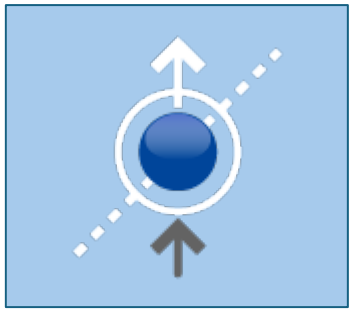


Масштабируемые проекционные методы в линейном программировании

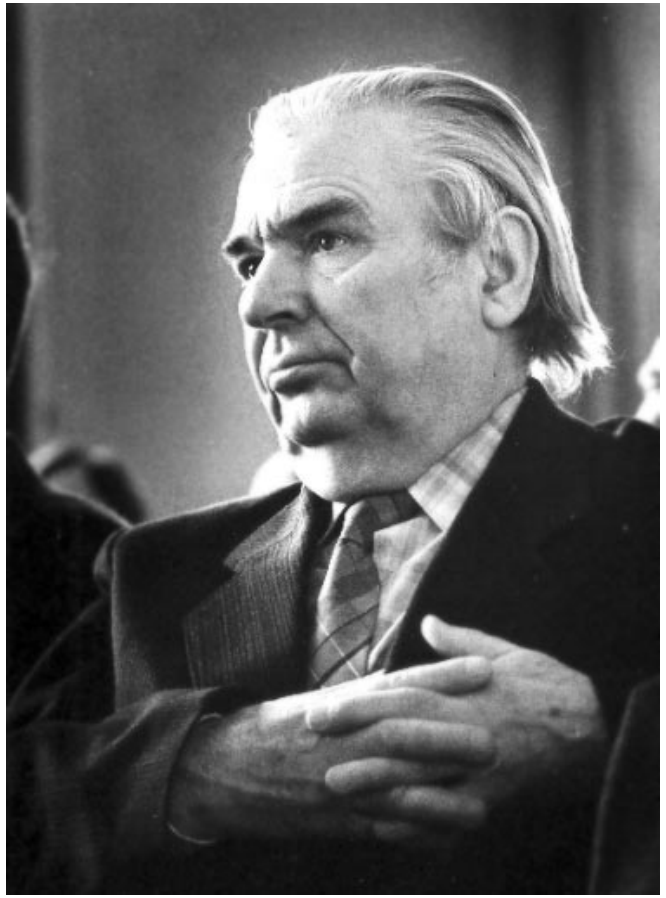
Соколинский Леонид Борисович

Южно-Уральский государственный университет
(национальный исследовательский университет)

Челябинск

Исследование выполнено при поддержке РНФ (грант № 23-21-00356)

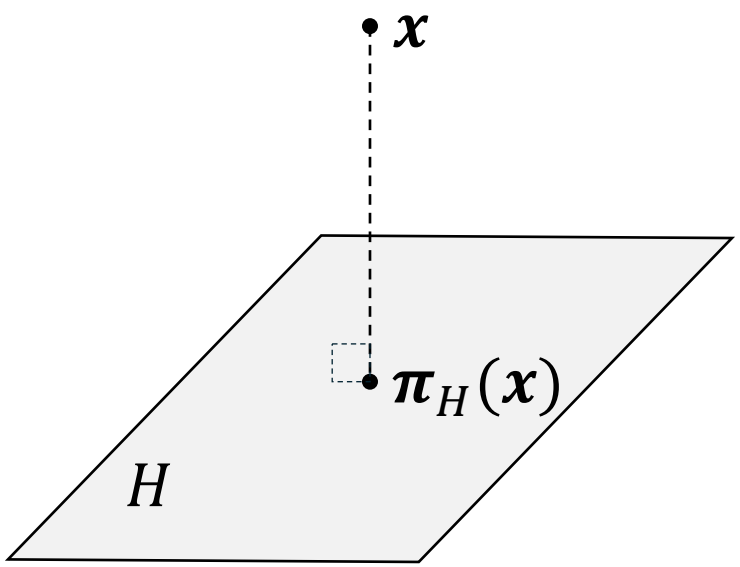
Истоки



1933 - 2013

- Академик РАН **Ерёмин Иван Иванович**
  https://ru.wikipedia.org/wiki/Ерёмин,_Иван_Иванович
- Проект LiFe http://life.susu.ru/

Ортогональная проекция на гиперплоскость в $\mathbb{R}^n$



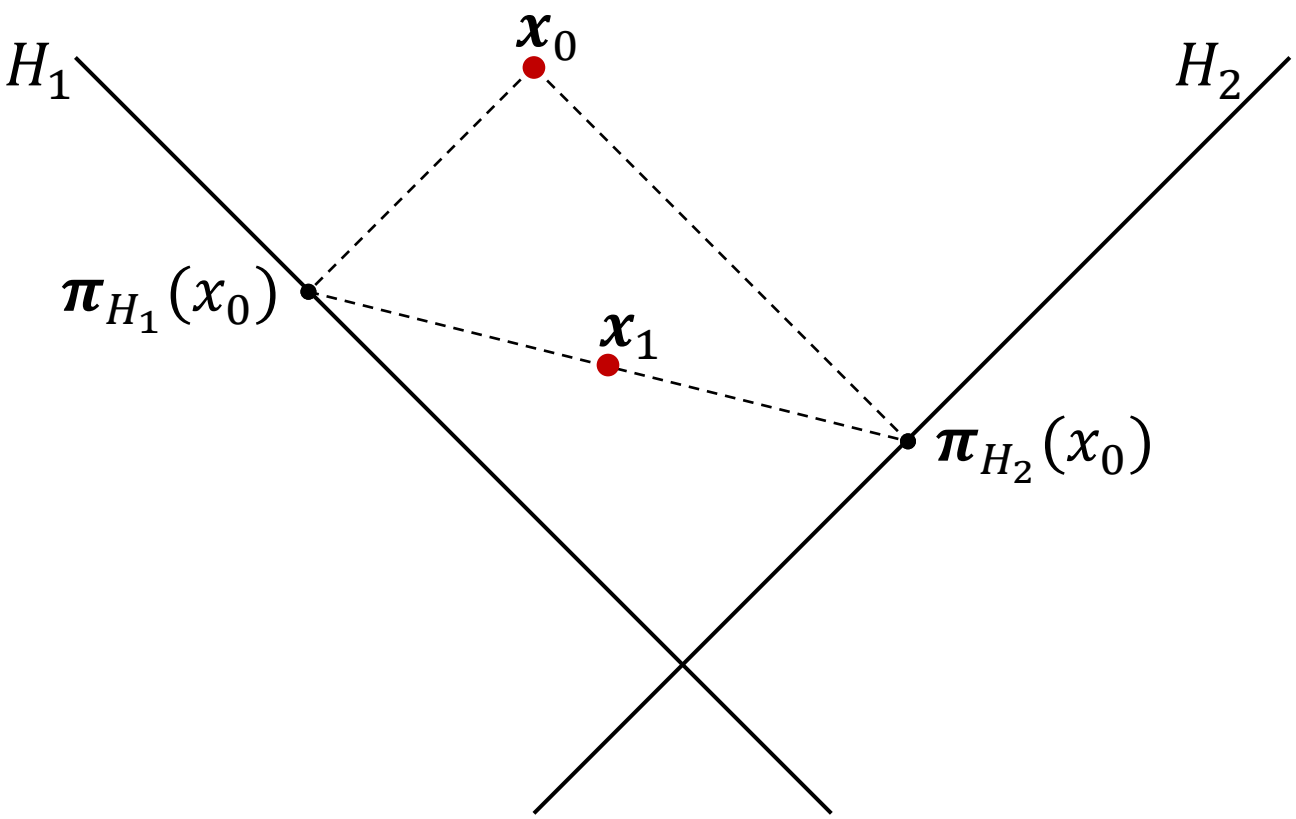
$$a_1 x_1 + \cdots + a_n x_n = b$$

$$\langle \boldsymbol{a}, \boldsymbol{x} \rangle = b$$

$$H = \{ \boldsymbol{x} \in \mathbb{R}^n | \langle \boldsymbol{a}, \boldsymbol{x} \rangle = b \}$$

$$\boldsymbol{\pi}_H(\boldsymbol{x}) = \boldsymbol{x} - \frac{\langle \boldsymbol{a}, \boldsymbol{x} \rangle - b}{\|\boldsymbol{a}\|^2} \boldsymbol{a}$$

