

## **СОДЕРЖАНИЕ**

|   |    |
|---|----|
| 1 Методы локальной оптимизации . . . . .                                | 3  |
| 1.1 Минимум функции одного переменного . . . . .                        | 3  |
| 1.2 Минимизация функций многих переменных . . . . .                     | 6  |
| 1.2.1 Спуск по координатам . . . . .                                    | 7  |
| 1.2.2 Метод градиентного спуска . . . . .                               | 9  |
| 1.2.3 Метод тяжелого шара . . . . .                                     | 12 |
| 1.3 Численная оптимизация местоположения склада торговой сети . . . . . | 12 |
| 1.3.1 Метод по координатного спуска . . . . .                           | 13 |
| 1.3.2 Метод градиентного спуска . . . . .                               | 25 |

# 1 Методы локальной оптимизации

Оптимизация – это задача нахождения экстремума (минимума или максимума) целевой функции в некоторой области конечномерного векторного пространства, ограниченной набором линейных и/или нелинейных равенств и/или неравенств.

Во многих практических важных случаях для целевой функции многих переменных  $f(\mathbf{x})$  задача оптимизации может быть сформулирована в виде:

$$f(\mathbf{x}) \rightarrow \min,$$

где  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  – вектор неизвестных (управляющих параметров);  $\min$  – минимальное значение функции в ограниченной или неограниченной области изменения неизвестных.

Для нахождения абсолютного минимума целевой функции  $f(\mathbf{x})$  существует только один способ: найти все локальные минимумы этой функции, сравнить их и выбрать из них тот, в котором функция принимает наименьшее значение.

## 1.1 Минимум функции одного переменного

Для функции одной переменной  $f(x)$ , задача нахождения минимума эквивалента задачи нахождения корней уравнения:

$$\frac{df(x)}{dx} = 0 \quad (1)$$

Эта одномерная задача нередко возникает в практических приложениях. Кроме того, большинство методов решения многомерных задач сводится к поиску одномерного минимума.

Предположим, что  $f(x)$  задана и кусочно-непрерывна на отрезке  $x \in [a, b]$ , и имеет на этом отрезке (включая его концы) только один локальный минимум. Построим итерационный процесс, сходящийся к этому минимуму.

Вычислим значение функции на концах отрезка  $x = a$  и  $x = b$ , а также в двух внутренних точках  $x_1 < x_2$ . Так как функция  $f(x)$  имеет минимум на отрезке  $x \in [a, b]$ , то справедливо утверждение:

$$f(a) \geq f(x_1), \quad f(x_2) \leq f(b)$$

Сравним все четыре значения функции между собой  $f(a)$ ,  $f(x_1)$ ,  $f(x_2)$  и  $f(b)$  и выберем среди них наименьшее.



Рисунок 1 – График кусочно-непрерывной функции  $y = f(x)$ , имеющей минимум на отрезке  $x \in [a, b]$

Из рисунка 1 видно, что наименьшее значение функция достигает в точке  $x = x_1$ :

$$f(x_1) < f(a) < f(x_2) < f(b)$$

Очевидно, что минимум функции  $f(x)$  расположен в одном из прилегающих к точке  $x = x_1$  отрезков, то есть минимум находится либо в пределах отрезка  $[a, x_1]$ , либо в  $[x_1, x_2]$  (рисунок 1, выделенная область).

Поэтому на первом шаге итерационного процесса отбрасывается отрезок  $[x_2, b]$ , и для поиска минимума функции  $f(x)$  рассматривается отрезок  $[a, x_2]$ , при этом область поиска минимума функции сужается:

$$|a - x_2| < |a - b|, \quad \text{так как} \quad x_2 < b.$$

Полагая  $b = x_2$ , на новом отрезке  $[a, b]$  вновь необходимо выбрать две внутренние точки, вычислить в них и на концах отрезка значения функции  $f(x)$ , и сделать следующий шаг итерационного процесса.

Критерием остановки итерационного процесса является условие выполнения неравенства, которое гарантирует малость размера области поиска ми-

нимума по сравнению с заранее заданной погрешность метода:

$$(b - a) \leq \epsilon,$$

где  $\epsilon$  – погрешность метода.

*Симметричный метод* поиска минимума функции одной переменной  $f(x)$  основан на выборе внутренних точек  $x_1$  и  $x_2$  отрезка  $[a, b]$ , которые равноудалены от концов этого отрезка. Например, если точки  $x_1$  и  $x_2$  делят отрезок  $[a, b]$  на три равные части (рисунок 2), то координаты этих точек могут быть определены из соотношений:

$$x_1 = a + \frac{b - a}{3} = \frac{2a + b}{3}, \quad x_2 = b - \frac{b - a}{3} = \frac{a + 2b}{3}.$$



Рисунок 2 – Схематическое изображение точек деления отрезка  $[a, b]$

Оценка длины отрезка после первого итерационного шага составит:

$$\ell_1 = (b - a) - \frac{b - a}{3} = \frac{2}{3} \cdot (b - a),$$

после второго шага:

$$\ell_2 = \ell_1 - \frac{\ell_1}{3} = \frac{2}{3} \cdot \ell_1 = \left(\frac{2}{3}\right)^2 \cdot (b - a),$$

а после  $k$ -ого итерационного шага:

$$\ell_k = \left(\frac{2}{3}\right)^k \cdot (b - a).$$

Таким образом, чтобы погрешность вычисления  $\ell_k$  была менее  $\epsilon$ , для числа итераций  $k$  справедлива оценка:

$$\left(\frac{2}{3}\right)^k \cdot (b - a) \leq \epsilon \quad \rightarrow \quad k = \left\lceil \frac{\ln(b - a) - \ln(\epsilon)}{\ln(3) - \ln(2)} \right\rceil$$

Симметричный метод поиска минимума функции является аналогом метода дихотомии для нахождения корня уравнения  $f(x) = 0$ . Метод применим к

недифференцируемым функциям и всегда сходится. Следует отметить, что если на отрезке  $[a, b]$  функция  $f(x)$  имеет несколько локальных минимумов, то итерационный процесс сойдется к одному из этих минимумов, но не обязательно к наименьшему.

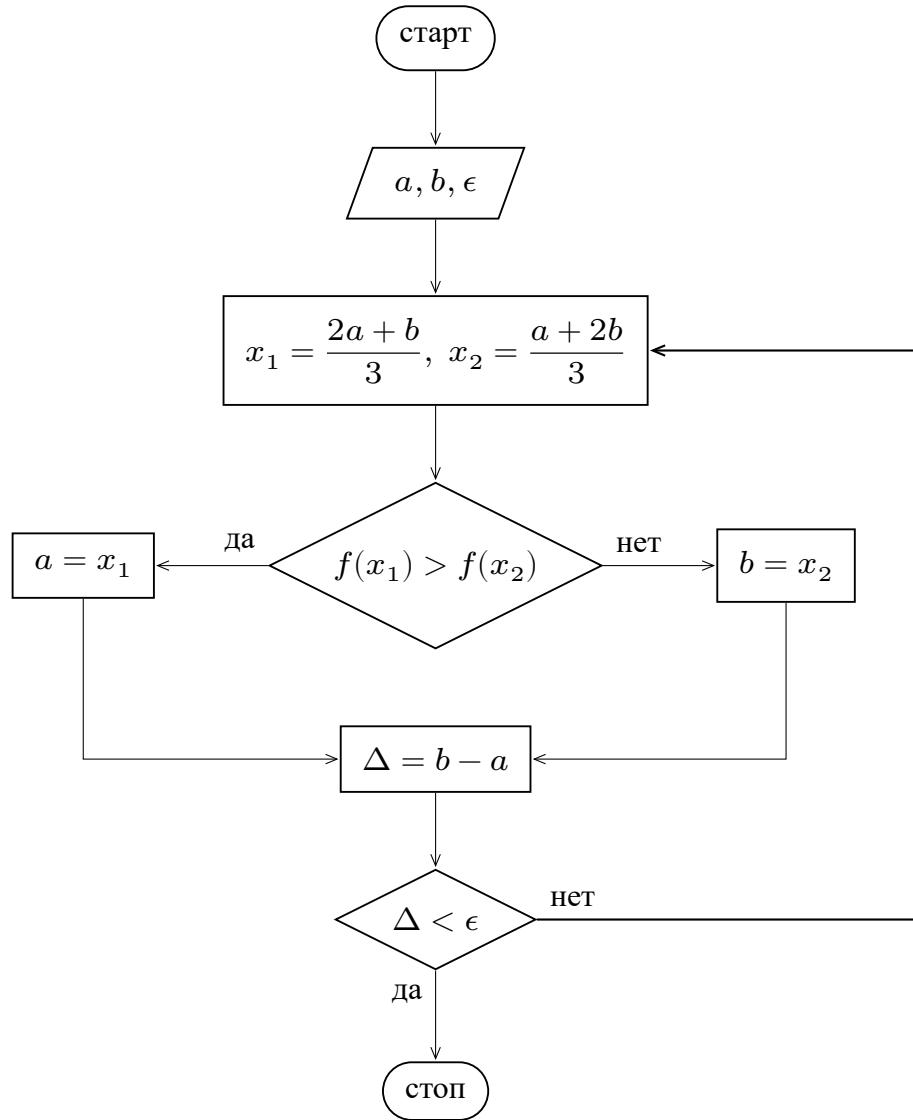


Рисунок 3 – Блок-схема алгоритма нахождения минимума функции  $f(x)$  одного переменного

## 1.2 Минимизация функций многих переменных

Задача безусловной минимизации (оптимизации) состоит в нахождении минимума или максимума функции  $f(x)$  в отсутствие каких-либо ограничений на область изменения переменных задачи  $x \in (-\infty, +\infty)$ .

