Международная научная конференция «Суперкомпьютерные дни в России»
(Москва, 23-24 сентября 2024 г.)

Исследование масштабируемости параллельной реализации алгоритма AlFaMove для линейного программирования на кластерной вычислительной системе

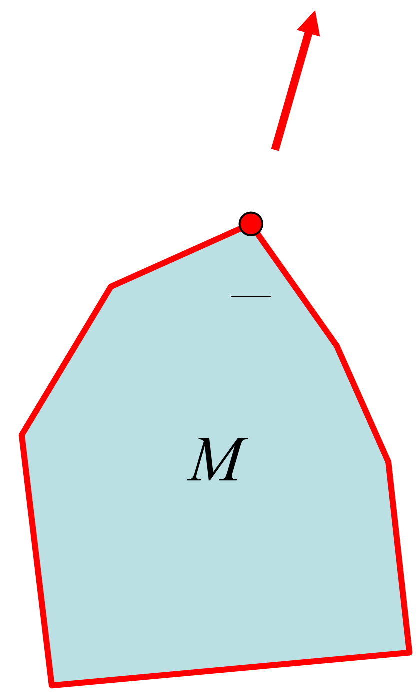
Н.А. Ольховский,
д.ф.-м.н. Л.Б. Соколинский

Южно-Уральский государственный университет
(национальный исследовательский университет)

Исследование выполнено при финансовой поддержке РНФ (проект №23-21-00356).

Задача линейного программирования

$$\_ = \arg\ \max\left\{\left\langle \quad , \quad \right\rangle \vee \qquad \le \quad \right\}$$



Допустимый многогранник

- – матрица
- – вектор размерности *m*
- – вектор размерности *n*
- – целевая функция