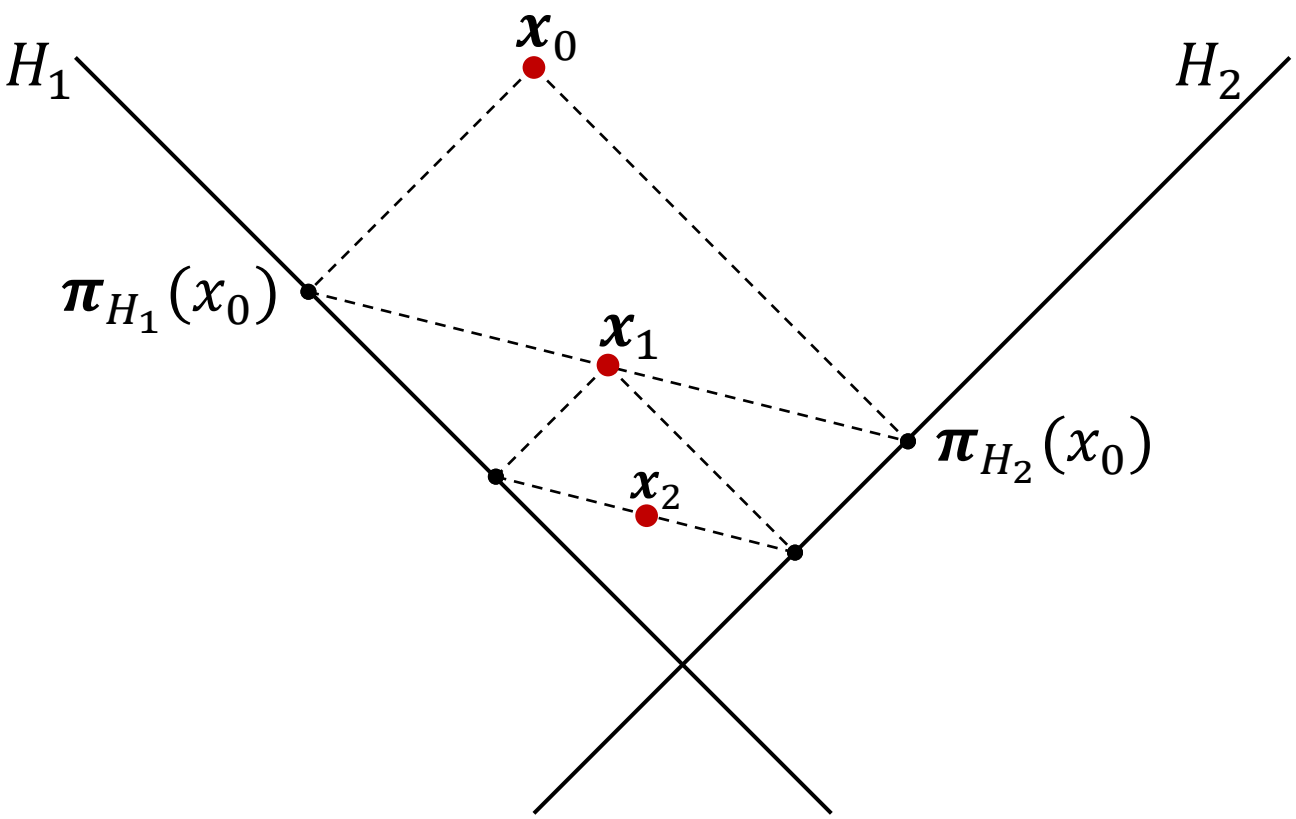
Идея проекционного метода



$$\varphi(x) = \frac{1}{2}\left(\pi_{H_1}(x) + \pi_{H_2}(x)\right)$$

$$x_1 = \varphi(x_0)$$

Идея проекционного метода

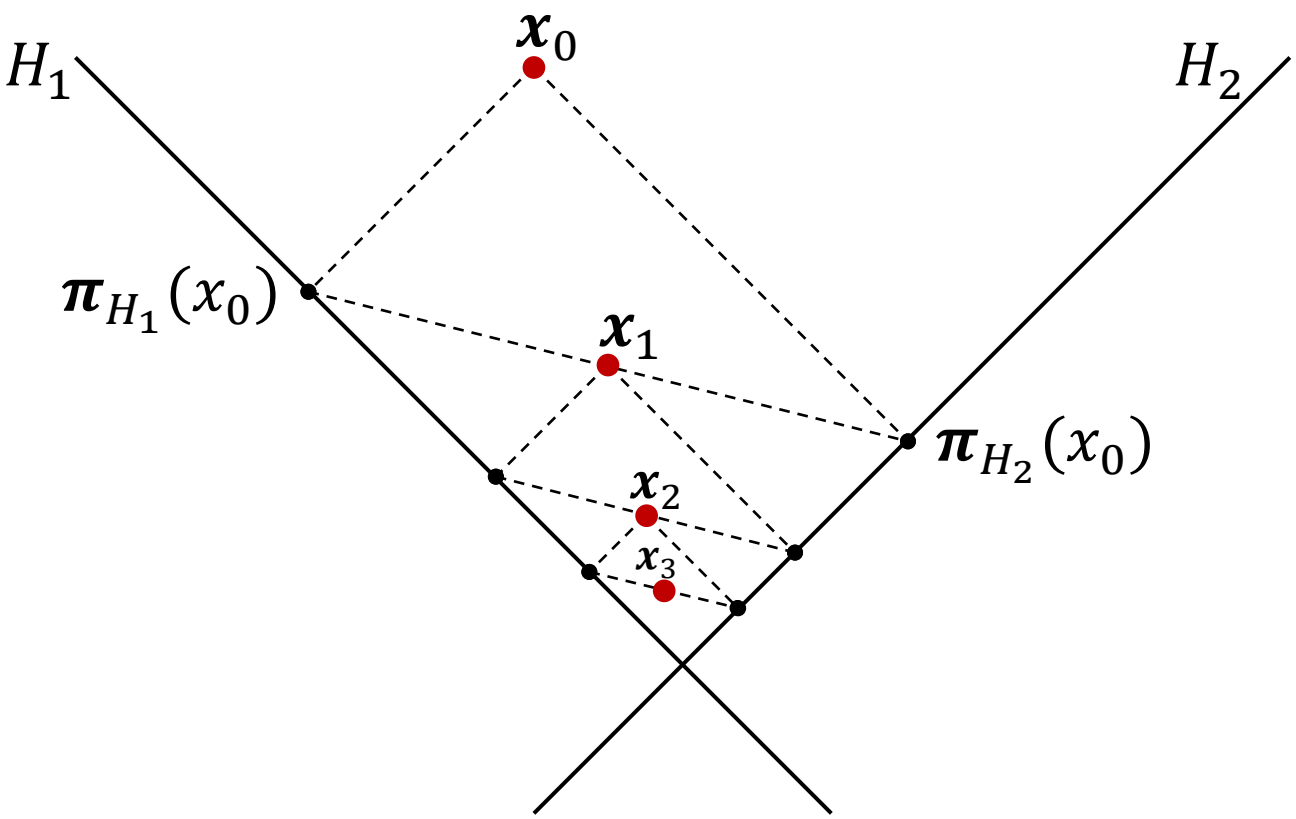


$$\varphi(x) = \frac{1}{2}\left(\pi_{H_1}(x) + \pi_{H_2}(x)\right)$$

$$x_1 = \varphi(x_0)$$

$$x_2 = \varphi(x_1)$$

Идея проекционного метода



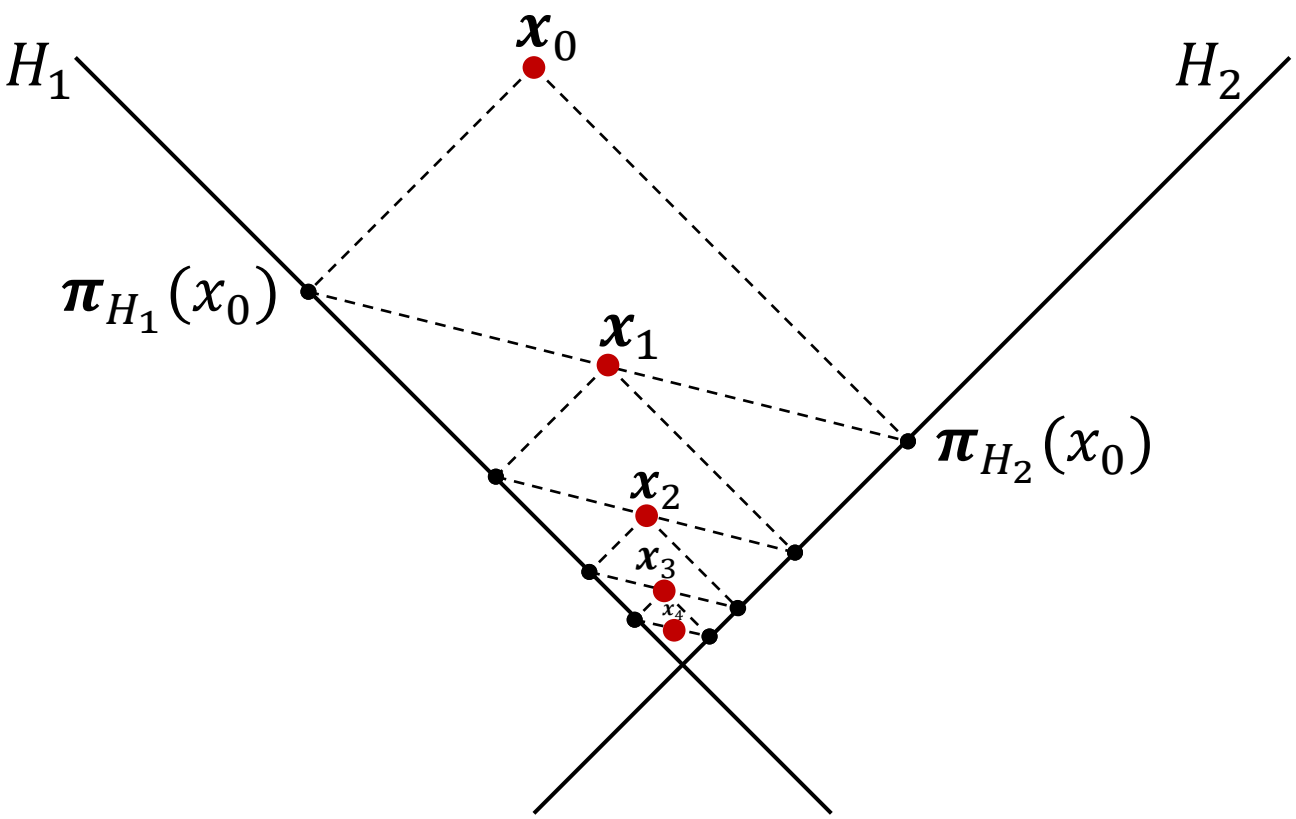
$$\varphi(x) = \frac{1}{2}\left(\pi_{H_1}(x) + \pi_{H_2}(x)\right)$$

$$x_1 = \varphi(x_0)$$