Большинство практических задач оптимизации содержит ограничения,

которые обусловлены технико-экономическим смыслом решаемой задачи, однако многие алгоритмы решения задач с ограничениями предполагают сведение ее к последовательности задач безусловной оптимизации.

### 1.2.1 Спуск по координатам

Идея метода по координатного спуска заключается в том, что задача поиска минимума функции  $f(x_1, x_2, \dots, x_n) \rightarrow \min$  разделяется на серию задач *одномерной минимизации* этой функции вдоль направления одной из координатных осей:

$$\begin{aligned} f(\textcolor{red}{x}_1, x_2 = \text{const}, \dots, x_n = \text{const}) &\rightarrow \min \\ f(x_1 = \text{const}, \textcolor{red}{x}_2, \dots, x_n = \text{const}) &\rightarrow \min \\ \dots & \\ f(x_1 = \text{const}, x_2 = \text{const}, \dots, \textcolor{red}{x}_n) &\rightarrow \min \end{aligned}$$

- 1) Выбирают нулевое приближение  $\boldsymbol{x}_0 = (x_{10}, x_{20}, \dots, x_{n0})$ .
- 2) Фиксируют (считают постоянными) значения всех координат кроме  $x_1$ . Тогда функция  $f(\boldsymbol{x})$  будет зависеть только от одной переменной  $x_1$ :

$$\phi_1(x_1) = f(x_1, x_{20} = \text{const}, \dots, x_{n0} = \text{const})$$

- 3) Используя метод одномерной минимизации, находится минимум функции одной переменной  $\phi_1(x_1) \rightarrow \min$ , который можно обозначить через  $m_1$ .
- 4) Сделан переход из начальной точки  $\boldsymbol{x}_0$  в точку “частного“ минимума по направлению, параллельному оси  $x_1$ :

$$(x_{10}, x_{20}, \dots, x_{n0}) \rightarrow (m_1, x_{20}, \dots, x_{n0}),$$

и значение функции уменьшается:

$$f(x_{10}, x_{20}, \dots, x_{n0}) > f(m_1, x_{20}, \dots, x_{n0})$$

- 5) Затем из новой точки  $(m_1, x_{20}, \dots, x_{n0})$  осуществляется спуск по направлению, параллельному оси  $x_2$ , то есть находится минимум функции:

$$\phi_2(x_2) = f(m_1 = \text{const}, x_2, \dots, x_n = \text{const}),$$

который обозначим  $m_2$ .

- 6) Таким образом, сделан переход во вторую точку “частного“ минимума по направлению, параллельному оси  $x_2$ :

$$(m_1, x_{20}, \dots, x_{n0}) \rightarrow (m_1, m_2, \dots, x_{n0}),$$

и значение функции уменьшается:

$$f(m_1, x_{20}, \dots, x_{n0}) > f(m_1, m_2, \dots, x_{n0})$$

- 7) Процесс спуска по координатам повторяется для всех переменных задачи  $x_3, x_4, \dots, x_n$ , а приход в точку  $\mathbf{x}_m = (m_1, m_2, \dots, m_n)$  завершает цикл спусков.
- 8) Конечную точку цикла спусков можно принять за нулевое приближение  $\mathbf{x}_0 = \mathbf{x}_m$  и повторить цикл спусков по координатам  $x_1, x_2, \dots, x_n$  до тех пор, пока не выполнено условие останова итерационного процесса.

Практически можно задать некоторое число  $\epsilon > 0$ , связанное с выбранной точностью вычислений, и проводить итерации до тех пор, пока на  $k$ -ой итерации не будут выполнены одно или несколько неравенств вида:

$$\|\mathbf{x}_k - \mathbf{x}_{k-1}\| < \epsilon_1, \quad \|f(\mathbf{x}_k) - f(\mathbf{x}_{k-1})\| < \epsilon_2 \quad (2)$$

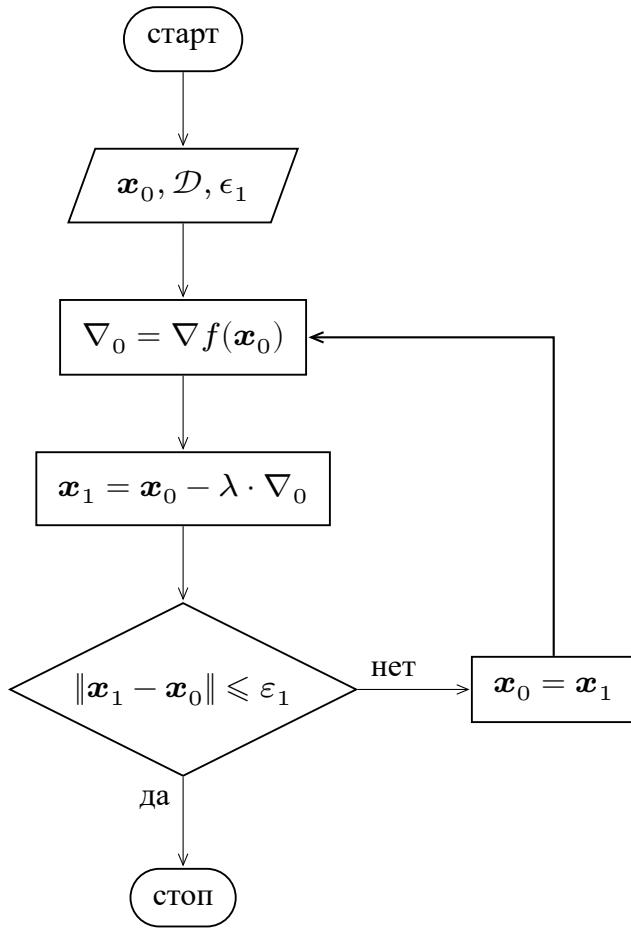


Рисунок 4 – Блок-схема алгоритма нахождения минимума функции  $f(\mathbf{x})$  многих переменных методом градиентного спуска

### 1.2.2 Метод градиентного спуска

Градиентный спуск – метод нахождения локального экстремума (минимума или максимума) функции многих переменных  $f(x_1, x_2, \dots, x_n)$  с помощью движения вдоль градиента этой функции. Это наиболее простой в реализации из всех методов локальной оптимизации, но имеет относительно малую (линейную) скорость сходимости.

Градиент  $\nabla$  это вектор, указывающий направление наибольшего возрастания некоторой функции  $f$ , значение которой меняется от одной точки пространства к другой (скалярного поля), а по величине (модулю) равный скорости роста этой величины в этом направлении. Компонентами вектора градиента являются частные производные  $f$  по всем её аргументам:

$$\nabla f = \left( \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right) \quad (3)$$

Для случая трёхмерного пространства градиентом скалярной функции  $f(x, y, z)$  называется векторная функция:

$$\text{grad } f = \nabla f,$$

где  $\nabla$  – векторный дифференциальный оператор набла, компоненты которого являются частными производными по координатам:

$$\nabla = \left( \frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right)$$

Следует отметить, что оператор набла не принадлежит тому же пространству, что и обычные векторы, а говоря точнее, скалярное и векторное произведение для него определено с некоторыми различиями. Оператор  $\nabla$  действует на те скалярные поля, что стоят от него справа, и не действует на стоящие от него слева. Поэтому скалярное и векторное произведение с участием  $\nabla$  *не коммутативны* и не антисимметричны, как это свойственно для таких произведений обычных векторов.

Минимизация целевой функции  $f(\mathbf{x})$  сводится к итерационному процессу последовательного выбора нового вектора неизвестных  $\mathbf{x}_{k+1}$ , такого чтобы значение функции в новой точке было меньше чем в предыдущих:

$$f(\mathbf{x}_0) > f(\mathbf{x}_1) > \dots > f(\mathbf{x}_k) > f(\mathbf{x}_{k+1}) > \dots$$

Предполагая, что новый вектор неизвестных мало отличается от предыдущего ( $\mathbf{x}_{k+1} - \mathbf{x}_k \approx 0$ ), можно воспользоваться линейным приближением для разложения в ряд Тейлора целевой функции:

$$f(\mathbf{x}_{k+1}) = f(\mathbf{x}_k) + (\nabla f(\mathbf{x}_k), \mathbf{x}_{k+1} - \mathbf{x}_k), \quad (4)$$

где  $k$  – номер итерационного шага процесса;  $\mathbf{x}_k$  – значение неизвестных на  $k$ -ой итерации.