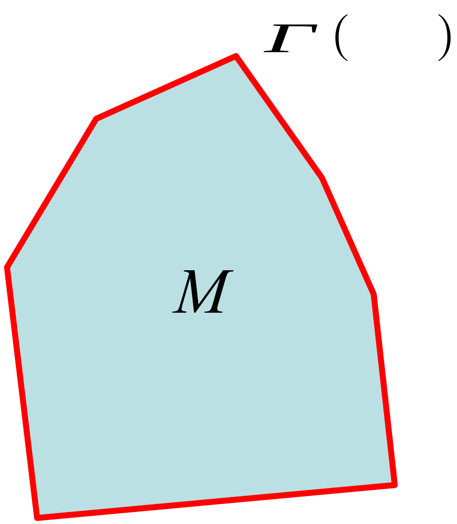
– скалярное произведение

Допустимый многогранник

– номер строки матрицы

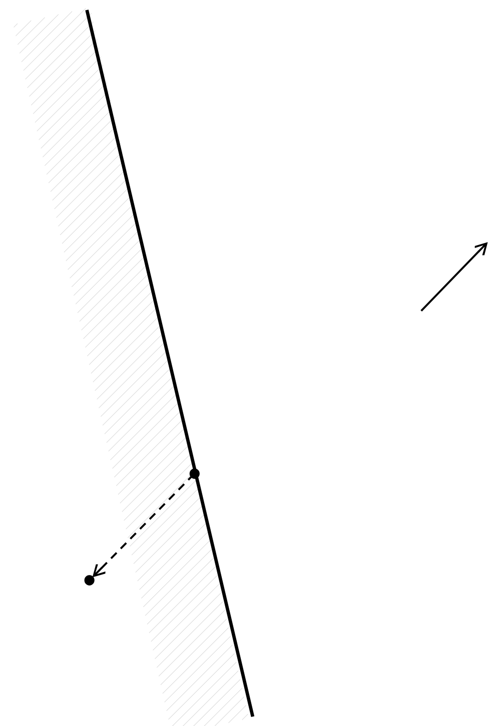
$=¿ \quad =1¿ \quad \wedge$

Обозначим через множество граничных точек допустимого многогранника



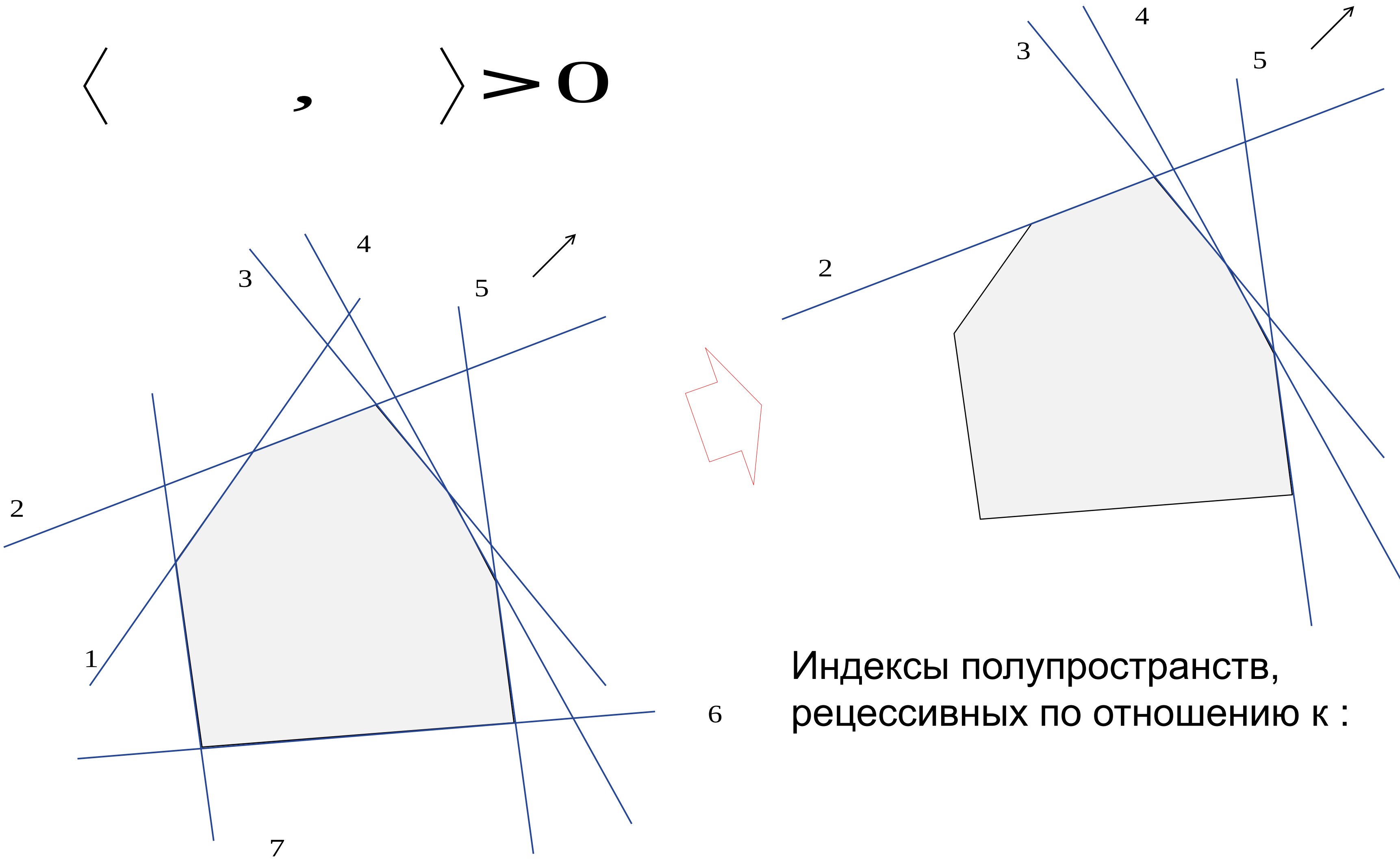
Рецессивное полупространство

$$\forall \quad \in \quad , \forall \quad \in \mathbb{R}_{¿0} : \quad + \quad \notin \hat{}$$



Критерий рецессивности

$$\langle \quad , \quad \rangle > 0$$

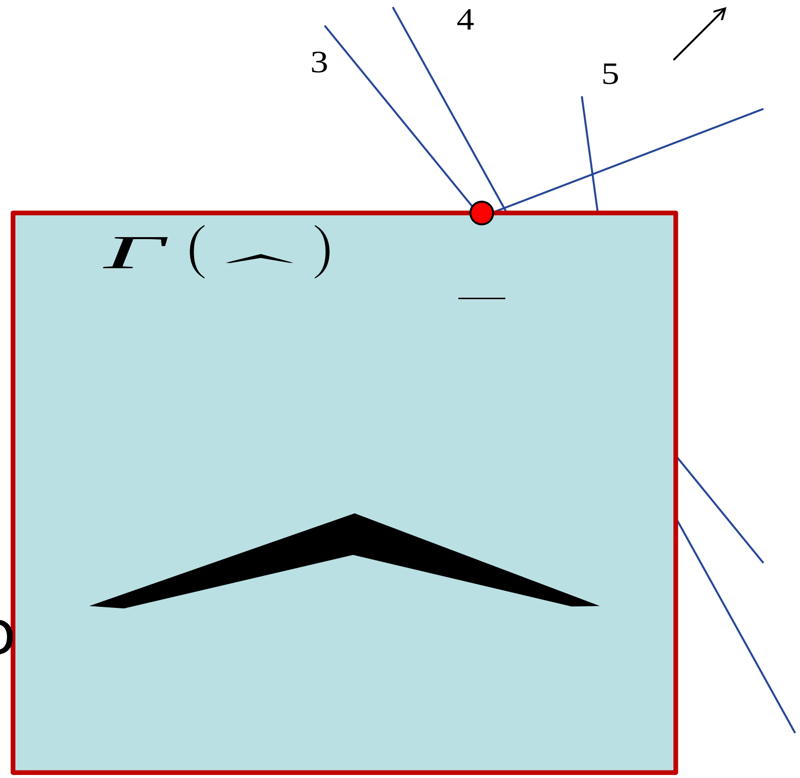


Индексы полупространств, рецессивных по отношению к :