$$x_2 = \varphi(x_1)$$

$$x_3 = \varphi(x_2)$$

Идея проекционного метода



$$\varphi(x) = \frac{1}{2}\left(\pi_{H_1}(x) + \pi_{H_2}(x)\right)$$

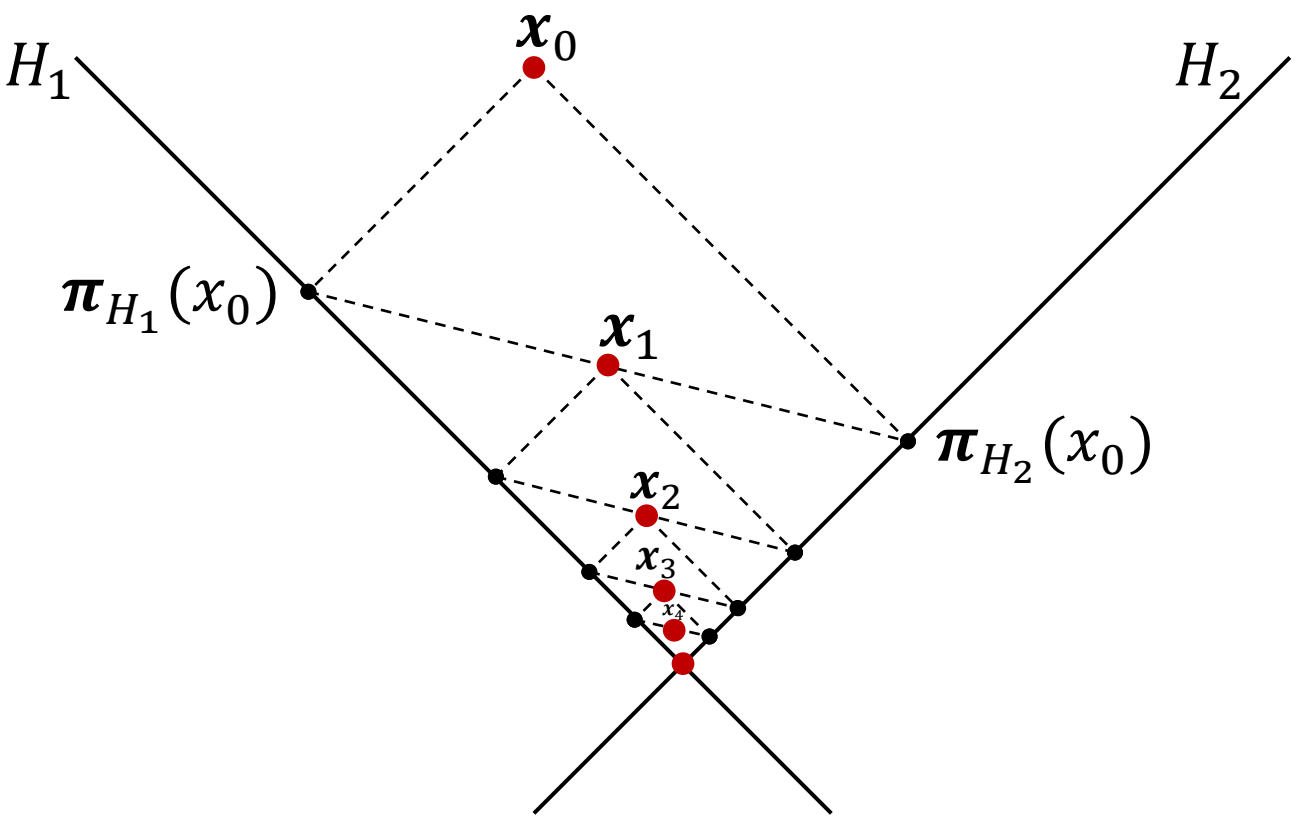
$$x_1 = \varphi(x_0)$$

$$x_2 = \varphi(x_1)$$

$$x_3 = \varphi(x_2)$$

$$x_4 = \varphi(x_3)$$

Идея проекционного метода



$$\varphi(x) = \frac{1}{2}\left(\pi_{H_1}(x) + \pi_{H_2}(x)\right)$$

$$x_1 = \varphi(x_0)$$

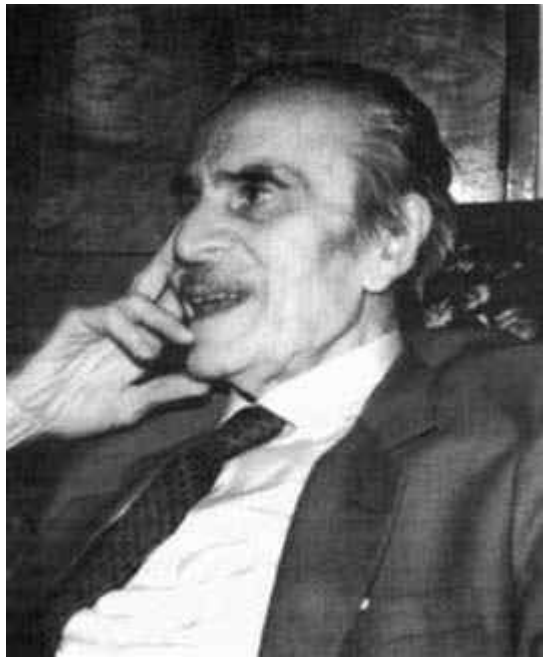
$$x_2 = \varphi(x_1)$$

$$x_3 = \varphi(x_2)$$

$$x_4 = \varphi(x_3)$$

...