Если в качестве нового вектора неизвестных выбрать:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \lambda \cdot \nabla f(\mathbf{x}_k), \quad (5)$$

то из (4) получим:

$$f(\mathbf{x}_{k+1}) = f(\mathbf{x}_k) - \lambda \cdot \|\nabla f(\mathbf{x}_k)\|^2 \rightarrow f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k) \quad (6)$$

где  $\lambda > 0$  – малое положительное число (параметр метода), имеющий смысл скорости градиентного спуска;  $\|\nabla f(\mathbf{x}_k)\| \geq 0$  – норма вектора градиента (неотрицательное число):

$$\|\nabla f\| = \sqrt{(\nabla f, \nabla f)}$$

Таким образом, выбор нового вектора неизвестных  $\mathbf{x}_{k+1}$  в соответствии с выражением (5), гарантирует монотонное убывание целевой функции  $f(\mathbf{x})$  в каждой итерации. Поэтому основная идея метода градиентного спуска заключается в том, чтобы последовательно идти в направлении наибольшего уменьшения целевой функции, которое задаётся антиградиентом  $-\nabla f(\mathbf{x})$ .

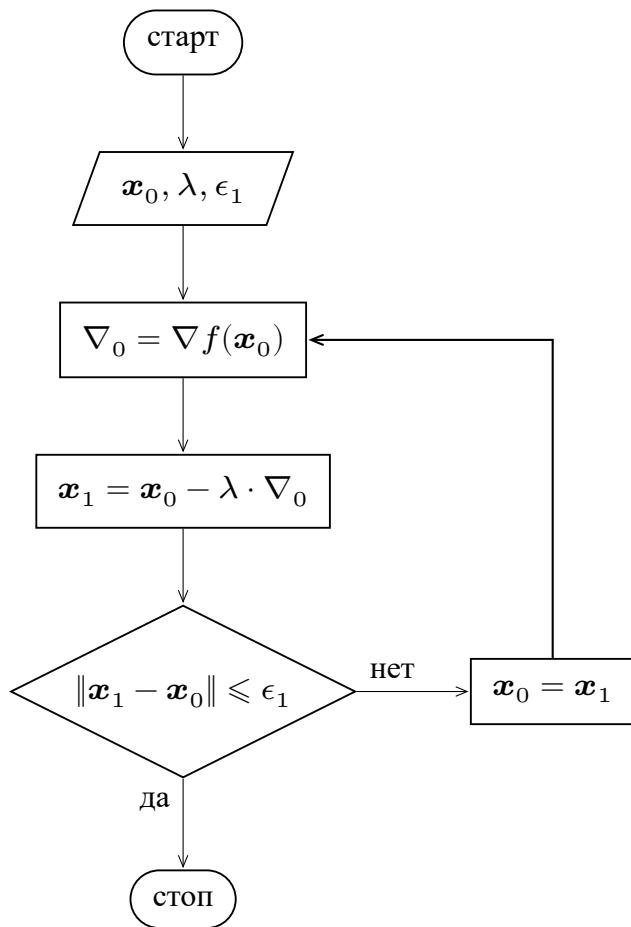


Рисунок 5 – Блок-схема алгоритма нахождения минимума функции  $f(\mathbf{x})$  многих переменных методом градиентного спуска

### 1.2.3 Метод тяжелого шара

Поиск минимума функции многих переменных  $f(\mathbf{x})$  методом “тяжелого шара“ основан на аналогии движения материальной частицы массой  $m$  в консервативном силовом поле  $\mathbf{F}(\mathbf{x})$  в вязкой среде.

В соответствии с принципом минимальной энергии тело смещается в положение, которое минимизирует общую потенциальную энергию системы  $f(\mathbf{x}) \rightarrow \min$ . Поэтому если предположить, что функция  $f(\mathbf{x})$  является потенциальной энергией частицы в консервативном силовом поле  $\mathbf{F}(\mathbf{x}) = -\nabla f(\mathbf{x})$ , и частица перемещается в пространстве  $\mathbf{x}$  минимизируя свою энергию, то уравнение движения этой частицы можно записать в виде:

$$\begin{cases} \frac{d\mathbf{x}}{dt} = \mathbf{v} \\ m \frac{d\mathbf{v}}{dt} = \mathbf{F} - \alpha \cdot \mathbf{v} \end{cases} \quad (7)$$

где  $\mathbf{x}$  – положение частицы в выбранной системе координат;  $\mathbf{v}$  и  $\alpha$  – скорость и коэффициент вязкого трения частицы в среде, соответственно.

Этот метод используется в методе стохастического градиентного спуска и в качестве расширения алгоритмов обратного распространения ошибок для обучения искусственных нейронных сетей.

Поиск минимума данным методом начинается с задания начальных условий, которые, как правило, формулируются в виде:

$$\begin{cases} \mathbf{x}(0) = \mathbf{x}_0 \\ \mathbf{v}(0) = \mathbf{v}_0 \end{cases}, \quad (8)$$

где  $\mathbf{x}_0$  – начальное приближение для поиска минимума функции;  $\mathbf{v}_0$  – “начальная скорость“ в пространстве неизвестных.

Масса частицы  $m$  и коэффициент вязкого трения  $\alpha$  являются эвристическими параметрами метода и выбираются произвольным образом, отражающим специфику решаемой задачи.

## 1.3 Численная оптимизация местоположения склада торговой сети

Рассмотрим задачу нахождения координат расположения склада готовой продукции (распределительного центра), таких чтобы суммарное расстояние от склада до потребителей продукции (магазинов торговой сети) было минималь-

ным.

### 1.3.1 Метод по координатного спуска

Будем полагать, что магазины торговой сети находятся в городах, которые имеют определенные местоположения заданные их географическими координатами.

Географические координаты определяют местоположение точки на поверхности Земли, и строятся по принципу сферических координат. Центр системы географических координат помещается в центр Земли, а положение точки в пространстве определяется расстоянием  $r$  от центра Земли, широтой  $\phi$  и долготой  $\lambda$  (рисунок 6):

- 1) широта  $\phi$  – угол между местным направлением зенита и плоскостью экватора, отсчитываемый в обе стороны от экватора.
- 2) долгота  $\lambda$  – двугранный угол между плоскостью меридиана, проходящего через данную точку, и плоскостью начального нулевого меридиана.

Радиус-вектор  $\mathbf{r}$  точки  $A$ , находящейся на поверхности Земли, имеет в декартовой системе координат компоненты:

$$\mathbf{r} = (r_x, r_y, r_z),$$

где  $r_{x,y,z}$  – компоненты вектора  $\mathbf{r}$ , то есть проекции этого вектора на оси декартовой системы координат:

$$\begin{cases} r_x &= \mathcal{R} \cdot \cos(\phi) \cdot \cos(\lambda) \\ r_y &= \mathcal{R} \cdot \cos(\phi) \cdot \sin(\lambda) \\ r_z &= \mathcal{R} \cdot \sin(\phi) \end{cases},$$

$\mathcal{R}$  – радиус планеты Земля.

Таким образом, для определения местоположения *на поверхности Земли*, достаточно задать широту  $\phi$  и долготу  $\lambda$  географического объекта.

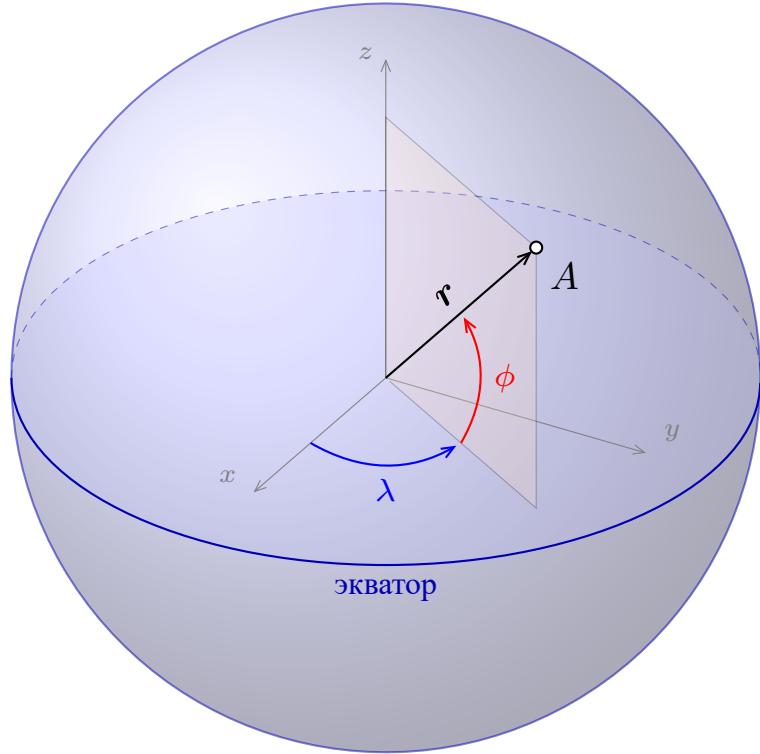


Рисунок 6 – Географическая система координат на поверхности Земли

Кратчайшее расстояние между двумя точками  $A$  и  $B$  находящимися на сферической поверхности Земли равно длине дуги большого круга. Если известны географические координатами точек  $A(\phi_a, \lambda_a)$  и  $B(\phi_b, \lambda_b)$ , то расстояние на поверхности Земли между этими точками составляет (рисунок 7):

$$d_{AB} = R \cdot \alpha, \quad (9)$$

где  $\alpha$  – центральный угол между радиус-векторами  $a$  и  $b$  точек  $A$  и  $B$ , лежащих на поверхности Земли.

