Розробка методу для задачі пошуку точки розміщення логістичного хабу

Умова

Задано **N** прямих, які задають дорогу.

Знайти розміщення точки (хабу), від якої відстань до прямих була б якомога меншою. Під відстанню до прямих будемо розуміти відстань до найбільш віддаленої прямої. Прямі задані двома точками.

Вхідні дані

Ввести **N** – кількість точок (**N <= 10^5**).

В наступних **N** рядках ввести по 4 значення. Це дві точки, які описують пряму.

Вихідні дані

Вивести шукану відповідь з точністю шість знаків після коми.

Математична постановка

Математичну постановку у цій постановці виконаємо таким чином: Нехай дано множину прямих $L = \{L_i \mid i = \overline{1,n}\}$, кожна з яких задається двома точками площини $L_i = \{(x_{1i}, y_{1i}), (x_{2i}, y_{2i})\}$, $i = \overline{1,n}$, при чому $\forall i \in \{1, 2, ..., n\}$ $(x_{1i}, y_{1i}) \neq (x_{2i}, y_{2i})$. Необхідно знайти таку точку площини $A^*(x^*, y^*)$, $(x, y) \in R^2$, відстань від якої до найдальшої прямої була б мінімальною.

Введемо в розгляд функцію відстані $d_i(x, y) = d(L_i, A)$, яка рівна евклідовій відстані від точки $A(x, y)$ до прямої L_i, $i = \overline{1,n}$.

Тоді задача розміщення логістичного хабу буде полягати в знаходженні такої точки $A^*(x^*, y^*)$, для якої б виконувалась умова:

$$A^* \in Arg \min_{A(x,y) \in R^2} \max_{i=\overline{1,n}} d(L_i, A).$$

Структура, що описує лінію

```
struct line {
    double a, b, c;
    line (double x1, double y1, double x2, double y2) {
        double aa = (y2 - y1);
        double bb= (x1 - x2);
        double cc = x2 * (y1 - y2) + y2 * (x2 - x1);
        double norm = sqrt(aa * aa + bb * bb);
        this->a = aa / norm;
        this->b = bb / norm;
        this->c = cc / norm;
    }
    double get(double x, double y) {
        return abs(a * x + b * y + c);
    }
};
```

Відстань від (x_0, y_0) до *i*-ої прямої $a_i \cdot x + b_i \cdot y + c_i = 0$

$$dist(x_0, y_0, a_i, b_i, c_i) = \frac{|a_i \cdot x_0 + b_i \cdot y_0 + c_i|}{\sqrt{a_i^2 + b_i^2}}.$$

Поділивши на $\sqrt{a_i^2 + b_i^2}$ значення коефіцієнтів прямої $a_i' = \frac{a_i}{\sqrt{a_i^2+b_i^2}}$, $b_i' = \frac{b_i}{\sqrt{a_i^2+b_i^2}}$, $c_i' = \frac{c_i}{\sqrt{a_i^2+b_i^2}}$, отримаємо, що відстань

$$dist(x_0, y_0, a_i, b_i, c_i) = |a_i' x_0 + \ b_i' y_0 + c_i'|.$$

Цільова функція буде мати вигляд

$$f(x, y) = \max_{i=\overline{1,n}} dist(x, y, a_i, b_i, c_i) \to \min_{\substack{x \epsilon R \\ y \epsilon R}}.$$

Уведемо дані та сформуємо всю інформацію про прямі у векторі

```
vector<line> lines:

  int n; cin >> n;
    for (int i = 0; i < n; i++) {
        int x1, y1, x2, y2;
        cin >> x1 >> y1 >> x2 >> y2;
        line s(x1, y1, x2, y2);
        lines.push_back(s);
    }
```

Відстань від точки до прямих будемо знаходити за допомогою функції:

```
double  dist(double x, double y) {
    double ans = 0.0;
    for (auto i : lines) {
        ans  = max(ans, i.get(x, y));
    }
    return ans;
}
```

Метод покоординатного спуску

Алгоритм

1. Визначення початкового наближення $A_0(x_0, y_0)$.

