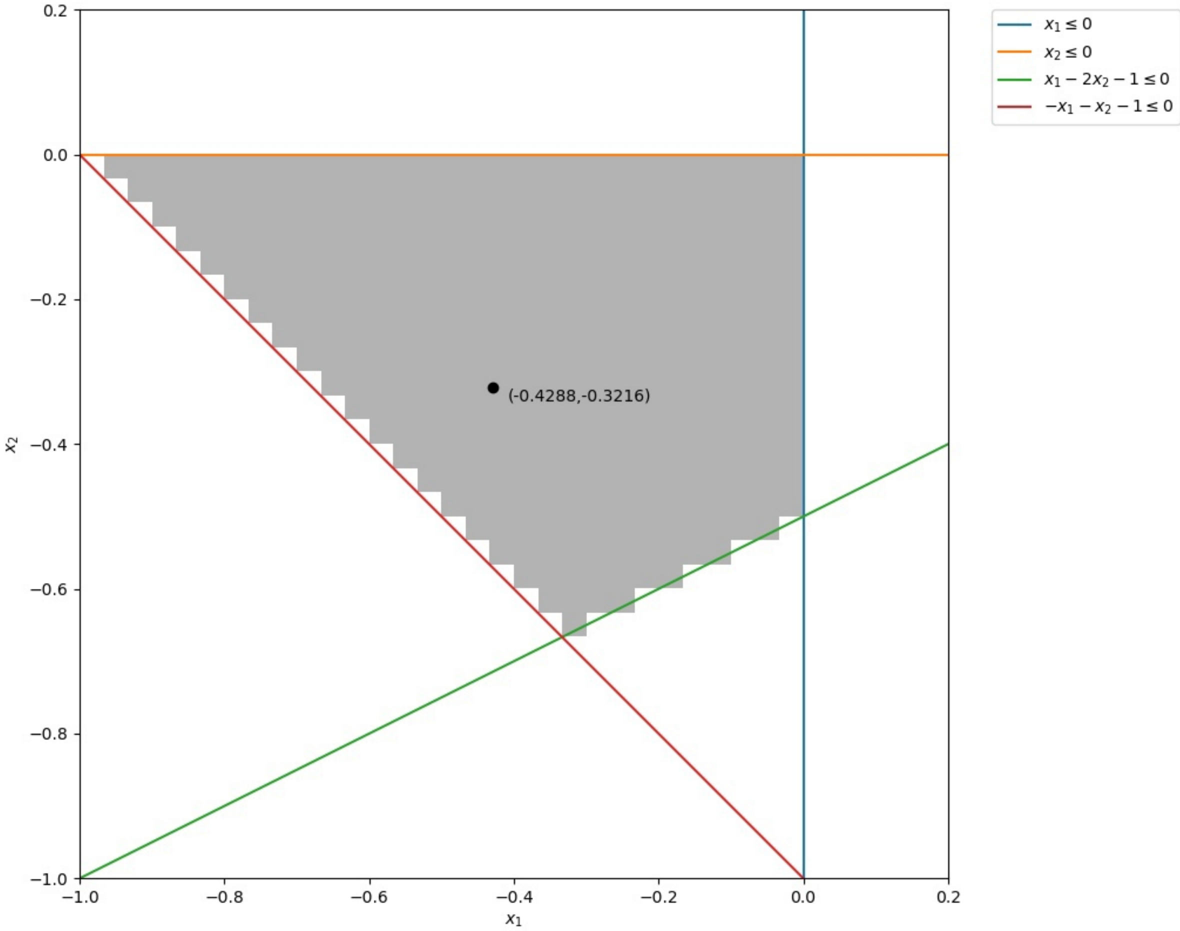


Рис. 2: Область для внутренней точки

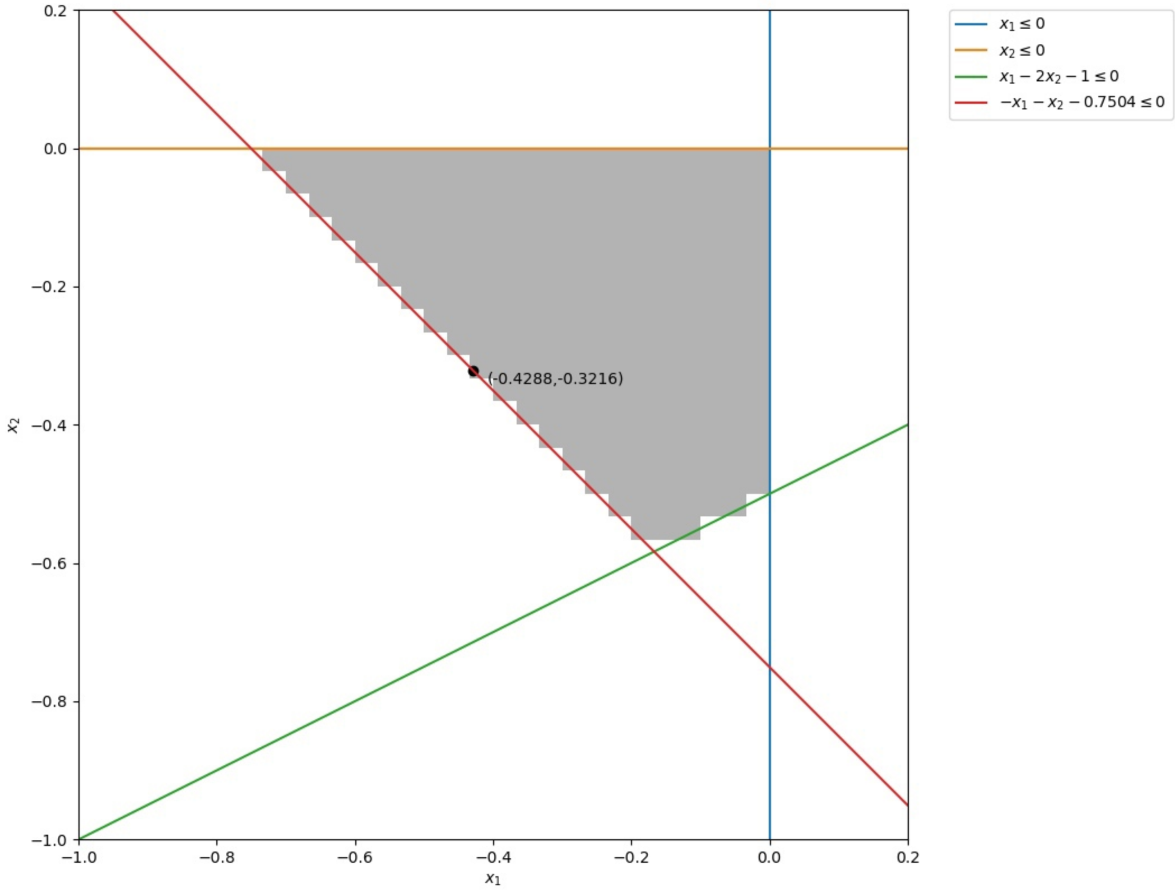


Рис. 3: Область для точки на границе

```
Проверка условия Слейтера ----> решим задачу минимизации
=>
Найденная точка: x= [0. 0.]

Проверка условия Слейтера ----> решим задачу минимизации
(граничные условия)
=>
Найденная точка: x= [0. 0.]

Проверка условия Слейтера ----> решим СЛАУ:
x1 <= 0
x2 <= 0
x1 - 2x2 - 1<= 0
-x1 - x2 - 1 <= 0
=>
Найденная точка: x = [0,0]
```

Рис. 4: Доказательство существования непустой области, образованной введенными ограничениями

Пусть $\varphi_0(x)$ – выпуклая функция, $S = \{X \mid \varphi_i(x) \le 0$ при $i = \overline{1,m}\} \subset R^n$ – выпуклое множество. Тогда задача имеет вид $\min \varphi_0(x)$, где $x \in S$