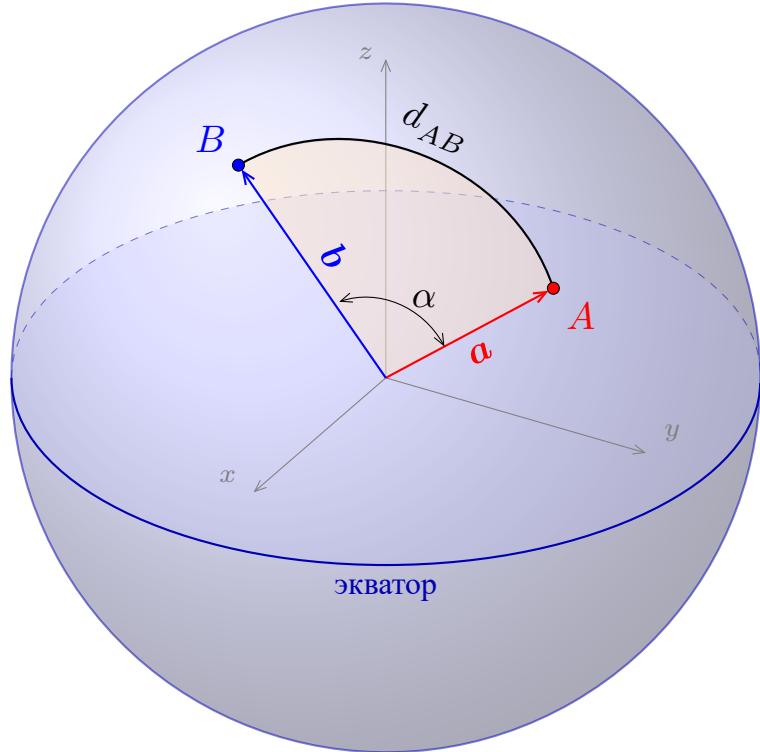


Рисунок 7 – Расстояние  $d_{AB}$  (длина дуги) на поверхности сферы между точками  $A$  и  $B$

Косинус угла  $\alpha$  между векторами  $a$  и  $b$  можно определить, воспользовавшись геометрическим свойством скалярного произведения этих векторов:

$$(a, b) = \|a\| \cdot \|b\| \cdot \cos(\alpha) \quad \rightarrow \quad \cos(\alpha) = \frac{(a, b)}{\|a\| \cdot \|b\|}, \quad (10)$$

где  $\|a\| = \|b\| = R$  – длины радиус-векторов, с учетом постоянства радиуса Земли.

С алгебраической точки зрения, *скалярное произведение векторов* – это операция над двумя векторами  $a$  и  $b$ , результатом которой является число (скаляр), значение которого не зависит от выбранной системы координат, то есть скалярное произведение является *инвариантом*:

$$(a, b) = a_x \cdot b_x + a_y \cdot b_y + a_z \cdot b_z, \quad (11)$$

где  $a_{x,y,z}$  и  $b_{x,y,z}$  – компоненты векторов в декартовой системе координат:

$$\mathbf{a} = \begin{cases} a_x &= \mathcal{R} \cdot \cos(\phi_a) \cdot \cos(\lambda_a) \\ a_y &= \mathcal{R} \cdot \cos(\phi_a) \cdot \sin(\lambda_a) \\ a_z &= \mathcal{R} \cdot \sin(\phi_a) \end{cases} \quad (12)$$

$$\mathbf{b} = \begin{cases} b_x &= \mathcal{R} \cdot \cos(\phi_b) \cdot \cos(\lambda_b) \\ b_y &= \mathcal{R} \cdot \cos(\phi_b) \cdot \sin(\lambda_b) \\ b_z &= \mathcal{R} \cdot \sin(\phi_b) \end{cases} \quad (13)$$

Косинус угла (10) между векторами  $\mathbf{a}$  и  $\mathbf{b}$  с учетом (12) и (13):

$$\begin{aligned} \cos(\alpha) = & \cos(\phi_a) \cos(\lambda_a) \cdot \cos(\phi_b) \cos(\lambda_b) + \\ & \cos(\phi_a) \sin(\lambda_a) \cdot \cos(\phi_b) \sin(\lambda_b) + \\ & \sin(\phi_a) \cdot \sin(\phi_b) \end{aligned}$$

или

$$\cos(\alpha) = \cos(\phi_a) \cos(\phi_b) \cdot \left( \begin{array}{l} \cos(\lambda_a) \cdot \cos(\lambda_b) + \\ + \sin(\lambda_a) \cdot \sin(\lambda_b) \end{array} \right) + \sin(\phi_a) \sin(\phi_b)$$

Воспользовавшись формулой для разности углов тригонометрических функций (косинуса):

$$\cos(\lambda_a) \cdot \cos(\lambda_b) + \sin(\lambda_a) \cdot \sin(\lambda_b) = \cos(\lambda_a - \lambda_b)$$

окончательно получим:

$$\cos(\alpha) = \cos(\phi_a) \cos(\phi_b) \cdot \cos(\Delta\lambda) + \sin(\phi_a) \sin(\phi_b),$$

где  $\Delta\lambda = \lambda_a - \lambda_b$  – разность долготы географических координат точек  $A$  и  $B$ , соответственно.

Таким образом, кратчайшее расстояние на поверхности Земли между двумя точками (9) может быть найдено из соотношения:

$$d_{AB} = \mathcal{R} \cdot \arccos (\cos(\phi_a) \cos(\phi_b) \cdot \cos(\Delta\lambda) + \sin(\phi_a) \sin(\phi_b)), \quad (14)$$

1) С помощью картографических веб сервисов (например, [Яндекс.Карты](#), [кар-](#)

ты Google, и т.п.). определим географические координаты всех городов, в которых находятся магазины торговой сети Полученные данные представим в таблице 1.

Таблица 1 – Географические координаты магазинов торговой сети

| Город   | Географические координаты |                   |
|---------|---------------------------|-------------------|
|         | широта $\phi$             | долгота $\lambda$ |
| Москва  | 55.66352                  | 37.62964          |
| Саратов | 51.53440                  | 46.03121          |
| Воронеж | 56.67071                  | 39.16190          |

- 2) Построим целевую функцию  $f(\phi, \lambda)$  – суммарное расстояние вдоль поверхности Земли от склада до магазинов в городах Москва, Саратов и Воронеж:

$$f(\phi, \lambda) = d_1(\phi, \lambda) + d_2(\phi, \lambda) + d_3(\phi, \lambda),$$

где  $\phi$  и  $\lambda$  – географическая широта и долгота местоположения склада;  $d_1$ ,  $d_2$  и  $d_3$  – расстояние вдоль поверхности Земли (14) между складом и городами Москва, Саратов и Воронеж, соответственно:

$$\begin{aligned}d_1(\phi, \lambda) &= \mathcal{R} \cdot \arccos (\cos(\phi) \cos(\phi_1) \cdot \cos(\lambda - \lambda_1) + \sin(\phi) \sin(\phi_1)) \\d_2(\phi, \lambda) &= \mathcal{R} \cdot \arccos (\cos(\phi) \cos(\phi_2) \cdot \cos(\lambda - \lambda_2) + \sin(\phi) \sin(\phi_2)) \\d_3(\phi, \lambda) &= \mathcal{R} \cdot \arccos (\cos(\phi) \cos(\phi_3) \cdot \cos(\lambda - \lambda_3) + \sin(\phi) \sin(\phi_3))\end{aligned}$$

где  $\phi_1 = 55.66352$  и  $\lambda_1 = 37.62964$  – широта и долгота города Москва;  
 $\phi_2 = 51.53440$  и  $\lambda_2 = 46.03121$  – широта и долгота города Саратов;  
 $\phi_3 = 56.67071$  и  $\lambda_3 = 39.16190$  – широта и долгота города Воронеж (таблица 1).