Решение системы линейных уравнений



Gianfranco Cimmino

Cimmino G. Calcolo approssimato per le soluzioni dei sistemi di equazioni lineari // La Ric. Sci. XVI, Ser. II, Anno IX, 1. **1938**. P. 326–333.

$$A\boldsymbol{x} = \boldsymbol{b}$$

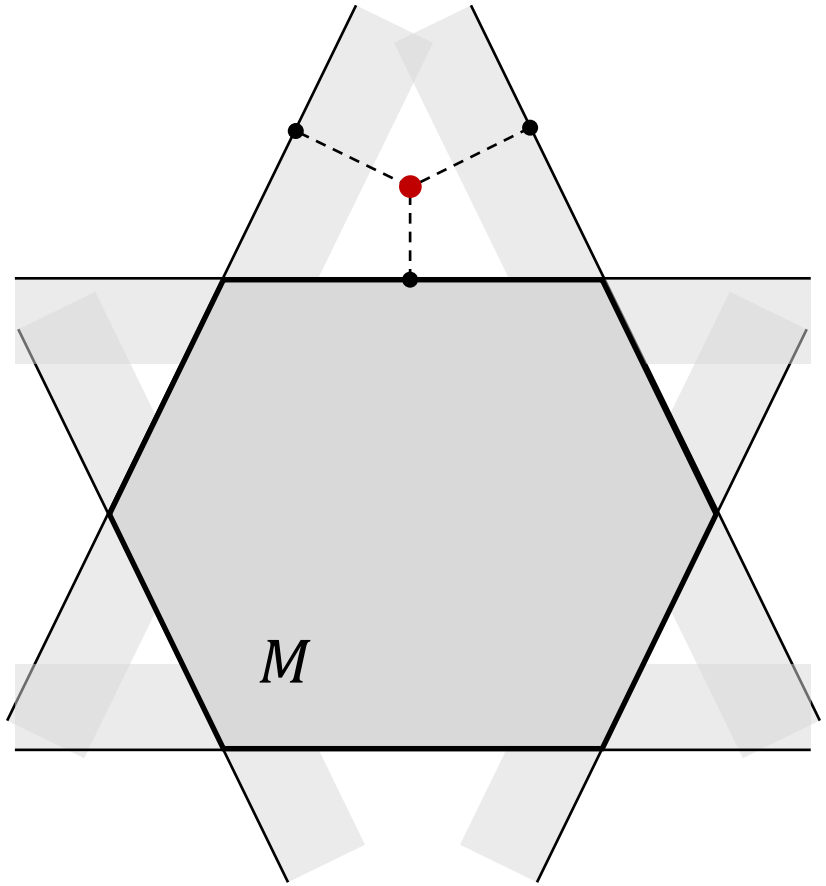
$$A \in \mathbb{R}^m \times \mathbb{R}^n, A = \begin{pmatrix} \boldsymbol{a}_1 \\ \vdots \\ \boldsymbol{a}_m \end{pmatrix}$$

$$\boldsymbol{\varphi}(\boldsymbol{x}) = \boldsymbol{x} - \frac{1}{m}\sum_{i=1}^{m} \frac{\langle \boldsymbol{a}_i, \boldsymbol{x} \rangle - b_i}{\|\boldsymbol{a}_i\|^2} \boldsymbol{a}_i$$

$$\boldsymbol{\varphi}^1(\boldsymbol{x}) = \boldsymbol{\varphi}(\boldsymbol{x});\ \boldsymbol{\varphi}^k(\boldsymbol{x}) = \boldsymbol{\varphi}\left(\boldsymbol{\varphi}^{k-1}(\boldsymbol{x})\right)$$

$$\boldsymbol{\eta}(\boldsymbol{x}) = \lim_{k \to \infty} \boldsymbol{\varphi}^k(\boldsymbol{x})$$

Переход к линейным неравенствам



$$A\boldsymbol{x} \leq \boldsymbol{b}$$

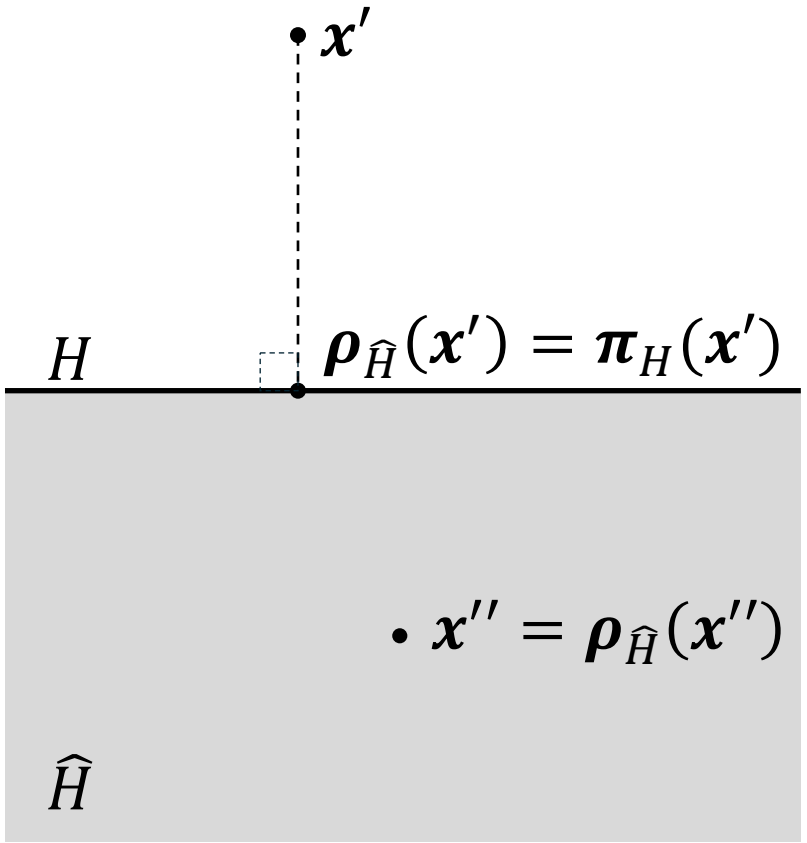
$$A \in \mathbb{R}^m \times \mathbb{R}^n, A = \begin{pmatrix} \boldsymbol{a}_1 \\ \vdots \\ \boldsymbol{a}_m \end{pmatrix}$$

$$\widehat{H}_i = \{\boldsymbol{x} \in \mathbb{R}^n | \langle \boldsymbol{a}_i, \boldsymbol{x} \rangle \leq b_i \}$$

$$H_i = \{\boldsymbol{x} \in \mathbb{R}^n | \langle \boldsymbol{a}_i, \boldsymbol{x} \rangle = b_i \}$$

$$(i = 1, \dots, m)$$

Проекция точки на полупространство в $\mathbb{R}^n$



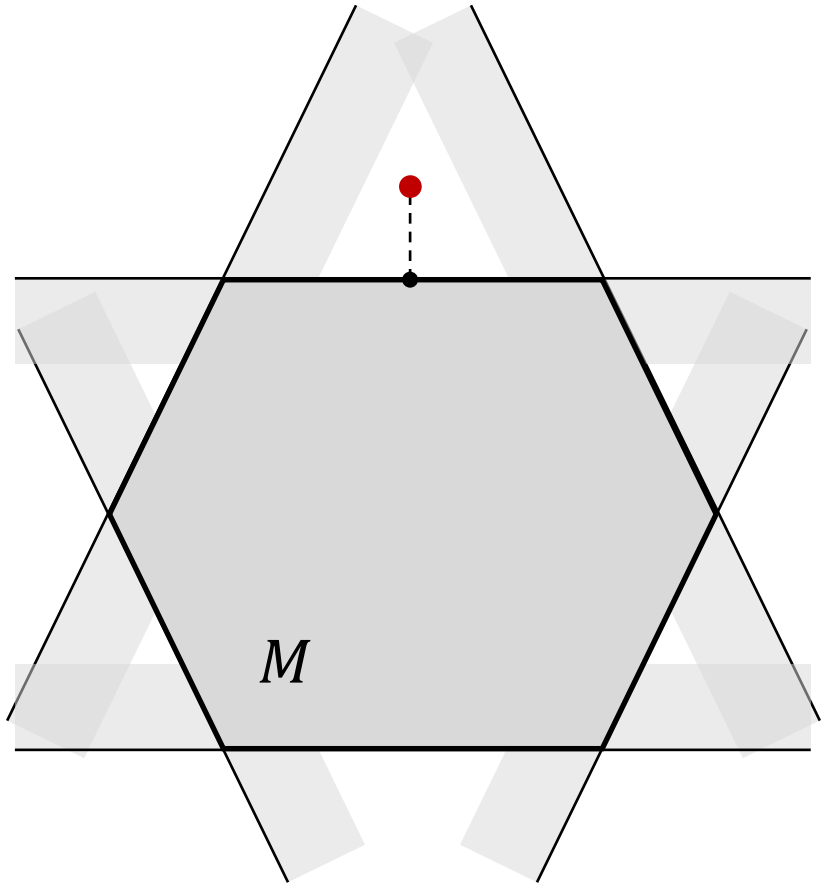
$$\widehat{H} = \{\boldsymbol{x} \in \mathbb{R}^n | \langle \boldsymbol{a}, \boldsymbol{x} \rangle \leq b \}$$

$$H = \{\boldsymbol{x} \in \mathbb{R}^n | \langle \boldsymbol{a}, \boldsymbol{x} \rangle = b \}$$

$$\boldsymbol{\pi}_H(\boldsymbol{x}) = \boldsymbol{x} - \frac{\langle \boldsymbol{a}, \boldsymbol{x} \rangle - b}{\|\boldsymbol{a}\|^2} \boldsymbol{a}$$

$$\boldsymbol{\rho}_{\widehat{H}}(\boldsymbol{x}) = \begin{cases} \boldsymbol{\pi}_H(\boldsymbol{x}) & \left| \boldsymbol{x} \notin \widehat{H} \right. \\ \boldsymbol{x} & \left| \boldsymbol{x} \in \widehat{H} \right. \end{cases}$$