Основные условия применимости:

1. $\varphi_i(x)$ – выпуклые для $i = \overline{0,m}$
2. $\varphi_i(x)$ – непрерывно дифференцируемые для $i = \overline{0,m}$
3. $\|\nabla\varphi_i(x)\| \le K$ для $i = \overline{0,m}$
4. $\|\nabla\varphi_i(x) - \nabla\varphi_i(y)\| \le R\|x - y\|$ $\forall x, y \in R^n$, $i = \overline{0,m}$
5. S – компактное
6. Условие Слейтера: $\exists \bar{x} : \varphi_i(\bar{x}) < 0$ для $i = \overline{1,m}$

Введем ограничения такого вида, чтобы оптимальная точка $x^* = (-0,4288; -0,3213)$ целевой функции была внутренней (рис 2):

$$\begin{cases} x_1 \le 0 \\ x_2 \le 0 \\ x_1 - 2x_2 - 1 \le 0 \\ -x_1 - x_2 - 1 \le 0 \end{cases}$$

1. Все ограничения линейные, поэтому функции выпуклые и вогнутые одновременно. Выпуклость функции цели доказана в предыдущей работе.

2. Очевидно, что все функции непрерывно дифференцируемы

3. $\nabla\varphi_1(x) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \Rightarrow \|\nabla\varphi_1(x)\| = 1$

$\nabla\varphi_2(x) = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \Rightarrow \|\nabla\varphi_2(x)\| = 1$

$\nabla\varphi_3(x) = \begin{pmatrix} 1 \\ -2 \end{pmatrix} \Rightarrow \|\nabla\varphi_3(x)\| = \sqrt{5}$

$\nabla\varphi_4(x) = \begin{pmatrix} -1 \\ -1 \end{pmatrix} \Rightarrow \|\nabla\varphi_4(x)\| = \sqrt{2}$

$$\nabla\varphi_0(x) = \begin{pmatrix} 4 + \frac{12x_1}{\sqrt{1+3x_1^2+x_2^2}} \\ 1 + \frac{4x_2}{\sqrt{1+3x_1^2+x_2^2}} \end{pmatrix} \Rightarrow \|\nabla\varphi_0(x)\| =$$

$$= \sqrt{16 + \frac{96x_1}{\sqrt{1+3x_1^2+x_2^2}} + \frac{144x_1^2}{1+3x_1^2+x_2^2} + 1 + \frac{8x_2}{\sqrt{1+3x_1^2+x_2^2}} + \frac{16x_2^2}{1+3x_1^2+x_2^2}} = \sqrt{17 + \frac{8(x_2+12x_1)}{\sqrt{1+3x_1^2+x_2^2}} + \frac{16(9x_1^2+x_2^2)}{1+3x_1^2+x_2^2}}$$