Рецессивный многогранник

$\frown = ¿ \quad \in \mathcal{J} \; \frown$

Обозначим через множество граничных точек рецессивного многогранника



Ортогональная проекция

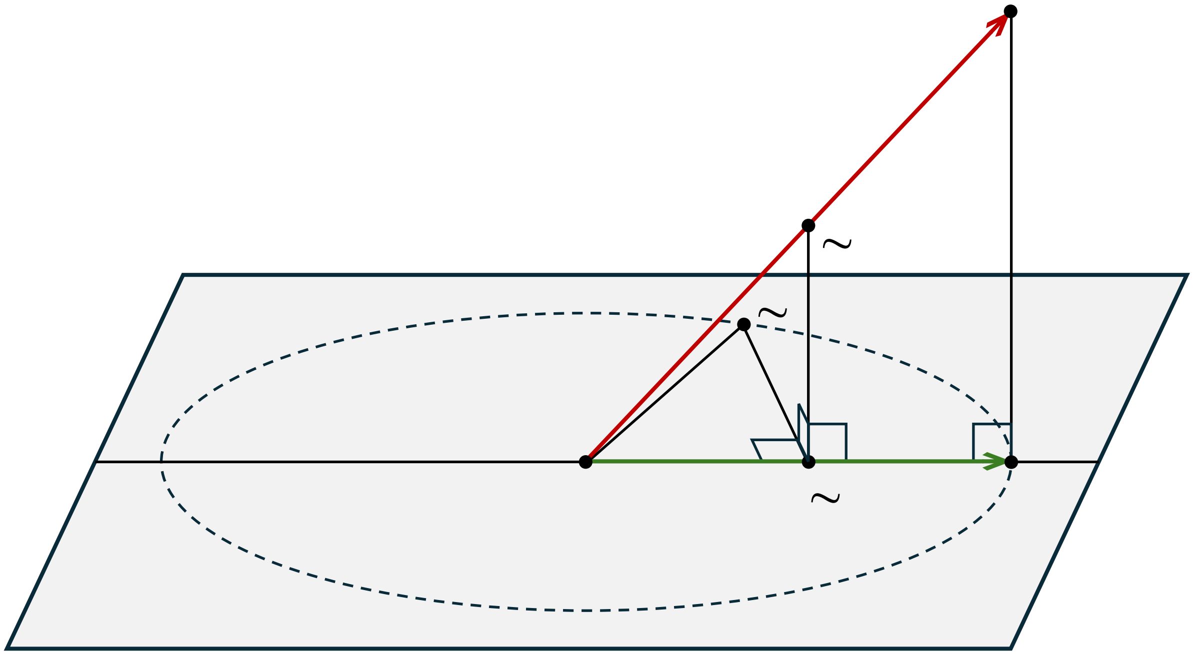
Пусть в пространстве имеется гиперплоскость

$$=\left\{ \quad \in I\!R \quad \middle| \langle \quad , \quad \rangle = \quad \right\}$$

Ортогональная проекция точки на гиперплоскость определяется формулой

$$(\quad) = \quad - \frac{\langle \quad , \quad \rangle -}{\| \quad \|^{2}} \qquad .$$

Оптимальный путь



Линейное многообразие

Пусть , , и . В этом случае множество индексов определяет в пространстве линейное многообразие

$$= ¿ \in$$

— размерность линейного многообразия .

Проблема:

Проекционное отображение

Определим проекционное отображение :

$$(\quad) = \frac{1}{|\quad|} \sum_{\in} \quad (\quad)$$

Известно, что отображение является непрерывным
-фейеровским отображением,
и последовательность точек