Решение системы линейных неравенств



$$A\boldsymbol{x} \leq \boldsymbol{b}$$

$$A \in \mathbb{R}^m \times \mathbb{R}^n, A = \begin{pmatrix} \boldsymbol{a}_1 \\ \vdots \\ \boldsymbol{a}_m \end{pmatrix}$$

$$\boldsymbol{\varphi}(\boldsymbol{x}) = \boldsymbol{x} - \frac{1}{m} \sum_{i=1}^{m} \frac{\max(\langle \boldsymbol{a}_i, \boldsymbol{x} \rangle - b_i)}{\|\boldsymbol{a}_i\|^2} \boldsymbol{a}_i$$

Псевдопроекция: $\boldsymbol{\eta}_M(\boldsymbol{x}) = \lim_{k \to \infty} \boldsymbol{\varphi}^k(\boldsymbol{x})$

Применение к линейному программированию

$$\overline{\boldsymbol{x}} = \arg\max\{f(\boldsymbol{x}) | A\boldsymbol{x} \leq \boldsymbol{b}\}$$

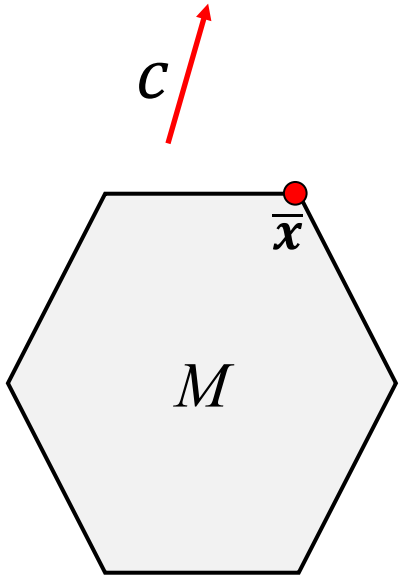
$\boldsymbol{x} \in \mathbb{R}^n$

A – матрица $m \times n$

$\boldsymbol{c}, \boldsymbol{b}$ – векторы размерности n

$f(\boldsymbol{x}) = \langle \boldsymbol{c}, \boldsymbol{x} \rangle$ – целевая функция

$\langle \boldsymbol{c}, \boldsymbol{x} \rangle$ – скалярное произведение

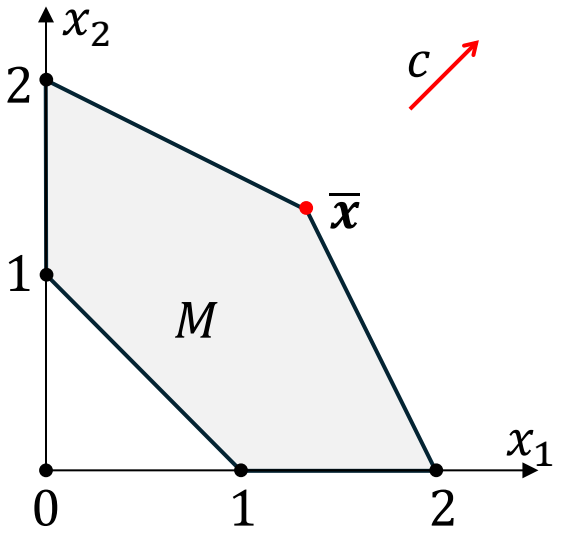


Допустимый многогранник: $M = \{\boldsymbol{x} | A\boldsymbol{x} \leq \boldsymbol{b}\}$

$$f(x_1, x_2) = x_1 + x_2 \to max$$

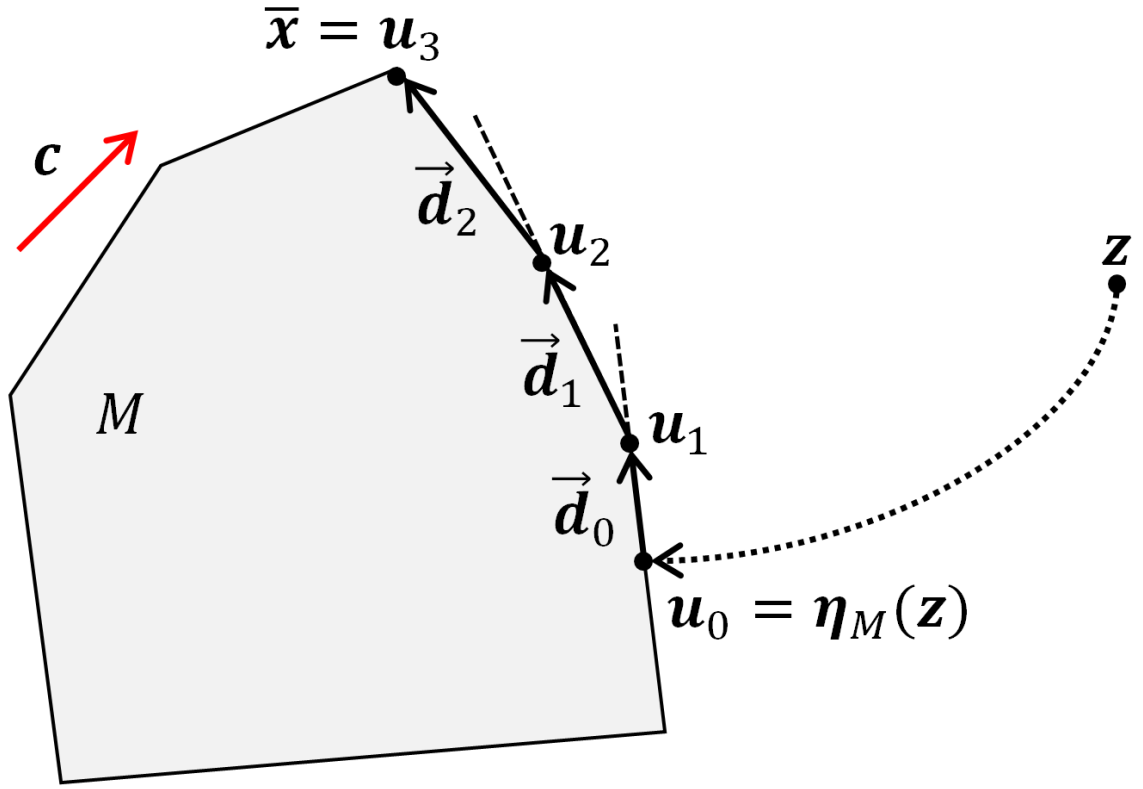
$$\begin{cases} x_1 + 2x_2 \leq 2 \\ 2x_1 + x_2 \leq 2 \\ x_1 + x_2 \geq 1 \\ x_1 \geq 0 \\ x_2 \geq 0 \end{cases}$$

$$\boldsymbol{c} = (1; 1)$$



Апекс-метод

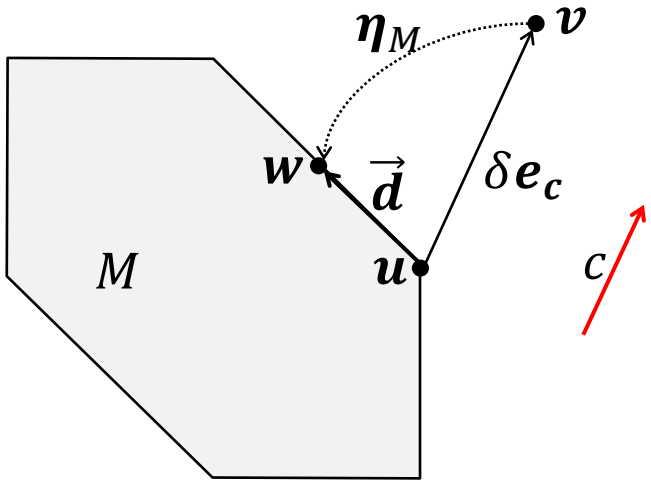
1. $\boldsymbol{u}_0 := \boldsymbol{\eta}_M(\boldsymbol{z})$
2. $i := 0$
3. Вычисляем вектор $\vec{\boldsymbol{d}}_i$ направления движения в сторону увеличения целевой функции
4. **if** $\vec{\boldsymbol{d}}_i = \boldsymbol{0}$ **goto** 8
5. Двигаемся до $\boldsymbol{u}_{i+1}$ – точки отрыва от поверхности допустимого многогранника M
6. $i := i + 1$
7. **goto** 3
8. **stop**