Указанная функция будет максимальна, если x_i принимает максимальное значение

$\nexists$ точку $(0,0) \Rightarrow \|\nabla\varphi_0(x)\| = \sqrt{17}$

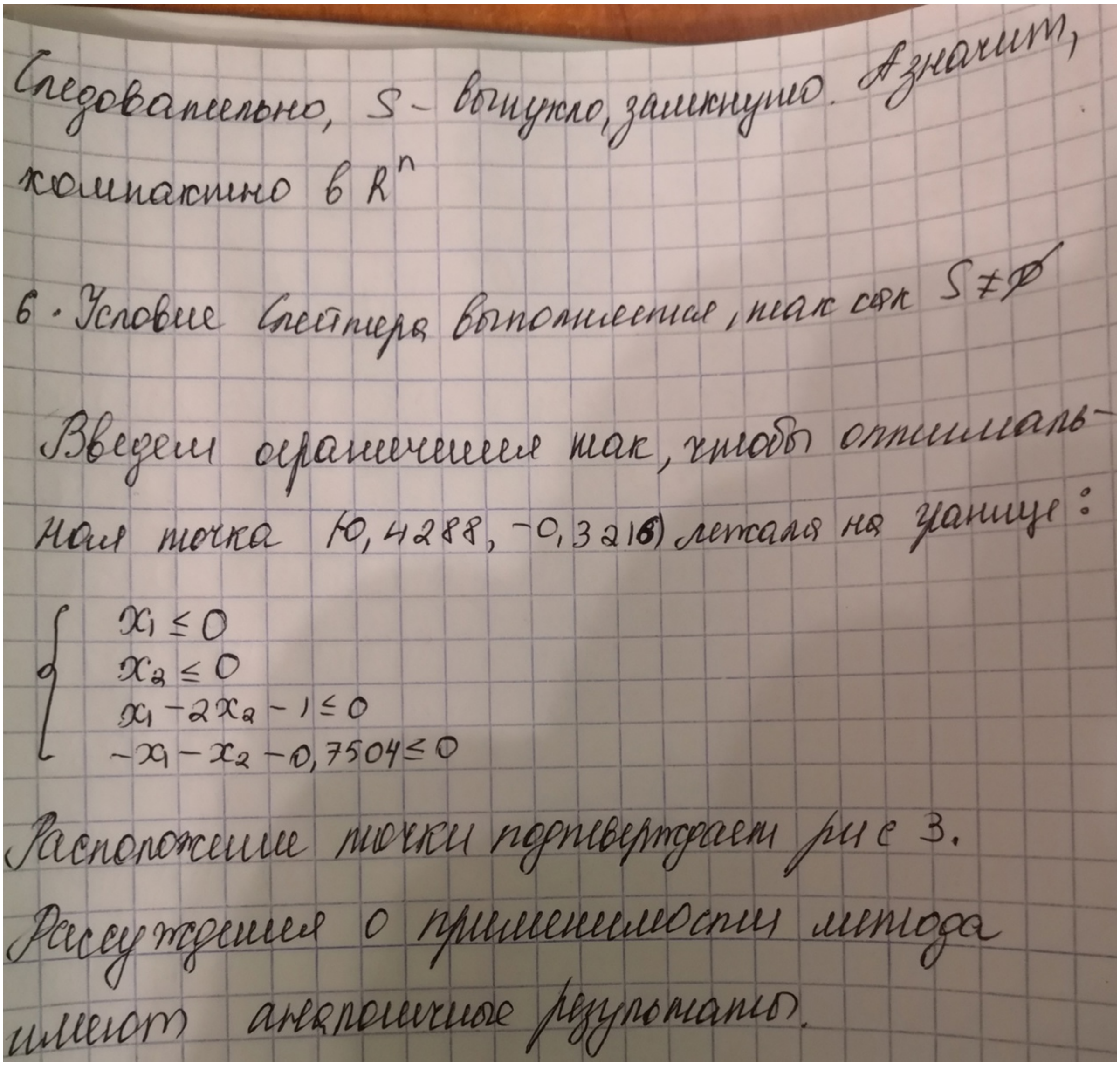
$\Rightarrow \|\nabla\varphi_i(x)\| \leq \sqrt{17}$

4. Функция цели удовлетворяет условию Липшица, что доказано в предыдущей работе

$\|\nabla\varphi_i(x) - \nabla\varphi_i(y)\| = 0 \quad \forall x, y \in S, \ i = \overline{1,m}$

$\Rightarrow 0 \leq R\|x-y\| \Rightarrow$ удовлетворяют условию Липшица

5. S – ограничено, так как образовано пересечением конечного числа замкнутых полуплоскостей


Следовательно, S - выпукло, замкнуто. А значит, компактно в R^n

6. Условие Слейтера выполняется, так как $S \neq \varnothing$

Введем ограничения так, чтобы оптималь-ная точка $(0{,}4288, -0{,}3216)$ лежала на границе:

$$\begin{cases} x_1 \leq 0 \\ x_2 \leq 0 \\ x_1 - 2x_2 - 1 \leq 0 \\ -x_1 - x_2 - 0{,}7504 \leq 0 \end{cases}$$

Расположение точки подтверждает рис 3.