$$\{\quad = \quad (\quad_0)\}^{\infty}_{=1}$$

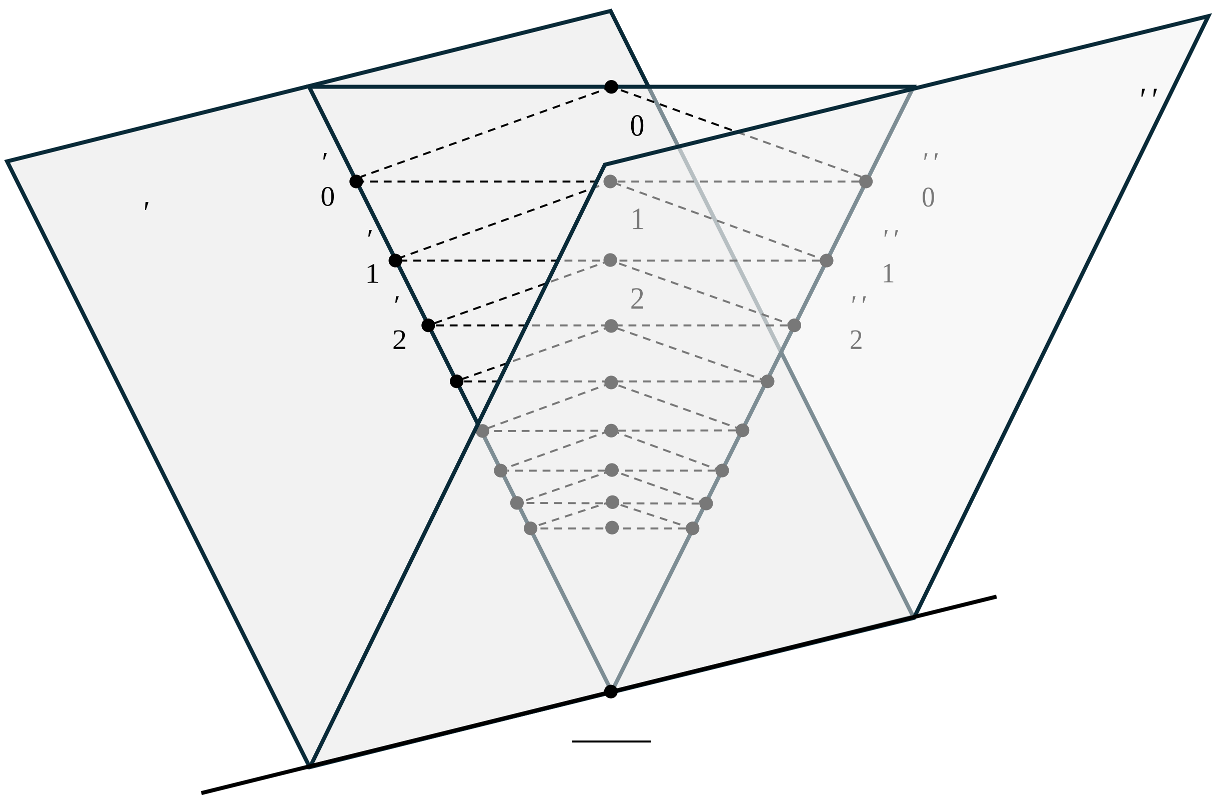
порождаемая этим отображением, начиная с произвольной точки ,
сходится к точке, принадлежащей :

$$\to \sim \in$$

Псевдопроекция

$$\lim_{\to\infty} \left\| \quad (\quad) - \quad (\quad) \right\| = 0$$

$$\forall \quad \in \mathbb{R} : \quad (\quad) = \quad (\quad)$$



Алгоритм движения по граням AlFaMove

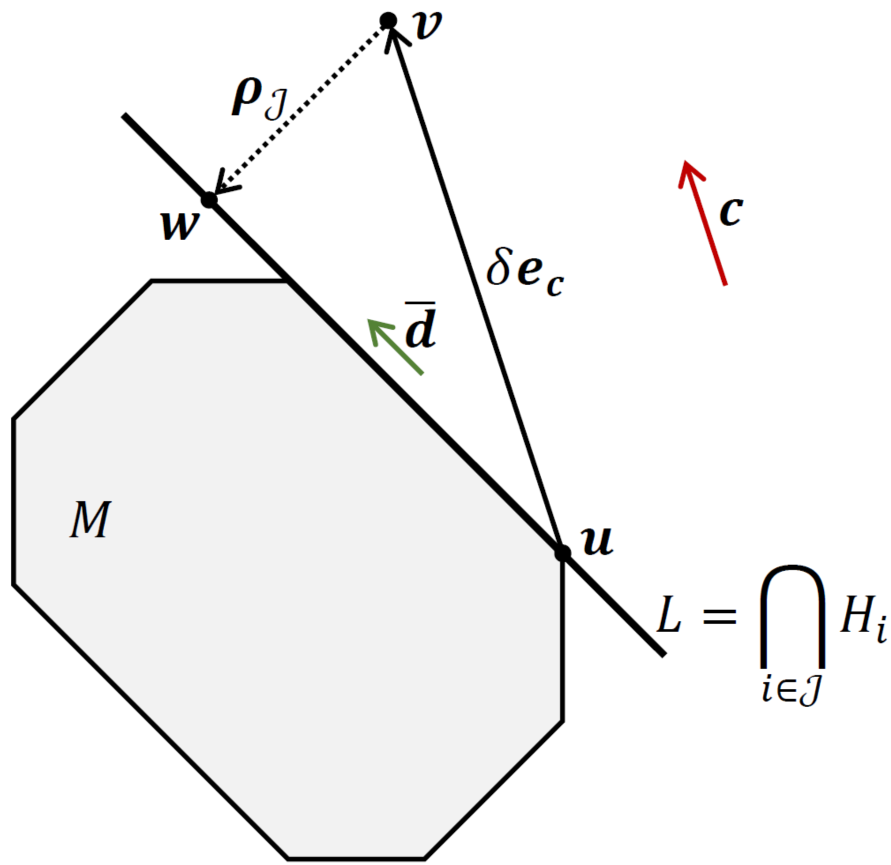
Алгоритм 3 Алгоритм движения по граням AlFaMove