Вычисление вектора направления движения

1. $\boldsymbol{e_c} \coloneqq \dfrac{\boldsymbol{c}}{\|\boldsymbol{c}\|}$
2. $\boldsymbol{v} \coloneqq \boldsymbol{u} + \delta \boldsymbol{e_c}$
3. $\boldsymbol{w} \coloneqq \boldsymbol{\eta}_M(\boldsymbol{v})$
4. **if** $\langle \boldsymbol{c}, \boldsymbol{w} \rangle \leq \langle \boldsymbol{c}, \boldsymbol{u} \rangle$ **goto** 7
5. $\vec{\boldsymbol{d}} \coloneqq \boldsymbol{w} - \boldsymbol{u}$
6. **goto** 8
7. $\vec{\boldsymbol{d}} \coloneqq \boldsymbol{0}$
8. **stop**

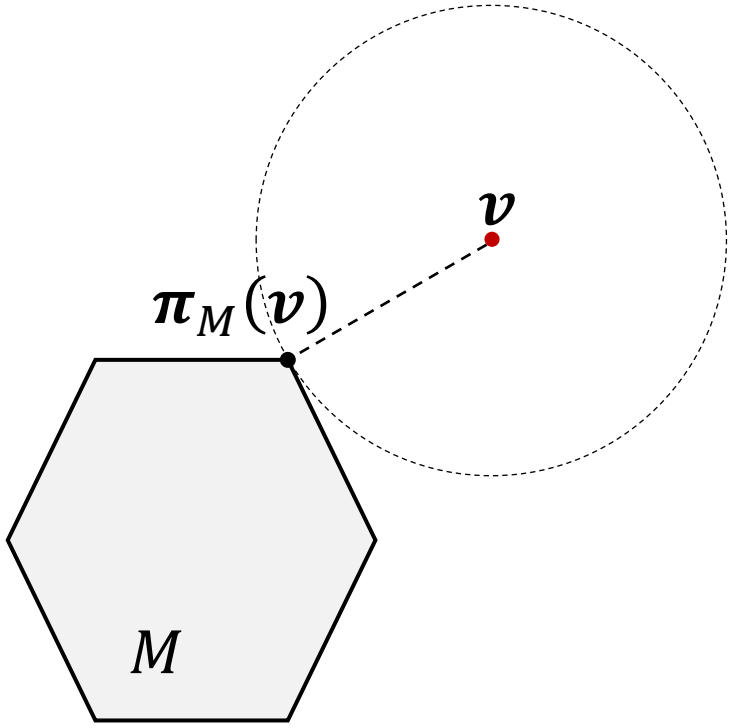
$\delta > 0$



Почему это работает

Метрическая проекция на выпуклое замкнутое множество:

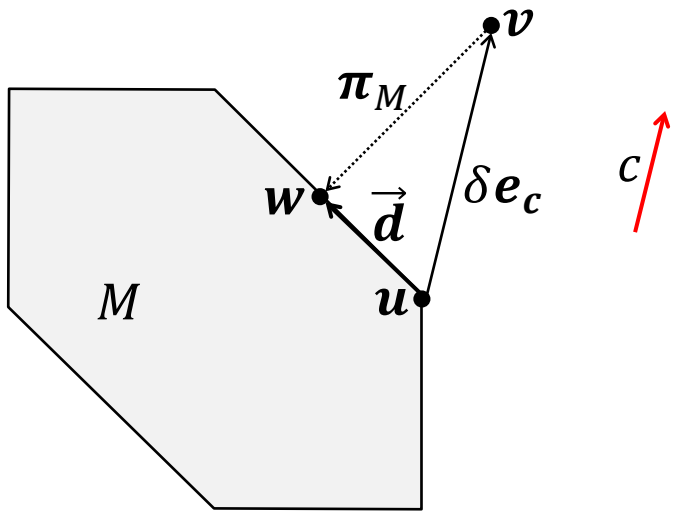
$$\boldsymbol{\pi}_M(\boldsymbol{v}) = \operatorname{Arg}\min_{\boldsymbol{x}\in M}\|\boldsymbol{v}-\boldsymbol{x}\|$$



Утверждение 1. Если в апекс-методе заменить псевдопроекцию на метрическую проекцию, то итерационный процесс сходится за конечное число шагов к решению задачи ЛП

Утверждение 2.

$$\lim_{\delta\to 0}\|\boldsymbol{\pi}_M(\boldsymbol{v})-\boldsymbol{\eta}_M(\boldsymbol{v})\| = 0$$



Параллельная реализация апекс-метода

- Использован параллельный программный каркас BSF: https://github.com/leonid-sokolinsky/BSF-skeleton
  - C++
  - MPI
- Исходные коды: https://github.com/leonid-sokolinsky/Apex-method

Характеристики вычислительного кластера «Торнадо ЮУрГУ»

Количество процессорных узлов	480
Процессоры	Intel Xeon X5680 (6 cores 3.33 GHz)
Количество процессоров в узле	2
Оперативная память узла	24 GB DDR3
Соединительная сеть	InfniBand QDR (40 Gbit/s)
Операционная система	Linux CentOS

Масштабируемость апекс-метода

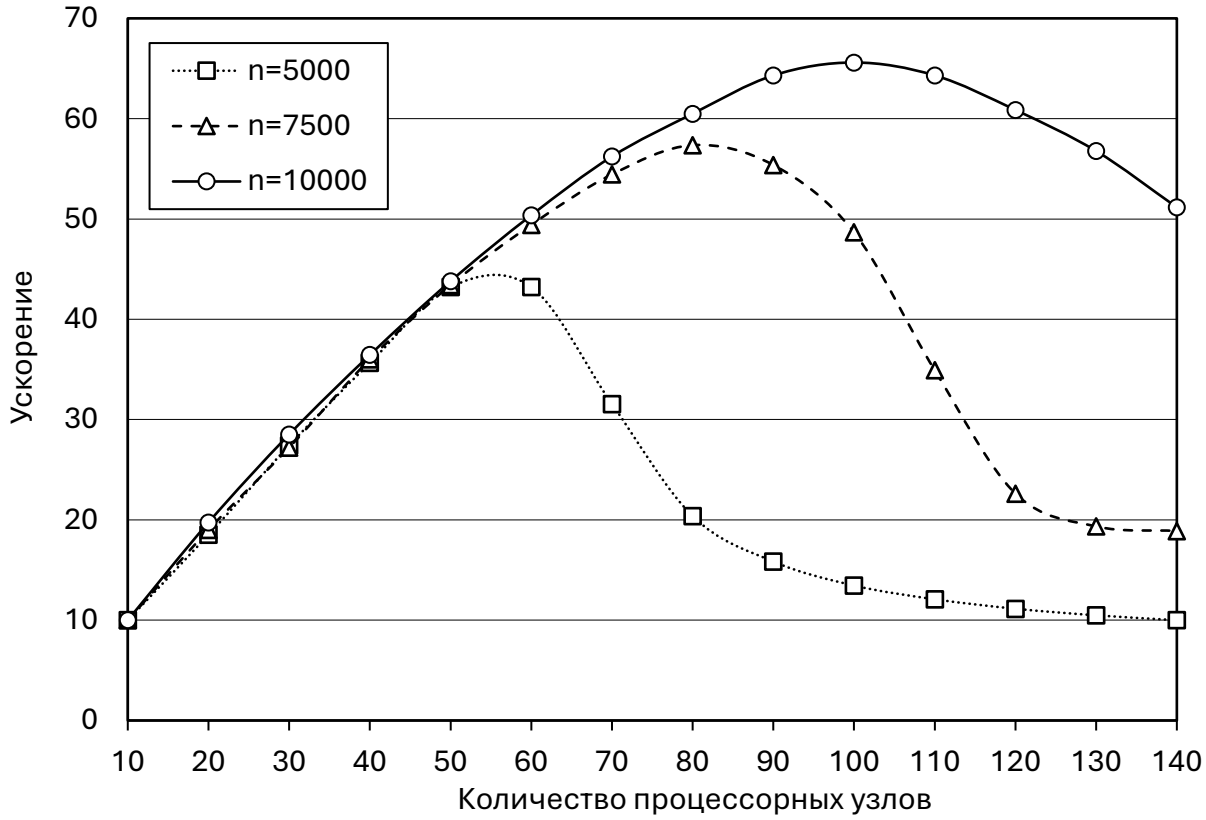
Масштабируемая задача ЛП

$$
\begin{cases}
x_0 & & & & \leq & 200 \\
 & x_1 & & & \leq & 200 \\
 & & \ddots & & \cdots & \cdots \\
 & & & x_{n-1} & \leq & 200 \\
x_0 & + x_1 & \cdots & + x_{n-1} & \leq & 200(n-1)+100 \\
x_0 & + x_1 & \cdots & + x_{n-1} & \geq & 100 \\
x_0 & & & & \geq & 0 \\
 & x_1 & & & \geq & 0 \\
 & & \ddots & & \cdots & \cdots \\
 & & & x_{n-1} & \geq & 0
\end{cases}
$$