Рассуждения о применимости метода имеют аналогичные результаты.

3 Описание алгоритма

3.1 Метод всевозможных направлений Зойтендейка

3.1.1 Начальный этап

1. Выбрать совокупность параметров $\{\xi_i\}_0^m : \xi_i > 0$, использующихся для улучшения свойств сходимости задачи. В решении рекомендовано взять $\xi_i = 1$ для $\forall i = \overline{0, m}$

2. Выбрать параметр дробления $0 < \lambda < 1$. Рекомендовано взять для решения задачи $\lambda = \frac{1}{2}$

3. Выбрать начальное приближение $x_0 \in S$, где $S = \{X | \phi_i(x) \leq 0$ для $\forall i = \overline{1, m}\}$ и параметр η_0

4. Взять критерий близости к почти активным ограничениям $\delta_0 = -\eta_0$, где $\delta_0 > 0$, $J_{\delta_k}(x_k) = \{i \in M | -\delta_k \leq \phi_i(x_k) \leq 0\}$ - множество номеров почти активных ограничений для $M = \overline{1,m}$

3.1.2 Поиск начального приближения

1. Найти $min\eta < 0$ при условии $\phi_i(x) \leq \eta$, где $i = \overline{1,m}$

2. Если $x_0 : \phi_i(x_0) \leq 0$, то эта точка является допустимой точкой для исходной задачи и $\eta_0 = min\eta$

3. Иначе итерации проводятся до тех пор, пока точка не окажется в области рассмотрения задачи

3.1.3 Основной этап

1. Известны $x_k \in S$ и $\delta_k > 0$

2. Решить вспомогательную задачу линейного программирования симплекс-методом для определения направления спуска $s : min\eta$ при условиях

$$\begin{cases} \bigtriangledown^T \phi_0(x_k) * s \leq \eta\xi_0 \\ \bigtriangledown^T \phi_i(x_k) * s \leq \eta\xi_i \end{cases} \tag{3}$$

для $i \in J_{\delta_k}(x_k)$

3. Обозначить найденные $s_{\delta_k}(x_k) = s_k$ и $\eta_{\delta_k}(x_k) = \eta_k$

4. Если $\eta_k < -\delta_k$:

   - Делаем шаг α_k по выбранному направлению $x_{k+1} = x_k + \alpha_k s_k$ и $\delta_{k+1} = \delta_k$
   - Иначе шаг не делается, то есть $x_{k+1} = x_k$ и $\delta_{k+1} = \lambda\delta_k$

5. Закончить работу алгоритма при $\delta_k < \delta_{0k}$, где $-\delta_{0k} = max\phi_i(x_k)$ для $i \notin J_0(x_k)$ - не для активных ограничений, и $\eta_k = 0$. При этом $\delta_k < \varepsilon$

3.1.4 Выбор величины шага по принципу дробления

1. Положить $\alpha_k = \alpha_0 * \lambda^{ik}$, где $\alpha_0 = 1$

2. Выбрать α_k, удовлетворяющее условиям:

$$\begin{cases} \phi_0(x_k + \alpha_k * s_k) - \phi_0(x_k) \leq \xi_0\eta_k\alpha_k \\ \phi_i(x_k + \alpha_k * s_k) \leq 0, i = \overline{1,m} \end{cases} \tag{4}$$

4 Практическое решение задач

Для рассмотрения начальной стадии работы алгоритма взята точка $(-0.2, -0.4)$ и найдена допустимая точка для множества при $\eta_0 = -2.29289322$.