2. Підставляємо x_0 у функцію f і розв’язуємо задачу мінімізації функції однієї змінної:

$$f(x_0, y) = \max_{i=\overline{1,n}} \frac{|a_i x_0 + b_i y + c_i|}{\sqrt{a_i^2 + b_i^2}} \to \min_{y \in R}. \quad (**)$$

Нехай $y_1 \in Arg \min_{y \in R} f(x_0, y)$ і $A_1(x_0, y_1)$

3. Підставляємо y_1 у функцію f. Розв’язуємо задачу

$$f(x, y_1) = \max_{i=\overline{1,n}} \frac{|a_i x + b_i y_1 + c_i|}{\sqrt{a_i^2 + b_i^2}} \to \min_{y \in R}. \quad (***)$$

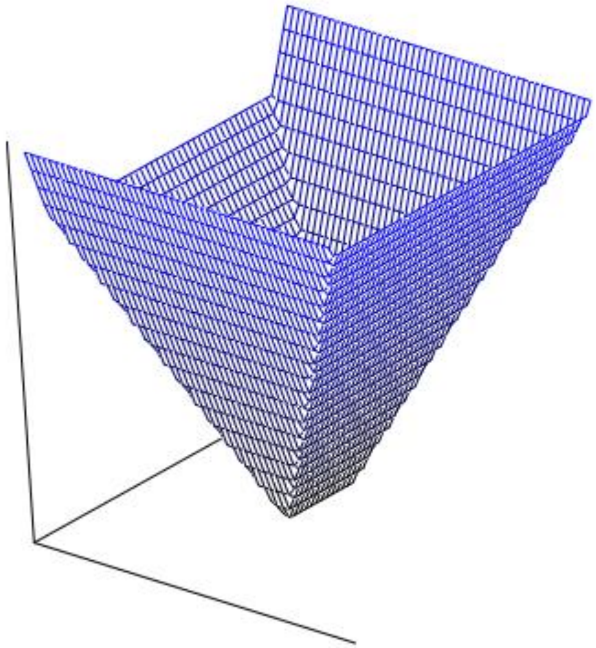
$x_1 \in Arg \min_{x \in R} f(x, y_1)$ і $A_2(x_1, y_1)$.

Процес ітераційно продовжується до того часу, поки $f(A_k) > f(A_{k+1})$.

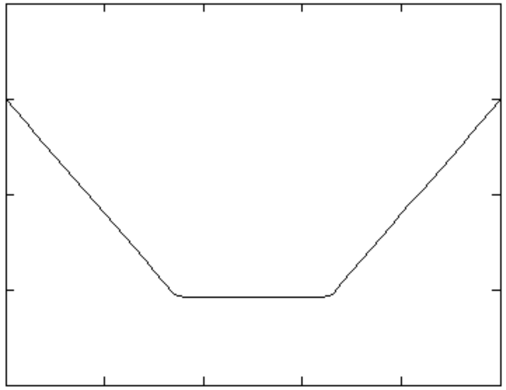
Критеріями зупинки алгоритму можуть бути:

- $\|A_k - A_{k+1}\| < \varepsilon$ – близькість точок, які генеруються на послідовних кроках;
- $\|f(A_k) - A(_{k+1})\| < \varepsilon$ – близькість значень цільової функції, отриманих на послідовних кроках;
- перевищення встановленого часу пошуку оптимального значення тощо.

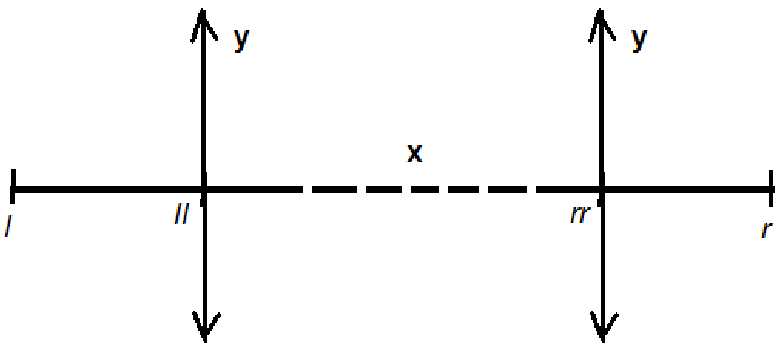
Метод золотого перерізу та тернарний пошук