Require: $\hat{H}_i = \{\boldsymbol{x} \in \mathbb{R}^n | \langle \boldsymbol{a}_i, \boldsymbol{x} \rangle \leqslant b_i\}$; $M = \bigcap_{i=1}^{m} \hat{H}_i$; $\hat{M} = \bigcap_{i \in \mathcal{I}} \hat{H}_i$; $i \in \mathcal{I} \Leftrightarrow \langle \boldsymbol{a}_i, \boldsymbol{c} \rangle > 0$

input $\boldsymbol{u}_0$
assert $\boldsymbol{u}_0 \in M \cap \Gamma(\hat{M})$
$\boldsymbol{d}_0 := \boldsymbol{D}(\boldsymbol{u}_0)$
assert $\boldsymbol{d}_0 \neq \boldsymbol{0}$
$k := 0$
repeat
  $\boldsymbol{u}_{k+1} := \boldsymbol{\mu}(\boldsymbol{u}_k, \boldsymbol{d}_k)$
  $\boldsymbol{d}_{k+1} := \boldsymbol{D}(\boldsymbol{u}_{k+1})$
  $k := k + 1$
until $\boldsymbol{d}_k = \boldsymbol{0}$
output $\boldsymbol{u}_k$ ▷ Решение задачи ЛП (1)
stop

Вычисление вектора движения

Вычисление вектора направления движения по грани линейного многообразия .



Процедура

Алгоритм 2 Вычисление вектора движения $\bar{\boldsymbol{d}} = \boldsymbol{D}(\boldsymbol{u})$

Require: $H_i = \{\boldsymbol{x} \in \mathbb{R}^n | \langle \boldsymbol{a}_i, \boldsymbol{x} \rangle = b_i\}$; $\boldsymbol{u} \in \Gamma(M)$

function $\boldsymbol{D}(\boldsymbol{u})$
    $\mathcal{U} := \emptyset$ ▷ $\mathcal{U}$ — множество индексов гиперплоскостей H_i, проходящих через точку $\boldsymbol{u}$
    for $i = 1 \ldots m$ **do**
        if $\langle \boldsymbol{a}_i, \boldsymbol{u} \rangle = b_i$ **then**
            $\mathcal{U} := \mathcal{U} \cup \{i\}$
        end if
    end for
    $\bar{\boldsymbol{d}} := \boldsymbol{0}$
    $f := -\infty$ ▷ f — значение целевой функции $f(\boldsymbol{x}) = \langle \boldsymbol{c}, \boldsymbol{x} \rangle$
    $\boldsymbol{e_c} := \boldsymbol{c}/\|\boldsymbol{c}\|$
    $\boldsymbol{v} := \boldsymbol{u} + \delta \boldsymbol{e_c}$ ▷ Большой параметр $\delta > 0$
    for $\mathcal{J} \in \mathcal{P}(\mathcal{U}) \backslash \emptyset$ **do** ▷ $\mathcal{P}(\mathcal{U})$ — множество всех подмножеств множества $\mathcal{U}$
        $\boldsymbol{w} := \boldsymbol{\rho}_{\mathcal{J}}(\boldsymbol{v})$
        $\boldsymbol{d} := \boldsymbol{w} - \boldsymbol{u}$
        $\boldsymbol{e_d} := \boldsymbol{d}/\|\boldsymbol{d}\|$
        if $(\boldsymbol{u} + \tau \boldsymbol{e_d}) \in M$ **then** ▷ Малый параметр $\tau > 0$
            if $\langle \boldsymbol{c}, \boldsymbol{u} + \tau \boldsymbol{e_d} \rangle > f$ **then**
                $f := \langle \boldsymbol{c}, \boldsymbol{u} + \tau \boldsymbol{e_d} \rangle$
                $\bar{\boldsymbol{d}} := \boldsymbol{d}$
            end if
        end if
    end for
    return $\bar{\boldsymbol{d}}$
end function

Вектор-функция

Обозначим

$$=\{\quad\in\{1,\dots,\quad\}|\langle\quad,\quad\rangle<\quad\langle\quad,\quad\rangle>0\quad\}$$