4.1 Решение задачи при внутренней оптимальной точке

Решение задачи многомерной минимизации при ограничениях такого вида, что оптимальная точка x^* находится внутри рассматриваемой области, с точностью $\varepsilon = 10^{-5}$:

x^*	(-0.42884464, -0.32163181)
ϕ_0^*	3.10912635
Число итераций	35

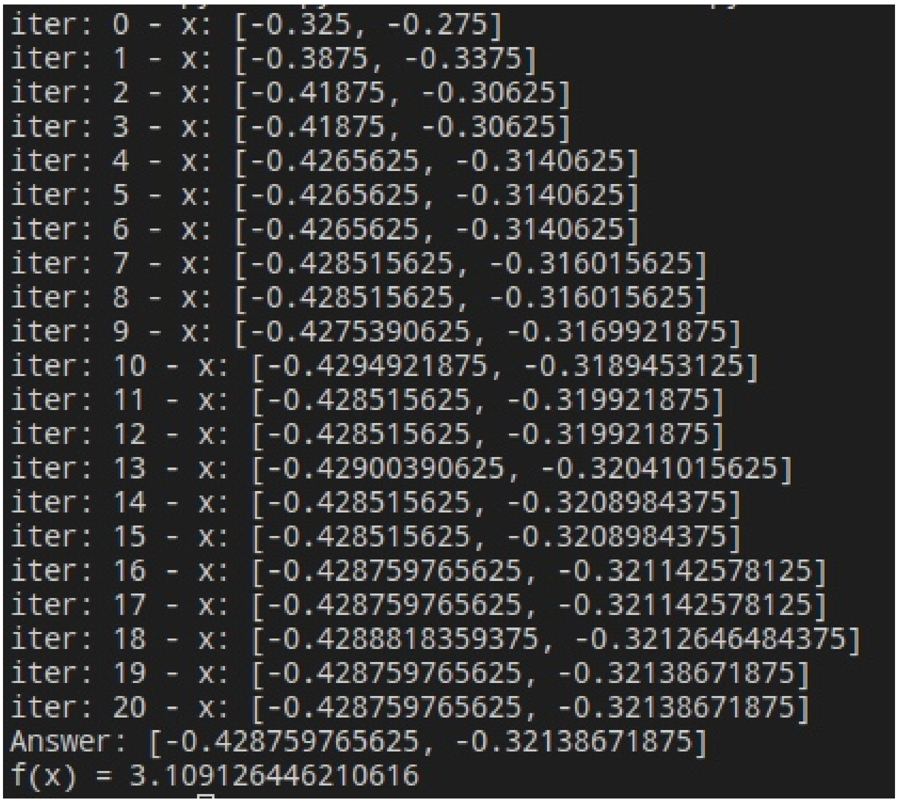
```
iter: 0 - x: [-0.325, -0.275]
iter: 1 - x: [-0.3875, -0.3375]
iter: 2 - x: [-0.41875, -0.30625]
iter: 3 - x: [-0.41875, -0.30625]
iter: 4 - x: [-0.41875, -0.30625]
iter: 5 - x: [-0.4265625, -0.3140625]
iter: 6 - x: [-0.4265625, -0.3140625]
iter: 7 - x: [-0.428515625, -0.316015625]
iter: 8 - x: [-0.428515625, -0.316015625]
iter: 9 - x: [-0.4275390625, -0.3169921875]
iter: 10 - x: [-0.4294921875, -0.3189453125]
iter: 11 - x: [-0.428515625, -0.319921875]
iter: 12 - x: [-0.428515625, -0.319921875]
iter: 13 - x: [-0.428515625, -0.319921875]
iter: 14 - x: [-0.42900390625, -0.32041015625]
iter: 15 - x: [-0.428515625, -0.3208984375]
iter: 16 - x: [-0.428759765625, -0.321142578125]
iter: 17 - x: [-0.428759765625, -0.321142578125]
iter: 18 - x: [-0.428759765625, -0.321142578125]
iter: 19 - x: [-0.4288818359375, -0.3212646484375]
iter: 20 - x: [-0.428759765625, -0.32138671875]
iter: 21 - x: [-0.4288818359375, -0.3215087890625]
iter: 22 - x: [-0.4288818359375, -0.3215087890625]
iter: 23 - x: [-0.42882080078125, -0.32156982421875]
iter: 24 - x: [-0.42882080078125, -0.32156982421875]
iter: 25 - x: [-0.428851318359375, -0.321600341796875]
iter: 26 - x: [-0.428851318359375, -0.321600341796875]
iter: 27 - x: [-0.4288360595703125, -0.3216156005859375]
iter: 28 - x: [-0.4288360595703125, -0.3216156005859375]
iter: 29 - x: [-0.42884368896484376, -0.32162322998046877]
iter: 30 - x: [-0.42884368896484376, -0.32162322998046877]
iter: 31 - x: [-0.4288475036621094, -0.3216270446777344]
iter: 32 - x: [-0.42884368896484376, -0.321630859375]
iter: 33 - x: [-0.42884368896484376, -0.321630859375]
iter: 34 - x: [-0.42884464263916017, -0.32163181304931643]
iter: 35 - x: [-0.42884464263916017, -0.32163181304931643]
Answer: [-0.42884464263916017, -0.32163181304931643]
f(x) = 3.1091263510349982
```