Типовий вигляд цільової функції

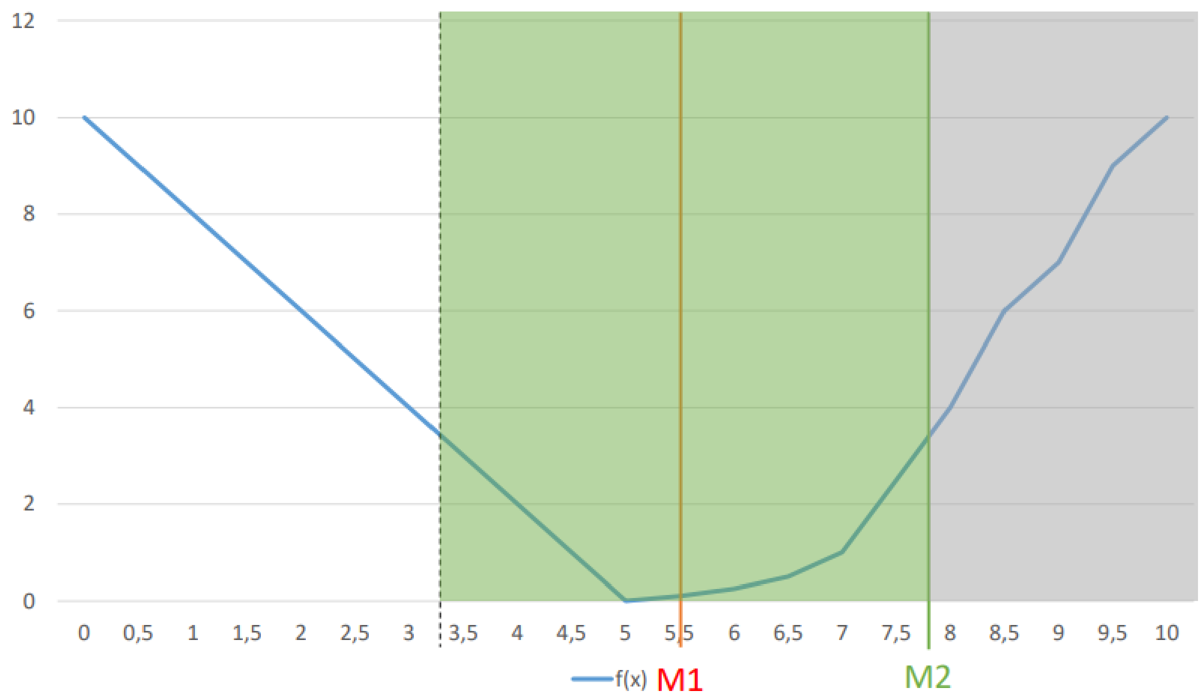


Типовий вигляд цільової функції з фіксованим одним параметром



Для визначення шуканого значення щодо змінної *x* кожен раз запускається метод золотого перерізу за змінною *y*, і досягнуте значення є індикатором вибору значення звуження проміжку за змінною *x*

Принцип роботи



f(M1) < f(M2)