- 3) Для решения задачи оптимизации целевой функции *методом по координатного спуска* определим область поиска минимума функции  $r(\phi, \lambda)$ .  
Например, в качестве границ области можно выбрать экстремальные значения широты и долготы городов, в которых расположены магазины торговой

сети:

$$\begin{cases} \phi_{\inf} = \min(\phi_1, \phi_2, \phi_3) = \phi_2 = 51.53440 \\ \phi_{\sup} = \max(\phi_1, \phi_2, \phi_3) = \phi_3 = 56.67071 \\ \lambda_{\inf} = \min(\lambda_1, \lambda_2, \lambda_3) = \lambda_1 = 37.62964 \\ \lambda_{\sup} = \max(\lambda_1, \lambda_2, \lambda_3) = \lambda_2 = 46.03121 \end{cases}$$

Таким образом, минимум целевой функции будем искать в области  $\mathcal{D}$ :

$$\mathcal{D} = \begin{cases} \phi_{\inf} \leq \phi \leq \phi_{\sup} \\ \lambda_{\inf} \leq \lambda \leq \lambda_{\sup} \end{cases} = \begin{cases} 51.53440 \leq \phi \leq 56.67071 \\ 37.62964 \leq \lambda \leq 46.03121 \end{cases}$$

- 4) Критерием прекращения итерационного поиска минимума целевой функции считать условие выполнения одного из неравенств:

$$\epsilon_1 \geq \xi, \quad f(\phi_k, \lambda_k) \geq f(\phi_{k-1}, \lambda_{k-1}) \quad (15)$$

где  $\epsilon_1 = 0.1$  – заданная точность определения местоположения склада;  $\xi$  – изменение местоположения склада на текущем  $k$  шаге итерационного процесса:

$$\xi = \sqrt{(\phi_k - \phi_{k-1})^2 + (\lambda_k - \lambda_{k-1})^2}; \quad (16)$$

$\phi_k$  и  $\lambda_k$  – значение географических координат склада на  $k$ -ом шаге итерационного процесса.

- 5) В качестве начального местоположение склада  $(\phi_0, \lambda_0)$  выберем любую произвольную точку внутри области поиска минимума целевой функции, например:

$$\begin{cases} \phi_0 = 52.0 \\ \lambda_0 = 44.0 \end{cases}.$$

- 6) Считаем постоянными значения долготы склада  $\lambda = \lambda_0 = \text{const}$  и рассматриваем вспомогательную функцию только одной переменной  $\phi$ :

$$\rho_1(\phi) = f(\phi, \lambda_0) = f(\phi, 44.0)$$

Находим минимум вспомогательной функции одной переменной  $\rho_1(\phi)$ , ис-

пользуя метод одномерной минимизации:

$$\phi_{\min} = \arg \left( \min_{\phi_{\inf} \leq \phi \leq \phi_{\sup}} \rho_1(\phi) \right) = 54.66658$$

где  $\phi_{\min}$  – положение “частного“ минимума функции  $f(\phi, 44.0)$ .

Считаем постоянными значения широты склада  $\phi = \phi_{\min} = \text{const}$  и рассматриваем вторую вспомогательную функцию только одной переменной  $\lambda$ :

$$\rho_2(\lambda) = f(\phi_{\min}, \lambda) = f(54.66658, \lambda)$$

Находим минимум функции одной переменной  $\rho_2(\lambda)$ , используя метод одномерной минимизации:

$$\lambda_{\min} = \arg \left( \min_{\lambda_{\inf} \leq \lambda \leq \lambda_{\sup}} \rho_2(\lambda) \right) = 39.49103$$

где  $\lambda_{\min}$  – положение “частного“ минимума функции  $f(54.66658, \lambda)$ .

- 7) Определяем изменение местоположения склада (16) в результате проведенных итерационных процессов (рисунок 8):

$$\begin{aligned} \xi &= \sqrt{(\phi_{\min} - \phi_0)^2 + (\lambda_{\min} - \lambda_0)^2} = \\ &= \sqrt{(54.66658 - 52.0)^2 + (39.49103 - 44.0)^2} = \\ &= 5.23846 \end{aligned}$$

Находим значение целевой функции в начальной  $f(\phi_0, \lambda_0)$  и конечной точки  $f(\phi_{\min}, \lambda_{\min})$  точках местоположения склада:

$$\begin{aligned} f(\phi_0, \lambda_0) &= f(52.0, 44.0) &= 1338.696 \\ f(\phi_{\min}, \lambda_{\min}) &= f(54.66658, 39.49103) &= 944.004 \end{aligned}$$

Сравниваем величину текущего “шага“  $\xi$  и заданную точность расчетов  $\epsilon_1$ , а также значения целевой функции  $f(\phi, \lambda)$  в начальной и конечной точки итерационного процесса:

$$\epsilon_1 = 0.1 < \xi = 5.23846$$

$$f(\phi_{\min}, \lambda_{\min}) = 944.004 < f(\phi_0, \lambda_0) = 1338.696$$

Критерий остановки итерационного процесса (15) не выполняется, следова-

тельно, итерационный процесс продолжаем!

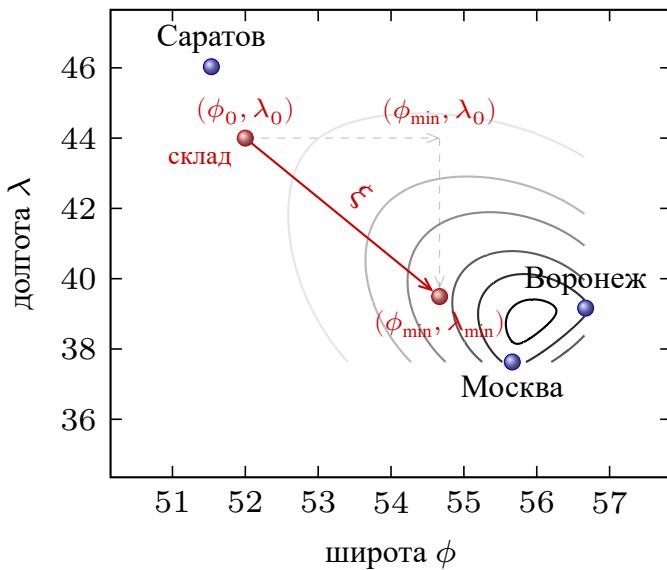


Рисунок 8 – “Рельеф“ целевой функции  $r(\phi, \lambda)$  и “траектория“ поиска местоположения склада  $(\phi, \lambda)$  торговой сети (маркерами обозначены города)

- 8) За новое местоположение склада торговой сети принимается положение “частного“ минимума целевой функции на предыдущем шаге итерационного процесса:

$$\begin{cases} \phi_0 = \phi_{\min} = 54.66658 \\ \lambda_0 = \lambda_{\min} = 39.49103 \end{cases}$$

- 9) Считаем постоянными значения долготы склада  $\lambda = \lambda_0 = \text{const}$  и рассматриваем вспомогательную функцию только одной переменной  $\phi$ :

$$\rho_1(\phi) = f(\phi, \lambda_0) = f(\phi, 39.49103)$$

Находим минимум вспомогательной функции одной переменной  $\rho_1(\phi)$ , используя метод одномерной минимизации:

$$\phi_{\min} = \arg \left( \min_{\phi_{\inf} \leq \phi \leq \phi_{\sup}} \rho_1(\phi) \right) = 55.94734$$

где  $\phi_{\min}$  – положение “частного“ минимума функции  $f(\phi, 39.49103)$ .

Считаем постоянными значения широты склада  $\phi = \phi_{\min} = \text{const}$  и рассмат-

риваем вторую вспомогательную функцию только одной переменной  $\lambda$ :

$$\rho_2(\lambda) = f(\phi_{\min}, \lambda) = f(55.94734, \lambda)$$

Находим минимум функции одной переменной  $\rho_2(\lambda)$ , используя метод одномерной минимизации:

$$\lambda_{\min} = \arg \left( \min_{\lambda_{\inf} \leq \lambda \leq \lambda_{\sup}} \rho_2(\lambda) \right) = 38.88818$$

где  $\lambda_{\min}$  – положение “частного“ минимума функции  $f(55.94734, \lambda)$ .

- 10) Определяем изменение местоположения склада (16) в результате проведенных итерационных процессов (рисунок 9):

$$\begin{aligned} \xi &= \sqrt{(\phi_{\min} - \phi_0)^2 + (\lambda_{\min} - \lambda_0)^2} = \\ &= \sqrt{(55.94734 - 54.66658)^2 + (38.88818 - 39.49103)^2} = \\ &= 1.41555 \end{aligned}$$

Находим значение целевой функции в начальной  $f(\phi_0, \lambda_0)$  и конечной точке  $f(\phi_{\min}, \lambda_{\min})$  точках местоположения склада:

$$\begin{aligned} f(\phi_0, \lambda_0) &= f(54.66658, 39.49103) = 944.004 \\ f(\phi_{\min}, \lambda_{\min}) &= f(55.94734, 38.88818) = 845.572 \end{aligned}$$

Сравниваем величину текущего “шага“  $\xi$  и заданную точность расчетов  $\epsilon_1$ , а также значения целевой функции  $f(\phi, \lambda)$  в начальной и конечной точки итерационного процесса:

$$\begin{aligned} \epsilon_1 &= 0.1 < \xi = 1.41555 \\ f(\phi_{\min}, \lambda_{\min}) &= 845.572 < f(\phi_0, \lambda_0) = 944.004 \end{aligned}$$