тогда

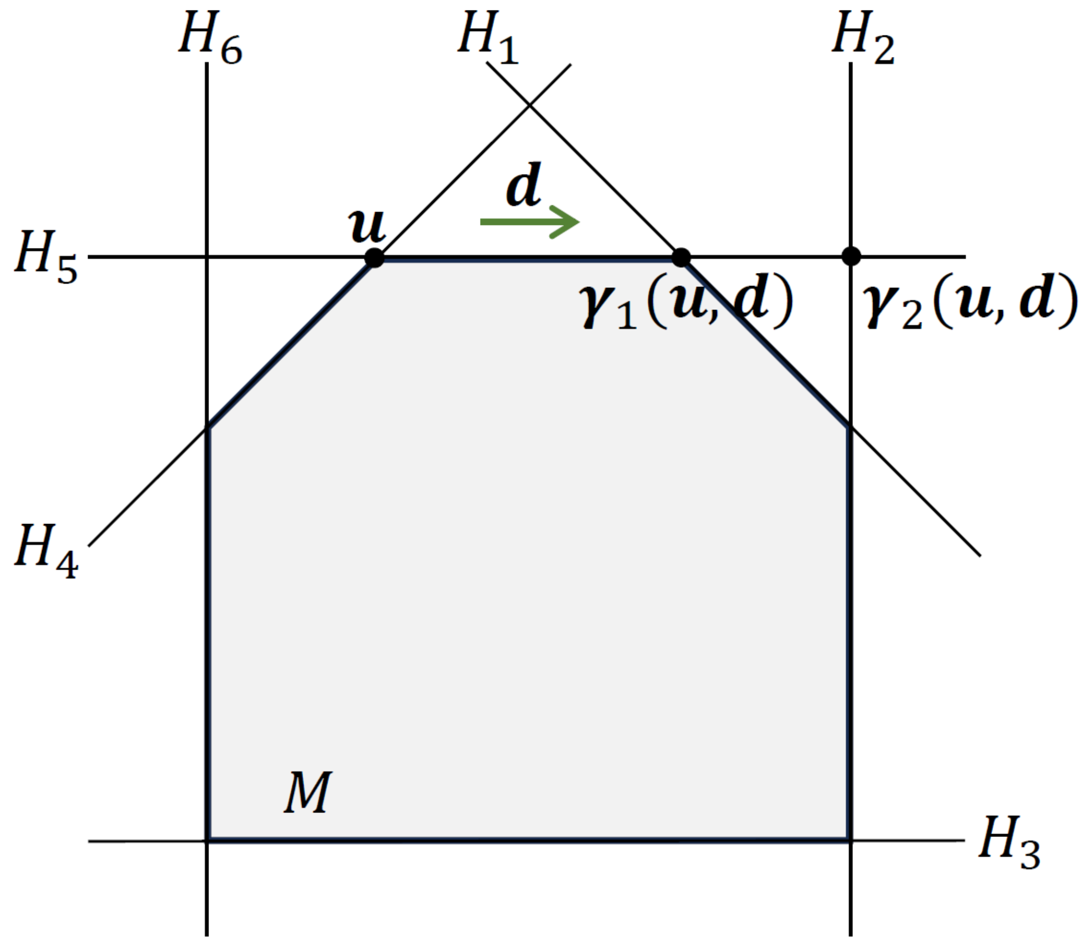
$$(\quad,\quad)=\arg\min_{\in}\{\|\quad-\quad\|\|\quad=\quad(\quad,\quad)\}$$

Здесь обозначает вектор-функцию, вычисляющую косоугольную проекцию точки на гиперплоскость по направлению :

$$(\quad,\quad)=\quad-\frac{\langle\quad,\quad\rangle-}{\langle\quad,\quad\rangle}\quad.$$

Вектор-функция

Действие функции :



Параллельный алгоритм

Основные моменты:

- BSF-каркас: мастер-рабочие
- Функции высшего порядка *Map* и *Reduce*
- Список , где

Репозитории:

- Параллельная реализация AlFaMove: https://github.com/nikolay-olkhovsky/AlFaMove
- Тестовые задачи ЛП: https://github.com/nikolay-olkhovsky/Set-of-LP-Problems

Вычислительные эксперименты

Суперкомпьютер «Торнадо ЮУрГУ»

Количество узлов:	480
Тип процессоров:	Intel Xeon X5680 (6 ядер, 3.33 ГГц)
Число процессоров на узел:	2
Память на узел:	25 ГБ DDR3
Память сопроцессора:	8 Гб
Тип системной сети:	InfiniBand QDR (40 Гбит/с)
Операционная система:	Linux CentOS 6.2