$$c = (10n, 10(n-1), 10(n-2), \ldots, 10)$$

$$\bar{x} = (200, 200, \ldots, 200, 100)$$

Графики ускорения

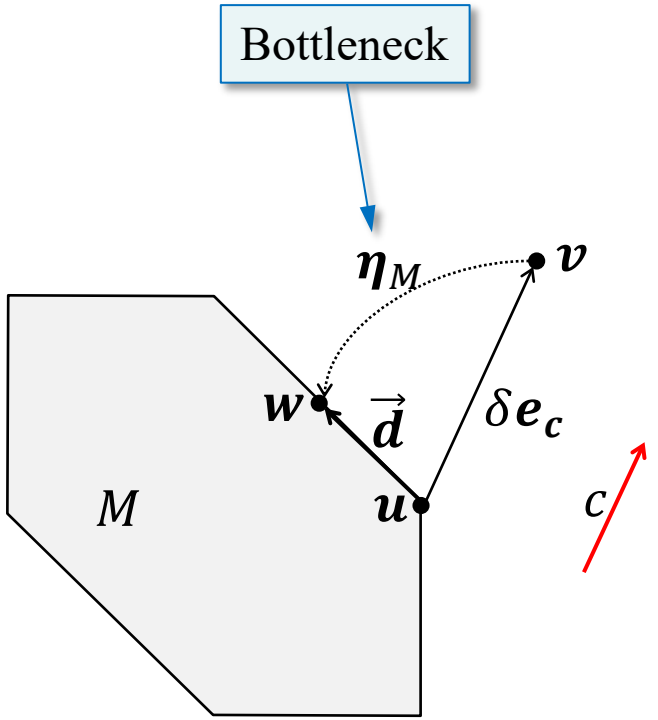


Задачи из NetLib-LP

No	Задача			Апекс-метод	
	Имя	Строк	Столбцов	Значение f(x)	Относительная ошибка
1	ADLITTLE	56	138	-225713.24	9.68E-04
2	AFIRO	27	51	464.75	8.61E-09
3	BLEND	74	114	30.811018	3.19E-05
4	FIT1D	24	1 049	9146.37	8.77E-07
5	KB2	43	68	1687.9152	3.54E-02
6	RECIPE	91	204	266.60404	2.23E-05
7	SC50A	50	78	64.568167	1.06E-04
8	SC50B	50	78	69.990792	1.32E-04
9	SC105	105	163	51.838	6.97E-03
10	SHARE2B	96	102	415.72001	2.40E-05

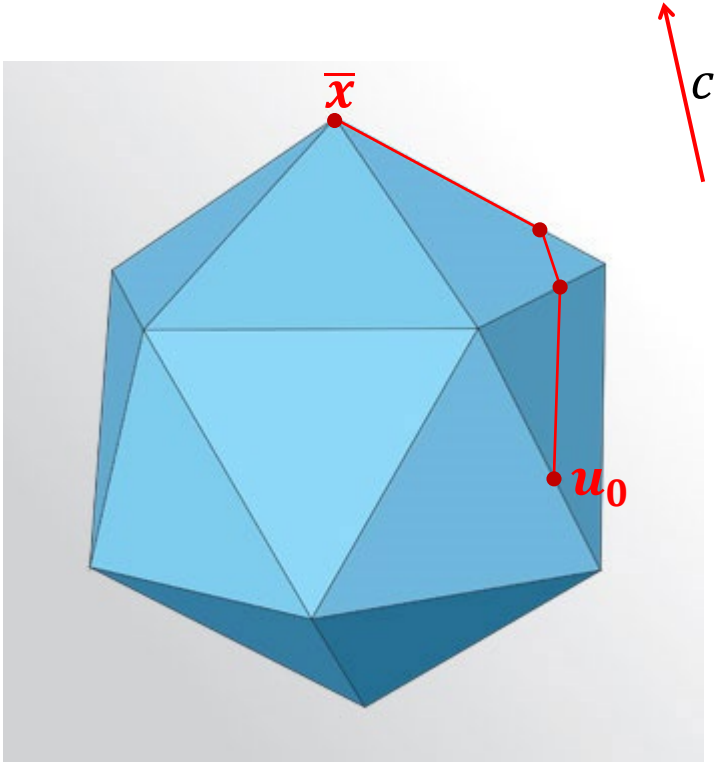
Медленная скорость сходимости процесса вычисления псевдопроекции

- Вычисление одного приближения $\boldsymbol{u}_i$ для задачи с $n = 10\,000, m = 20\,000$ заняло 44 минуты на 100 процессорных узлах
- 99% времени тратилось на вычисление псевдопроекций
- Скорость сходимости сильно зависит от угла между гиперплоскостями: при уменьшении угла скорость сходимости стремится к нулю

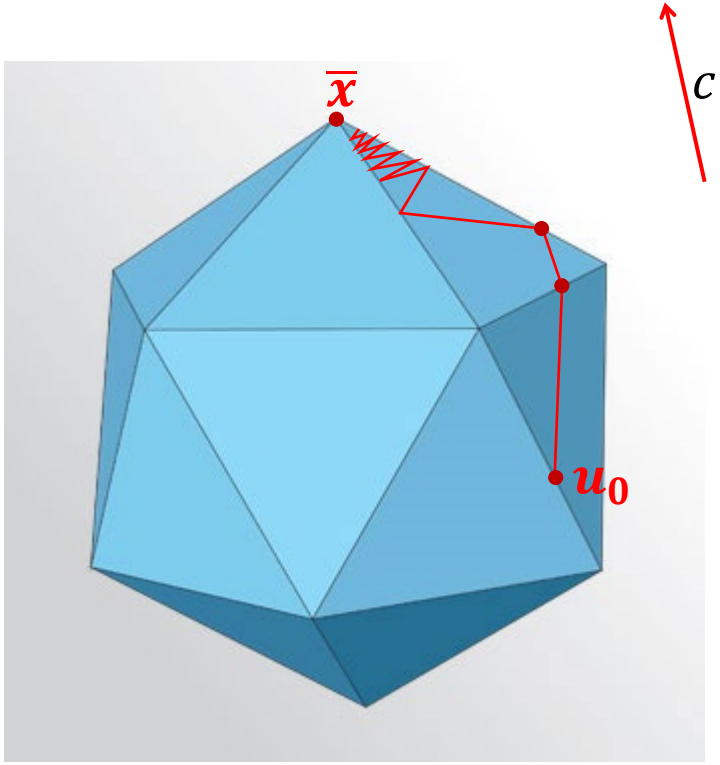


Нестандартное поведение вблизи решения

Метрическая проекция



Апекс-метод



Метод движения по граням SMM

- $I_{\boldsymbol{u}}$ – множество индексов всех гиперплоскостей H_i, проходящих через граничную точку $\boldsymbol{u}$
- $\Im_{\boldsymbol{u}}$ – множество всех собственных подмножеств множества $I_{\boldsymbol{u}}$
- Для каждого $\mathcal{I} \in \Im_{\boldsymbol{u}}$ строим линейное многообразие

$$L \coloneqq \bigcap_{i \in \mathcal{I}} H_i$$

- Строим псевдопроекцию на L и получаем единичный вектор $\vec{\boldsymbol{d}}_L$
- Если $\vec{\boldsymbol{d}}_L$ выходит за пределы допустимого многогранника, переходим к следующему линейному многообразию
- В итоге выбираем $\vec{\boldsymbol{d}}_L$, дающий максимальное приращение целевой функции

Алгоритм вычисления $\vec{\boldsymbol{d}}$

1. $\boldsymbol{e_c} \coloneqq \frac{\boldsymbol{c}}{\|\boldsymbol{c}\|}$
2. $\vec{\boldsymbol{d}} = 0$
3. $f = -\infty$
4. **for** $\mathcal{I} \in \Im_{\boldsymbol{u}}$ **do**
5. $\quad L \coloneqq \bigcap_{i \in \mathcal{I}} H_i$
6. $\quad \boldsymbol{v} \coloneqq \boldsymbol{u} + \delta \boldsymbol{e_c}$
7. $\quad \boldsymbol{w} \coloneqq \boldsymbol{\eta}_L(\boldsymbol{v})$
8. $\quad \boldsymbol{d} \coloneqq \boldsymbol{w} - \boldsymbol{u}$
9. $\quad \boldsymbol{e_d} \coloneqq \frac{\boldsymbol{d}}{\|\boldsymbol{d}\|}$
10. $\quad \boldsymbol{u}' \coloneqq \boldsymbol{u} + \varepsilon \boldsymbol{e_d}$
11. $\quad$ **if** $\boldsymbol{u}' \notin M$ **then**
12. $\quad\quad$ **continue**
13. $\quad$ **end if**
14. $\quad$ **if** $\langle \boldsymbol{c}, \boldsymbol{u}' \rangle > f$ **then**
15. $\quad\quad \vec{\boldsymbol{d}} = \boldsymbol{d}$
16. $\quad\quad f = \langle \boldsymbol{c}, \boldsymbol{u}' \rangle$
17. $\quad$ **end if**
18. **end for**

Почему это работает

Утверждение 3. Псевдопроекция на линейное многообразие совпадает с ортогональной проекцией.