Критерий остановки итерационного процесса (15) не выполняется, следовательно, *итерационный процесс продолжаем!*

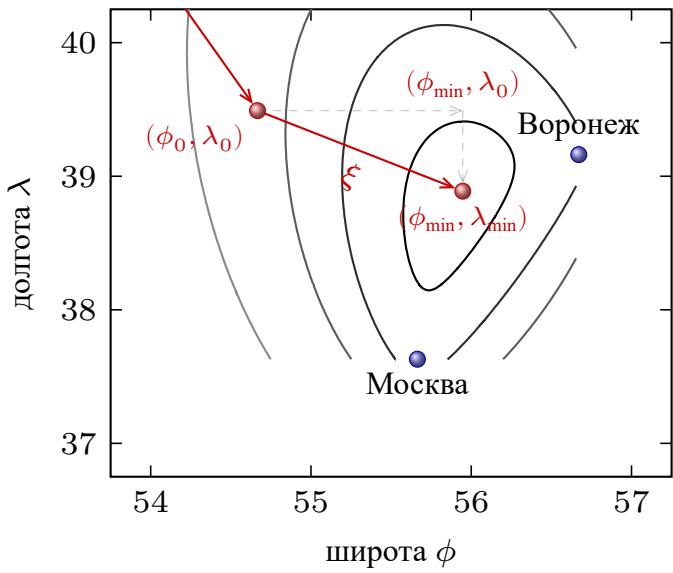


Рисунок 9 – “Рельеф“ целевой функции  $r(\phi, \lambda)$  и “траектория“ поиска местоположения склада  $(\phi, \lambda)$  торговой сети (маркерами обозначены города)

- 11) За новое местоположение склада торговой сети принимается положение “частного“ минимума целевой функции на предыдущем шаге итерационного процесса:

$$\begin{cases} \phi_0 = \phi_{\min} = 55.94734 \\ \lambda_0 = \lambda_{\min} = 38.88818 \end{cases}$$

- 12) Считаем постоянными значения долготы склада  $\lambda = \lambda_0 = \text{const}$  и рассматриваем вспомогательную функцию только одной переменной  $\phi$ :

$$\rho_1(\phi) = f(\phi, \lambda_0) = f(\phi, 38.88818)$$

Находим минимум вспомогательной функции одной переменной  $\rho_1(\phi)$ , используя метод одномерной минимизации:

$$\phi_{\min} = \arg \left( \min_{\phi_{\inf} \leq \phi \leq \phi_{\sup}} \rho_1(\phi) \right) = 55.88973$$

где  $\phi_{\min}$  – положение “частного“ минимума функции  $f(\phi, 38.88818)$ .

Считаем постоянными значения широты склада  $\phi = \phi_{\min} = \text{const}$  и рассматриваем вторую вспомогательную функцию только одной переменной  $\lambda$ :

$$\rho_2(\lambda) = f(\phi_{\min}, \lambda) = f(55.88973, \lambda)$$

Находим минимум функции одной переменной  $\rho_2(\lambda)$ , используя метод одномерной минимизации:

$$\lambda_{\min} = \arg \left( \min_{\lambda_{\inf} \leq \lambda \leq \lambda_{\sup}} \rho_2(\lambda) \right) = 38.83941$$

где  $\lambda_{\min}$  – положение “частного“ минимума функции  $f(55.88973, \lambda)$ .

- 13) Определяем изменение местоположения склада (16) в результате проведенных итерационных процессов (рисунок 10):

$$\begin{aligned}\xi &= \sqrt{(\phi_{\min} - \phi_0)^2 + (\lambda_{\min} - \lambda_0)^2} = \\ &= \sqrt{(55.88973 - 55.94734)^2 + (38.83941 - 38.88818)^2} = \\ &= 0.07548\end{aligned}$$

Находим значение целевой функции в начальной  $f(\phi_0, \lambda_0)$  и конечной точки  $f(\phi_{\min}, \lambda_{\min})$  точках местоположения склада:

$$\begin{aligned}f(\phi_0, \lambda_0) &= f(55.94734, 38.88818) = 845.572 \\ f(\phi_{\min}, \lambda_{\min}) &= f(55.88973, 38.83941) = 845.314\end{aligned}$$

Сравниваем величину текущего “шага“  $\xi$  и заданную точность расчетов  $\epsilon_1$ , а также значения целевой функции  $f(\phi, \lambda)$  в начальной и конечной точки итерационного процесса:

$$\begin{aligned}\epsilon_1 &= 0.1 > \xi = 0.07548 \\ f(\phi_{\min}, \lambda_{\min}) &= 845.314 < f(\phi_0, \lambda_0) = 845.572\end{aligned}$$

Критерий остановки итерационного процесса (15) выполняется, следовательно, *итерационный процесс прекращаем!*

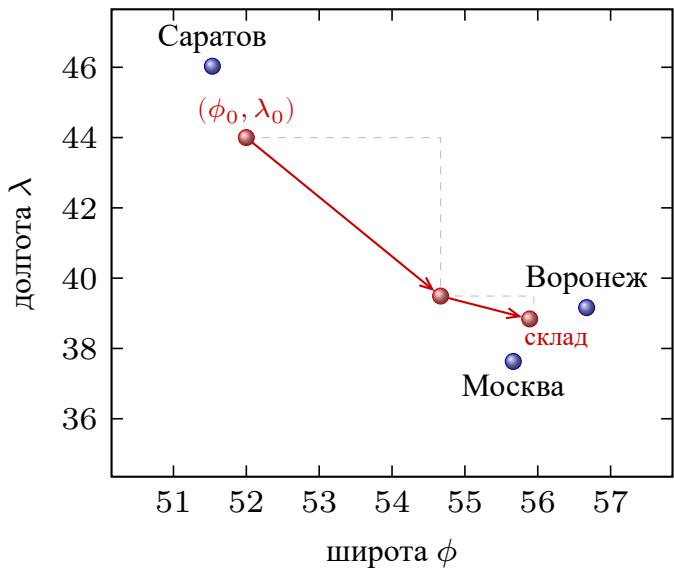


Рисунок 10 – “Рельеф“ целевой функции  $r(\phi, \lambda)$  и “траектория“ поиска местоположения склада  $(\phi, \lambda)$  торговой сети (маркерами обозначены города)

14) Полученные данные в процессе выполнения минимизации целевой функции методом по координатного спуска представлены в таблице 2.

Таблица 2 – “Траектория“ поиска местоположения склада торговой сети

| Итерация | Местоположение склада |                   | Шаг $\xi$ | Расстояние $f(\phi, \lambda)$ , км |
|----------|-----------------------|-------------------|-----------|------------------------------------|
|          | широта $\phi$         | долгота $\lambda$ |           |                                    |
| 0        | 52.00000              | 44.00000          | 0.00000   | 1338.696                           |
| 1        | 54.66658              | 39.49103          | 5.23846   | 944.004                            |
| 2        | 55.94734              | 38.88818          | 1.41555   | 845.572                            |
| 3        | 55.88973              | 38.83941          | 0.07548   | 845.314                            |

На рисунке 11 представлена зависимость суммарного расстояния между складом и всеми магазинами от номера итерации  $k$  поиска минимума целевой функции:

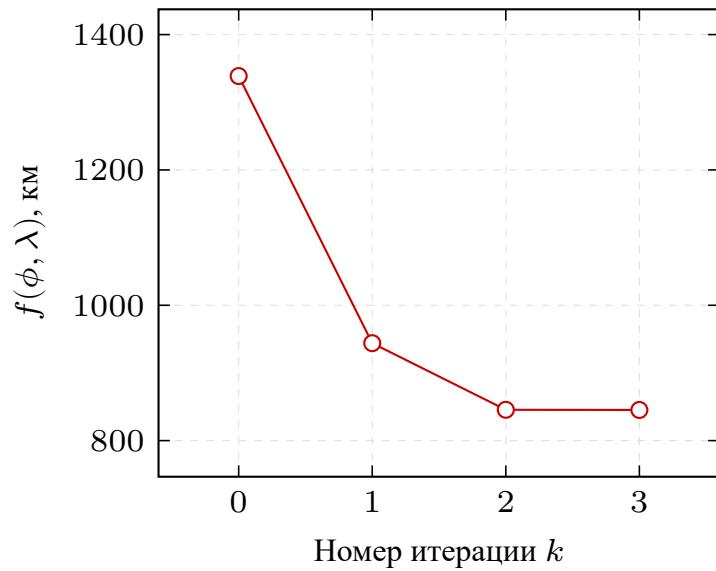


Рисунок 11 – Зависимость суммарного расстояния  $f(\phi, \lambda)$  между складом и магазинами торговой сети от числа итераций

На рисунке представлена зависимость изменение местоположения склада  $\xi$  от номера итерации  $k$  процесса поиска минимума целевой функции:

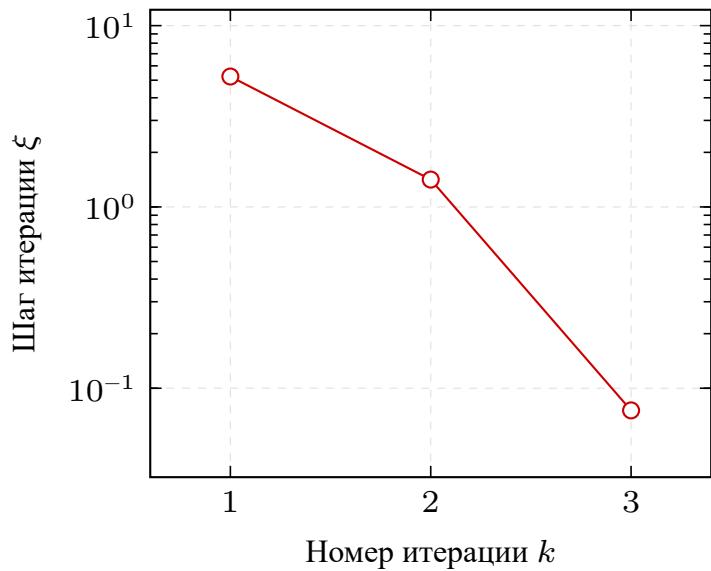
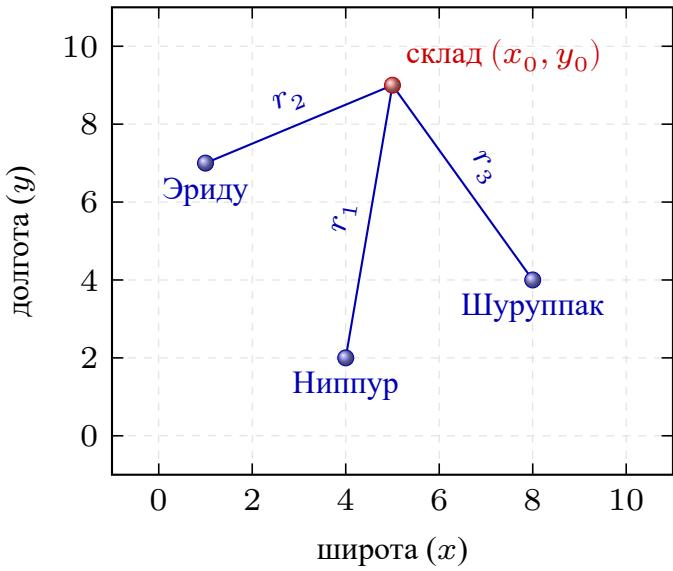


Рисунок 12 – Изменение местоположения склада  $\xi$  на  $k$ -ом шаге итерационного процесса

### 1.3.2 Метод градиентного спуска

Известны местоположение трех городов торговой сети в декартовой системе координат: Ниппур (4, 2), Эриду (1, 7) Шуруппак (8, 4).



1) Обозначим неизвестные:

$x$  и  $y$  – положения склада по горизонтальной и вертикальной оси координат, соответственно.

2) Целевая функция – суммарное расстояние от склада до всех магазинов:

$$f = r_1 + r_2 + r_3,$$

где  $r_1$ ,  $r_2$  и  $r_3$  – расстояние от *склада* до городов Ниппур, Эриду и Шуруппак, соответственно.

3) В выбранной декартовой системе координат и для нахождения расстояния от склада до каждого города воспользуемся теоремой [Пифагора Самосского](#):

$$\begin{aligned} r_1 &= \sqrt{(x_1 - x)^2 + (y_1 - y)^2} \\ r_2 &= \sqrt{(x_2 - x)^2 + (y_2 - y)^2} \\ r_3 &= \sqrt{(x_3 - x)^2 + (y_3 - y)^2}, \end{aligned}$$

где  $x_1$  и  $y_1$  – декартовы координаты города Ниппур;  $x_2$  и  $y_2$  – декартовы координаты города Эриду;  $x_3$  и  $y_3$  – декартовы координаты города Шуруппак. Таким образом, целевая функция – суммарное расстояние от склада до всех городов, с учетом данных задания о координатах городов (Ниппур, Эриду и

Шурупак), запишется в виде:

$$f(x, y) = \sqrt{(4-x)^2 + (2-y)^2} + \\ \sqrt{(1-x)^2 + (7-y)^2} + \\ \sqrt{(8-x)^2 + (4-y)^2}$$

4) Определим градиент целевой функции  $\nabla f(x, y)$ :

$$\nabla f(x, y) = \left( \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right)$$

Для этого найдем частные производные целевой функции от координат  $x$  и  $y$  положения склада:

$$\frac{\partial f}{\partial x} = -\frac{4-x}{\sqrt{(4-x)^2 + (2-y)^2}} \\ -\frac{1-x}{\sqrt{(1-x)^2 + (7-y)^2}} \\ -\frac{8-x}{\sqrt{(8-x)^2 + (4-y)^2}}$$

$$\frac{\partial f}{\partial y} = -\frac{2-y}{\sqrt{(4-x)^2 + (2-y)^2}} \\ -\frac{7-y}{\sqrt{(1-x)^2 + (7-y)^2}} \\ -\frac{4-y}{\sqrt{(8-x)^2 + (4-y)^2}}$$

- 5) Выбираем (в общем случае, произвольно) начальные координаты склада, например,  $x_0 = 5$  и  $y_0 = 9$ , скорость градиентного спуска  $\lambda = 2$  и точность расчёта  $\epsilon_1 = 0.25$  (единиц измерения).
- 6) Текущее суммарное расстояние от склада до всех городов:

$$R_0 = f(5, 9) = \sqrt{(4-5)^2 + (2-9)^2} \\ + \sqrt{(1-5)^2 + (7-9)^2} \\ + \sqrt{(8-5)^2 + (4-9)^2} = 17.37$$

Определим градиент целевой функции в начальной точке положения склада

$(x_0, y_0)$ :

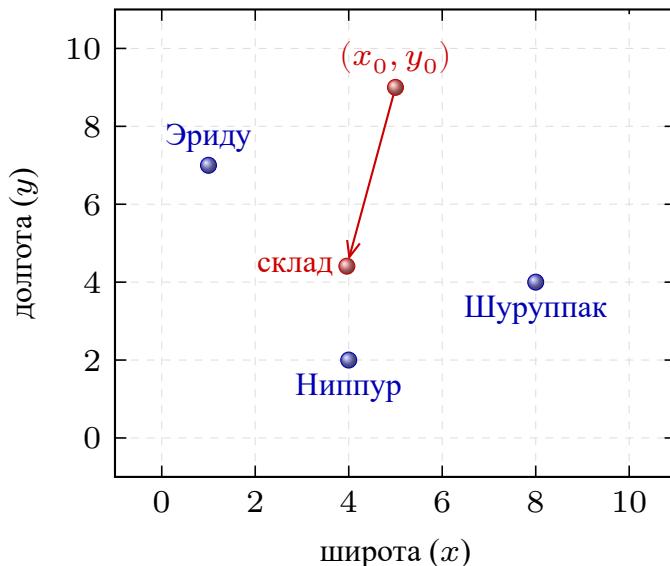
$$\begin{aligned}\left.\frac{\partial f}{\partial x}\right|_{\substack{x=5 \\ y=9}} &= -\frac{4-5}{\sqrt{(4-5)^2 + (2-9)^2}} \\ &\quad - \frac{1-5}{\sqrt{(1-5)^2 + (7-9)^2}} \\ &\quad - \frac{8-5}{\sqrt{(8-5)^2 + (4-9)^2}} = 0.52\end{aligned}$$

$$\begin{aligned}\left.\frac{\partial f}{\partial y}\right|_{\substack{x=5 \\ y=9}} &= -\frac{2-9}{\sqrt{(4-5)^2 + (2-9)^2}} \\ &\quad - \frac{7-9}{\sqrt{(1-5)^2 + (7-9)^2}} \\ &\quad - \frac{4-9}{\sqrt{(8-5)^2 + (4-9)^2}} = 2.29\end{aligned}$$

Зная градиент целевой функции в начальной точке  $\nabla f(x_0, y_0) = (0.52, 2.29)$ , определим новые координаты склада:

$$\begin{aligned}x_1 &= x_0 - \lambda \cdot \left.\frac{\partial f}{\partial x}\right|_{\substack{x=5 \\ y=9}} = 5 - 2 \cdot 0.52 = 3.96 \\ y_1 &= y_0 - \lambda \cdot \left.\frac{\partial f}{\partial y}\right|_{\substack{x=5 \\ y=9}} = 9 - 2 \cdot 2.29 = 4.41\end{aligned}$$

Новые координаты склада ( $x_1 = 3.96$ ;  $y_1 = 4.41$ ).