Рис. 5: Иллюстрация работы алгоритма

4.2 Решение задачи при граничной оптимальной точке

Решение задачи многомерной минимизации при ограничениях такого вида, что оптимальная точка x^* находится на границе рассматриваемой области, с точностью $\varepsilon = 10^{-3}$:

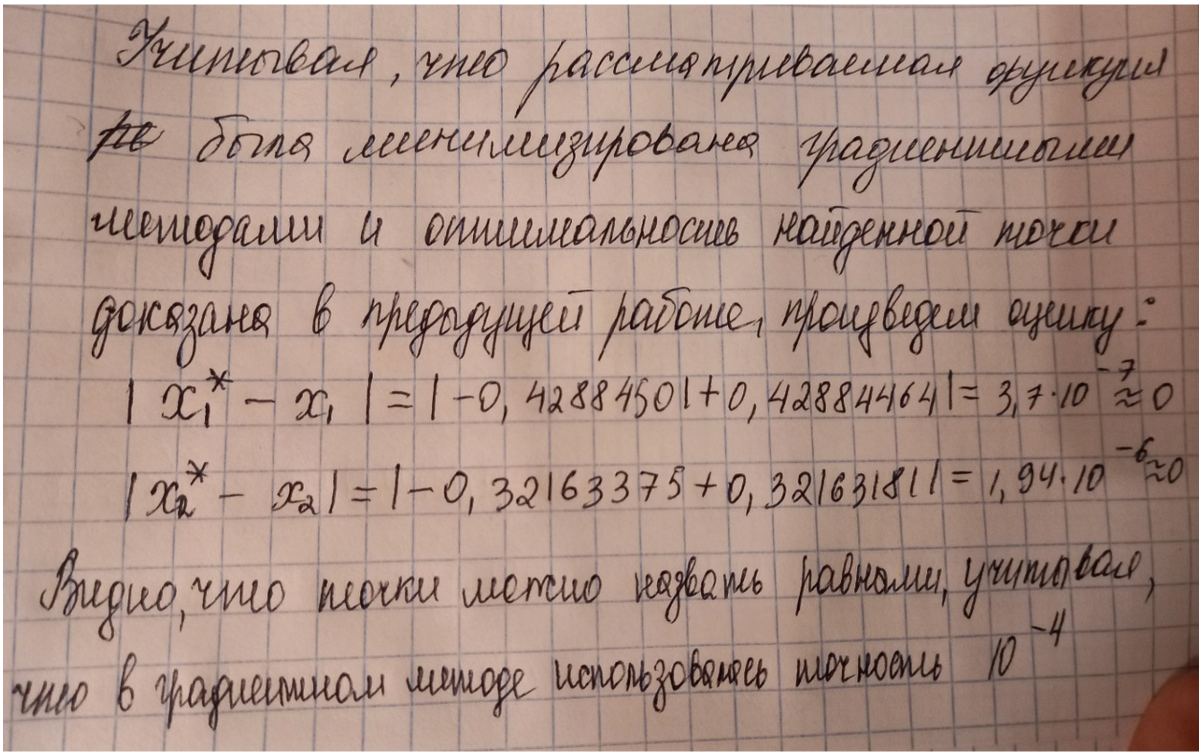
x^*	(-0.42875498, -0.32138672)
ϕ_0^*	3.10912645
Число итераций	20



```
iter: 0 - x: [-0.325, -0.275]
iter: 1 - x: [-0.3875, -0.3375]
iter: 2 - x: [-0.41875, -0.30625]
iter: 3 - x: [-0.41875, -0.30625]
iter: 4 - x: [-0.4265625, -0.3140625]
iter: 5 - x: [-0.4265625, -0.3140625]
iter: 6 - x: [-0.4265625, -0.3140625]
iter: 7 - x: [-0.428515625, -0.316015625]
iter: 8 - x: [-0.428515625, -0.316015625]
iter: 9 - x: [-0.4275390625, -0.3169921875]
iter: 10 - x: [-0.4294921875, -0.3189453125]
iter: 11 - x: [-0.428515625, -0.319921875]
iter: 12 - x: [-0.428515625, -0.319921875]
iter: 13 - x: [-0.42900390625, -0.32041015625]
iter: 14 - x: [-0.428515625, -0.3208984375]
iter: 15 - x: [-0.428515625, -0.3208984375]
iter: 16 - x: [-0.428759765625, -0.321142578125]
iter: 17 - x: [-0.428759765625, -0.321142578125]
iter: 18 - x: [-0.4288818359375, -0.3212646484375]
iter: 19 - x: [-0.428759765625, -0.32138671875]
iter: 20 - x: [-0.428759765625, -0.32138671875]
Answer: [-0.428759765625, -0.32138671875]
f(x) = 3.109126446210616
```