Гиперкуб с отсеченной вершиной

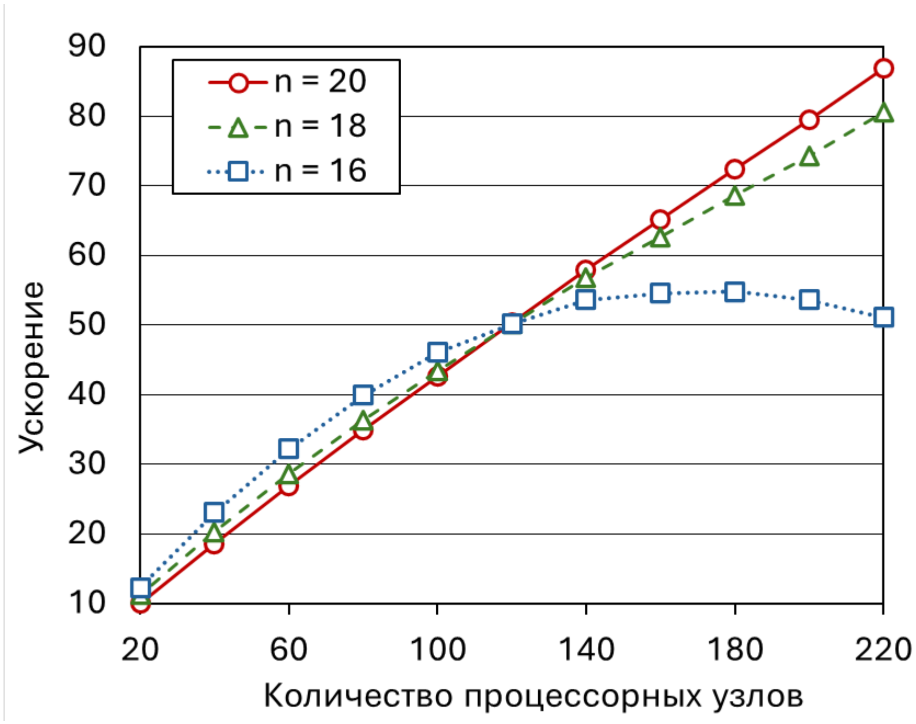
$$
\begin{cases}
x_1 & & & & & \leqslant & 200 \\
& x_2 & & & & \leqslant & 200 \\
& & & & & \vdots & \\
& & & x_n & & \leqslant & 200 \\
x_1 + & x_2 & \cdots & + & x_n & \leqslant & 200(n-1)+100 \\
x_1 \geqslant 0, & x_2 \geqslant 0, & \cdots & , x_n \geqslant 0. & & &
\end{cases}
$$

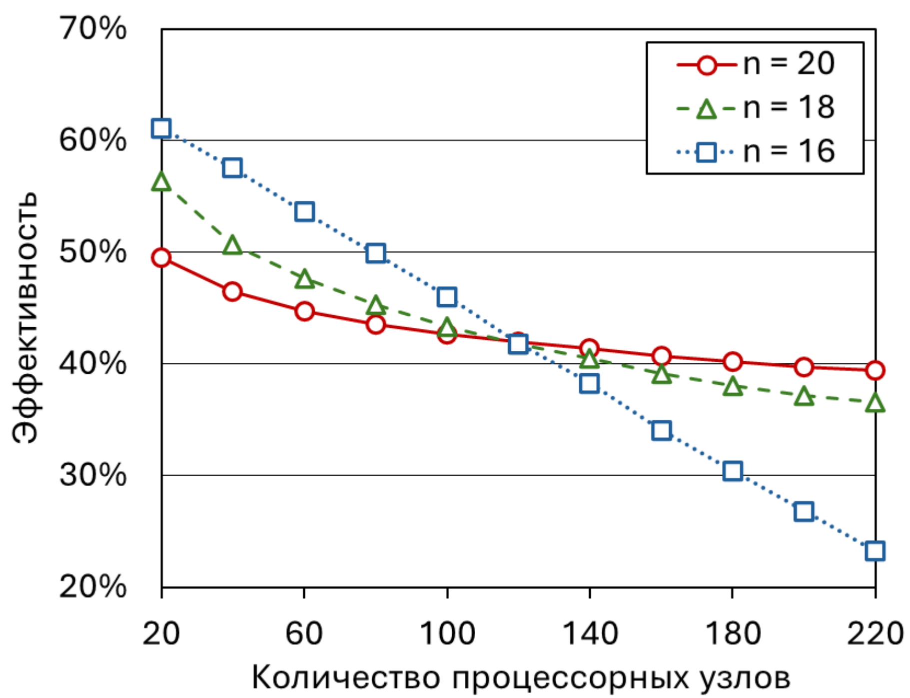
Градиент целевой функции: $\boldsymbol{c} = (1, 2, \ldots, n)$

Стартовая точка:

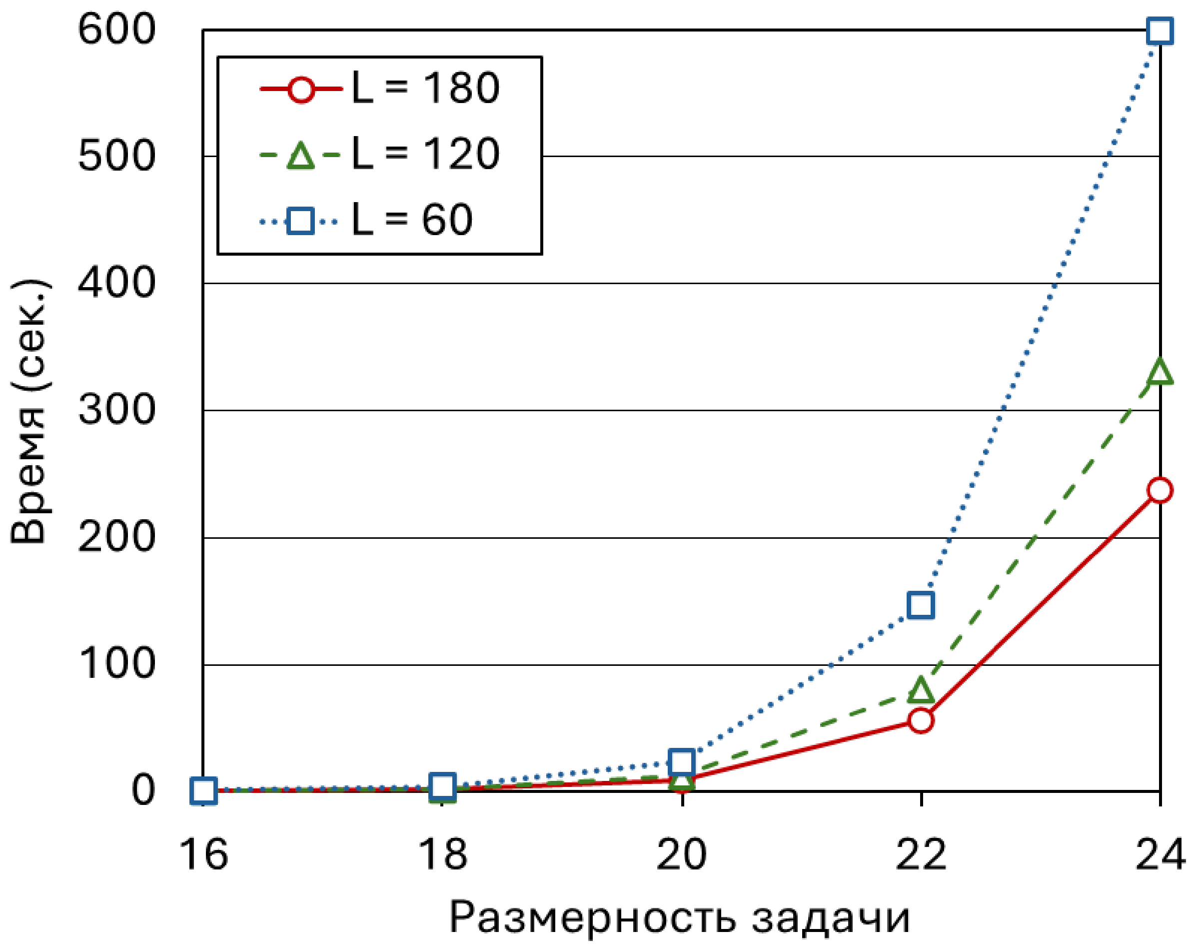
$$x_1 = 0, \quad \ldots \quad x_{n/2} = 0, \quad x_{n/2+1} = 200, \quad \ldots \quad x_n = 200$$

Ускорение и эффективность





Время работы алгоритма



Куб Кли-Минти

$$
\begin{cases}
x_1 & & & & & & & \leqslant & 5 \\
4x_1 & + & x_2 & & & & & \leqslant & 25 \\
8x_1 & + & 4x_2 & + & x_3 & & & \leqslant & 125 \\
& & & & & & & \vdots & \\
2^n x_1 + & & 2^{n-1}x_2 & \cdots & + & 4x_{n-1} & + & x_n & \leqslant & 5^n \\
x_1 \geqslant 0, & & x_2 \geqslant 0, & \cdots & & , x_n \geqslant 0. & & & &
\end{cases}
$$

Градиент целевой функции: $\boldsymbol{c} = \left(2^{n-1}, 2^{n-2}, \ldots, 2, 1\right)$

Стартовая точка: начало координат.

Сравнение с симплекс-методом

Размер-ность	AlFaMove				Simplex
	Граница масшт-ти	Время (сек.)	Относит. погреш-ть	Кол-во итераций	Кол-во итераций
5	10	0.2	$0.9 \cdot 10^{-12}$	9	31
6	15	2	$0.2 \cdot 10^{-12}$	11	63
7	20	13	$0.8 \cdot 10^{-11}$	13	127
8	25	126	$0.8 \cdot 10^{-11}$	15	255
9	30	1445	$0.2 \cdot 10^{-10}$	17	511

Благодарю за внимание!

Николай Ольховский
89226343351 (Telegram, Viber, WhatsApp)
olkhovskiina@susu.ru