Рассчитаем величину “шага“ – расстояния между двумя последовательными положениями склада:

$$\xi = \sqrt{(5 - 3.96)^2 + (9 - 4.41)^2} = 4.71$$

Сравниваем величину текущего “шага“  $\xi$  и заданную точность расчетов  $\epsilon_1$ :

$$\xi = 4.71 > \epsilon_1 = 0.25$$

Величина текущего “шага“  $r$  больше заданной точности расчетов  $\epsilon_1$ , следовательно, *итерационный процесс продолжаем!*

7) Текущее суммарное расстояние от склада до всех городов:

$$\begin{aligned} R_1 = f(3.96, 4.41) = & \sqrt{(4 - 3.96)^2 + (2 - 4.41)^2} \\ & + \sqrt{(1 - 3.96)^2 + (7 - 4.41)^2} \\ & + \sqrt{(8 - 3.96)^2 + (4 - 4.41)^2} = 10.41 \end{aligned}$$

Суммарное расстояние уменьшилось:

$$R_1 = 10.41 < R_0 = 17.37.$$

Рассчитаем градиент целевой функции в новой точке положения склада  $x_1 = 3.96$  и  $y_1 = 4.41$ :

$$\begin{aligned} \frac{\partial f}{\partial x} \Big|_{\substack{x_1 = 3.96 \\ y_1 = 4.41}} = & -\frac{4 - 3.96}{\sqrt{(4 - 3.96)^2 + (2 - 4.41)^2}} \\ & -\frac{1 - 3.96}{\sqrt{(1 - 3.96)^2 + (7 - 4.41)^2}} \\ & -\frac{8 - 3.96}{\sqrt{(8 - 3.96)^2 + (4 - 4.41)^2}} = -0.26 \end{aligned}$$

$$\begin{aligned} \frac{\partial f}{\partial y} \Big|_{\substack{x_1 = 3.96 \\ y_1 = 4.41}} = & -\frac{2 - 4.41}{\sqrt{(4 - 3.96)^2 + (2 - 4.41)^2}} \\ & -\frac{7 - 4.41}{\sqrt{(1 - 3.96)^2 + (7 - 4.41)^2}} \\ & -\frac{4 - 4.41}{\sqrt{(8 - 3.96)^2 + (4 - 4.41)^2}} = 0.44 \end{aligned}$$

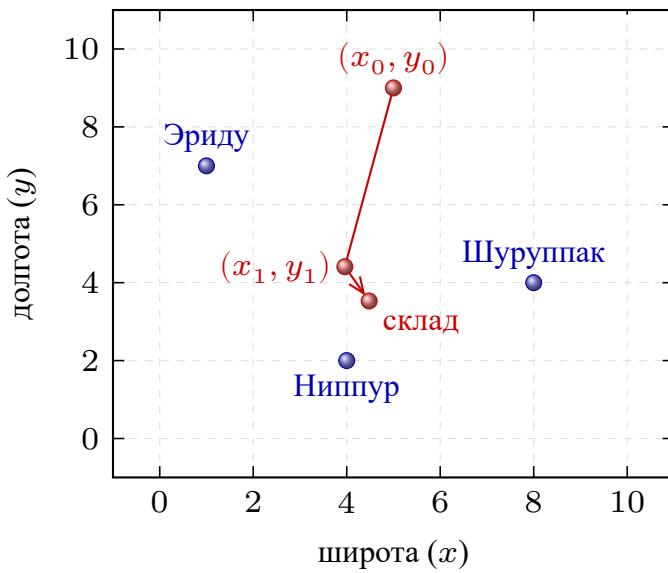
Зная градиент целевой функции в текущей точке  $\nabla f(x_1, y_1) = (-0.26, 0.44)$ ,

определим новые координаты склада:

$$x_2 = x_1 - \lambda \cdot \frac{\partial f}{\partial x} \Big|_{\substack{x=3.96 \\ y=4.41}} = 3.96 - 2 \cdot (-0.26) = 4.48$$

$$y_2 = y_1 - \lambda \cdot \frac{\partial f}{\partial y} \Big|_{\substack{x=3.96 \\ y=4.41}} = 4.41 - 2 \cdot 0.44 = 3.53$$

*Новые координаты склада ( $x_2 = 4.48, y_2 = 3.53$ ).*



Рассчитаем величину “шага“ – расстояния между двумя последовательными положениями склада:

$$\xi = \sqrt{(3.96 - 4.48)^2 + (4.41 - 3.53)^2} = 1.03$$

Сравниваем величину текущего “шага“  $\xi$  и заданную точность расчетов  $\epsilon_1$ :

$$\xi = 1.03 > \epsilon_1 = 0.25$$

Величина текущего “шага“  $\xi$  больше заданной точности расчетов  $\epsilon_1$ , следовательно, *итерационный процесс продолжаем!*

8) Текущее суммарное расстояние от склада до всех городов:

$$R_2 = f(4.48, 3.53) = \sqrt{(4 - 4.48)^2 + (2 - 3.53)^2} \\ + \sqrt{(1 - 4.48)^2 + (7 - 3.53)^2} \\ + \sqrt{(8 - 4.48)^2 + (4 - 3.53)^2} = 10.07$$

Рассчитаем градиент целевой функции в новой точке положения склада  $x_2 = 4.48$  и  $y_2 = 3.53$ :

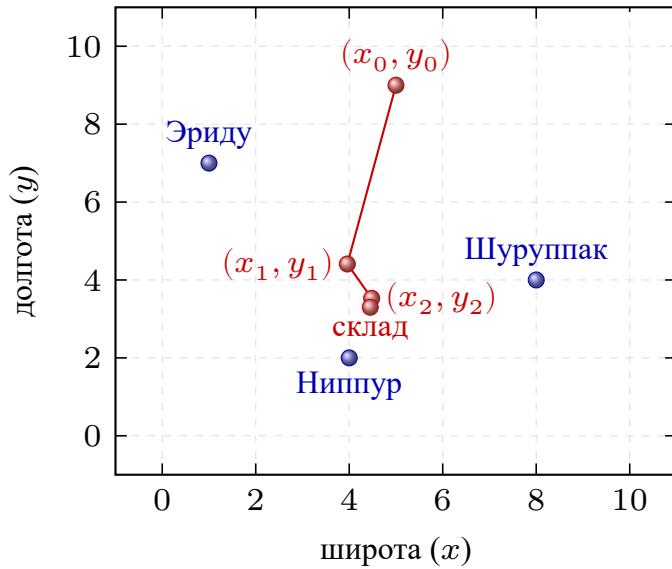
$$\frac{\partial f}{\partial x} \Big|_{\substack{x_2 = 4.48 \\ y_2 = 3.53}} = -\frac{4 - 4.48}{\sqrt{(4 - 4.48)^2 + (2 - 3.53)^2}} \\ -\frac{1 - 4.48}{\sqrt{(1 - 4.48)^2 + (7 - 3.53)^2}} \\ -\frac{8 - 4.48}{\sqrt{(8 - 4.48)^2 + (4 - 3.53)^2}} = 0.02$$

$$\frac{\partial f}{\partial y} \Big|_{\substack{x_2 = 4.48 \\ y_2 = 3.53}} = -\frac{2 - 3.53}{\sqrt{(4 - 4.48)^2 + (2 - 3.53)^2}} \\ -\frac{7 - 3.53}{\sqrt{(1 - 4.48)^2 + (7 - 3.53)^2}} \\ -\frac{4 - 3.53}{\sqrt{(8 - 4.48)^2 + (4 - 3.53)^2}} = 0.11$$

Зная градиент целевой функции в текущей точке  $\nabla f(x_2, y_2) = (0.02, 0.11)$ , определяют новое географическое положение склада:

$$x_3 = x_2 - \lambda \cdot \frac{\partial f}{\partial x} \Big|_{\substack{x=4.48 \\ y=3.53}} = 4.48 - 2 \cdot 0.02 = 4.45 \\ y_3 = y_2 - \lambda \cdot \frac{\partial f}{\partial y} \Big|_{\substack{x=4.48 \\ y=3.53}} = 3.53 - 2 \cdot 0.11 = 3.30$$

*Новые координаты склада ( $x_0 = 4.45$ ,  $y_0 = 3.30$ ).*



Рассчитаем величину “шага“  $\xi$  – расстояния между двумя последовательными положениями склада:

$$\xi = \sqrt{(4.48 - 4.45)^2 + (3.53 - 3.30)^2} = 0.23$$

Сравниваем величину текущего “шага“  $xi$  и заданную точность расчетов  $\epsilon_1$ :

$$\xi = 0.23 < \epsilon_1 = 0.25$$

Величина текущего “шага“  $\xi$  меньше заданной точности расчетов  $\epsilon_1$ , поэтому итерационный процесс поиска положения склада *прекращаем!*

- 9) Определим минимальное расстояние от склада до всех городов:

$$\begin{aligned} R_4 = f(4.45, 3.30) = & \sqrt{(4 - 4.45)^2 + (2 - 3.30)^2} \\ & + \sqrt{(1 - 4.45)^2 + (7 - 3.30)^2} \\ & + \sqrt{(8 - 4.45)^2 + (4 - 3.30)^2} = 10.05 \end{aligned}$$

Таким образом, с заданной точностью определены оптимальные координаты склада ( $x_{\min} = 4.45$ ,  $y_{\min} = 3.30$ ), при которых общее расстояние от всех городов до склада будет минимальным и составит  $R_{\min} = 10.05$ :