Рис. 6: Иллюстрация работы алгоритма

5 Обоснование результатов



Учитывая, что рассматриваемая функция была минимизирована градиентным методом и оптимальность найденной точки доказана в предыдущей работе, произведем оценку:

$$|x_1^* - x_1| = |-0,42884501 + 0,42884464| = 3,7\cdot 10^{-7} \approx 0$$

$$|x_2^* - x_2| = |-0,32163375 + 0,32163181| = 1,94\cdot 10^{-6} \approx 0$$

Видно, что точки можно назвать равными, учитывая, что в градиентном методе использовалась точность 10^{-4}

6 Выводы

Метод имеет ряд преимуществ. Во-первых, он гарантирует сходимость к оптимальной точке при выполнении основных условий применимости метода. К тому же решение вспомогательной задачи линейного программирования можт производиться любым наиболее подходящим под условия методом. Также учитываются при построении направления следующего шага только ограничения, близкие к активным, а на остальных ограничениях можно рассматривать задачу безусловной минимизации. Стоит сказать, что выбор величины шага позволяет оценить убывание значения функции в следующей точке по отношению к предыдущему шагу.

Недостатками данного метода могут являться достаточно жесткие условия применимости метода, налагаемые на функцию и область ее исследования, и сложность аналитического определения параметра дробления исходя из обстоятельств подбора шага и необходимости отодвигаться поближе к границе.

Зачастую в задачах на выбор начального приближения из рассматриваемого можества сложно подобрать такую точку, чтобы учесть все наборы ограничений. Для этого может использоваться метод всевозможных направлений Зойтендейка, позволяющий найти допустимую точку и выбрать ее в качестве начального приближения.

7 Приложения

URL: Выполненная лабораторная работа на GitHub
https://github.com/ThinkingFrog/OptimizationMethods/tree/main/MultiDimWithRestriction