- область, в якій немає мінімуму

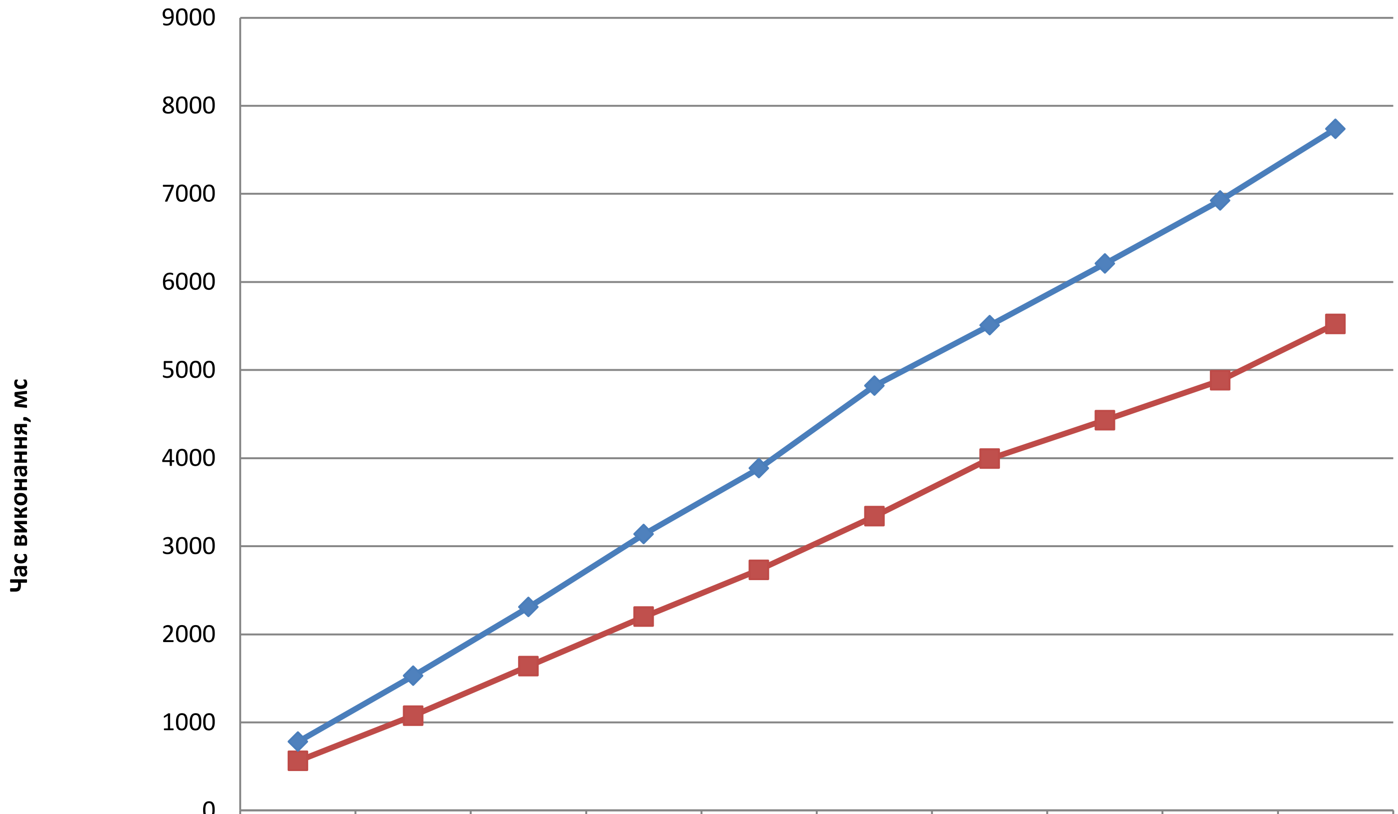
- область, в якій знаходиться мінімум

Реалізація

```
double Golden_Section_y (double x) {
    double l = -1e9, r = 1e9;
    while (r - l > eps) {
        double ll = l + 0.38 * (r - l);
        double rr = l + 0.62 * (r - l);
        if (dist(x, ll) < dist(x, rr))
            r = rr;
        else
            l = ll;
    }
    return l;
}
```

```
double Golden_Section_x () {
    double l = -1e9, r = 1e9;
    while (r - l > eps) {
        double ll = l + 0.38 * (r - l);
        double rr = l + 0.62 * (r - l);
        if (dist(ll,Golden_Section_y(ll)) <
                                    dist(rr,Golden_Section_y(rr)))
            r = rr;
        else
            l = ll;
    }
    return l;
}
```

Графіки залежності часу виконання тернарного пошуку та методу золотого перерізу від розмірності задачі



	10000	20000	30000	40000	50000	60000	70000	80000	90000	100000
Метод золотого перерізу	561	1076	1638	2199	2730	3338	3993	4430	4882	5522
Тернарний пошук	779	1528	2308	3135	3884	4820	5506	6208	6925	7737

Аналіз

- Для двовимірного випадку часова складність запропонованого підходу рівна $O(n \cdot log^2 C)$, а для методу покрокового спуску часова складність – $O(n^2 \cdot log C)$, де C – параметр, який залежить від діапазону пошуку та точності.
- При заміні коефіцієнтів у методі золотого перерізу 0.38 на 1/3, а 0.62 на 2/3 отримаємо метод тернарного пошуку.
- Метод тернарного пошуку потребує приблизно на 40% більше часових затрат, ніж метод золотого перерізу.

Тривимірний випадок

Умова

Розглянемо тривимірний випадок, коли прямі задають в просторі і потрібно знайти розміщення точки, від якої відстань до прямих була б якомога меншою.

Нехай, як і у двовимірному випадку, задано множину прямих $L_i = \{(x_{1i}, y_{1i}, z_{1i}), (x_{2i}, y_{2i}, z_{2i})\}$, $i = \overline{1,n}$, при чому $\forall i \in \{1, 2, ..., n\}$ $(x_{1i}, y_{1i}, z_{1i}) \neq (x_{2i}, y_{2i}, z_{2i})$. Тоді пряму будемо задавати системою двох рівнянь виду:

$$L_i : \begin{cases} a_{1i}x + b_{1i}y + c_{1i}z + d_{1i} = 0 \\ a_{2i}x + b_{2i}y + c_{2i}z + d_{2i} = 0 \end{cases},$$

де параметри $a_{1i}, a_{2i}, b_{i1}, b_{2i}, c_{1i}, c_{2i}, d_{1i}, d_{2i}$ обчислюють з рівностей:

$$\frac{x - x_{1i}}{x_{2i} - x_{1i}} = \frac{y - y_{1i}}{y_{2i} - y_{1i}} = \frac{z - z_{1i}}{z_{2i} - z_{1i}}$$

Цільова функція при цьому буде мати вигляд

$$f(x, y, z) = \max_{i = \overline{1,n}} \frac{\left\| \begin{matrix} i & j & k \\ x - x_{1i} & y - y_{1i} & z - z_{1i} \\ x_{2i} - x_{1i} & y_{2i} - y_{1i} & z_{2i} - z_{1i} \end{matrix} \right\|}{\sqrt{(x_{2i} - x_{1i})^2 + (y_{2i} - y_{1i})^2 + (z_{2i} - z_{1i})^2}} \to \min_{(x,y,z) \in R^3}$$

Схема розв’язання

```
double Golden_Section_z (double x, double y) {
    double l = -1e9, r = 1e9;
    while (r - l > eps) {
        double ll = l + 0.38 * (r - l);
        double rr = l + 0.62 * (r - l);
        if (dist(x, y, ll) < dist(x, y, rr))
            r = rr;
        else
            l = ll;
    }
    return l;
}

double Golden_Section_y (double x) {
double l = -1e9, r = 1e9;
    while (r - l > eps) {
        double ll = l + 0.38 * (r - l);
        double rr = l + 0.62 * (r - l);
        if (dist(x,ll,Golden_Section_z(x,ll)) <
                 dist(x,rr,Golden_Section_y(x,rr)))
            r = rr;
        else
            l = ll;
    }
    return l;
}

double Golden_Section_x () {
double l = -1e9, r = 1e9;
    while (r - l > eps) {
        double ll = l + 0.38 * (r - l);
        double rr = l + 0.62 * (r - l);
          double y1 = Golden_Section_y(ll);
          double z1 = Golden_Section_z(ll,y1);
          double y2 = Golden_Section_y(rr);
          double z2 = Golden_Section_z(rr,y2);
        if (dist(ll,y1,z1) < dist(rr,y2,z2))
            r = rr;
        else
            l = ll;
    }
    return l;
}
```

Лабораторна робота №5

1. Згенерувати від 10000 до 100000 з кроком 10000 прямих випадковим чином.

2. Використовуючи схему розв’язання з попереднього слайду побудувати графіки залежності часу виконання тернарного пошуку та методу золотого перерізу від розмірності задачі (аналогічно слайду 13). Описати порівняння (аналогічно слайду 14).

3*. Порівняти розглядувані методи з методом покоординатного спуску.

Дякую за увагу!