Параллельная реализация SMM

- Использован параллельный программный каркас BSF: https://github.com/leonid-sokolinsky/BSF-skeleton
  - C++
  - MPI
- Исходные коды: https://github.com/leonid-sokolinsky/BSF-Surface-movement-method

Характеристики вычислительного кластера «Торнадо ЮУрГУ»

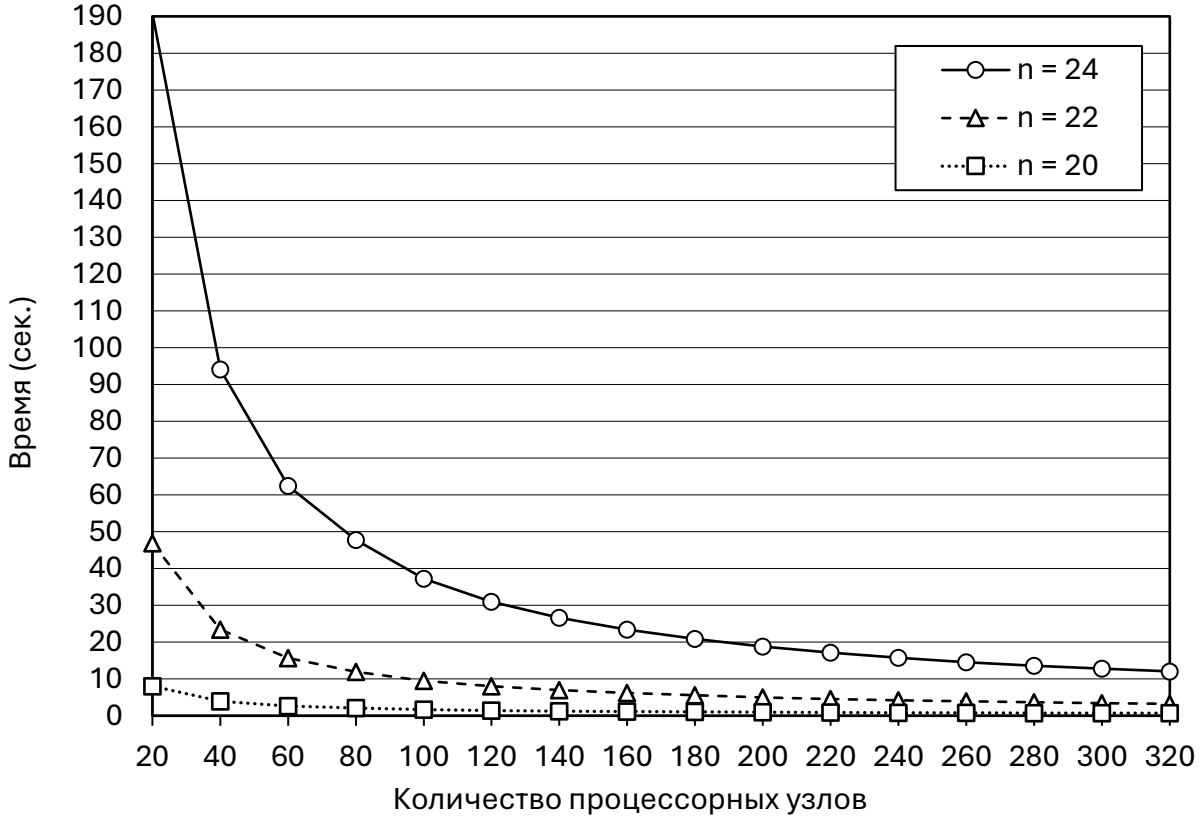
Количество процессорных узлов	480
Процессоры	Intel Xeon X5680 (6 cores 3.33 GHz)
Количество процессоров в узле	2
Оперативная память узла	24 GB DDR3
Соединительная сеть	InfniBand QDR (40 Gbit/s)
Операционная система	Linux CentOS

Тестовый набор задач

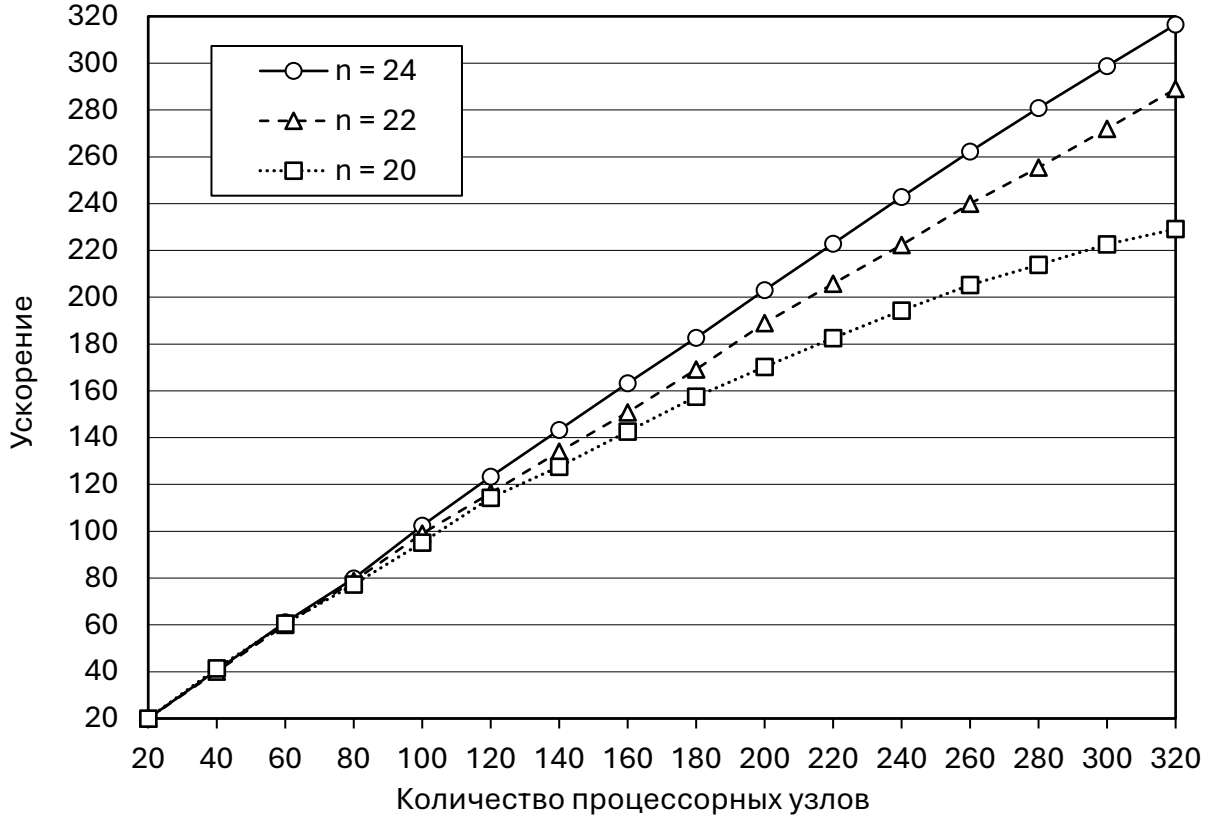
- Генератор случайных задач ЛП https://github.com/leonid-sokolinsky/BSF-LPP-Generator
- Набор сгенерированных задач: https://github.com/leonid-sokolinsky/Random-LP-Problems

Масштабируемость метода SMM

Время



Ускорение



Проблема комбинаторного взрыва

- Количество подмножеств множества из K элементов равно 2^K
- Если через точку проходит 30 гиперплоскостей, необходимо проверить $2^{30} - 2 = 1\,073\,741\,822$ линейных многообразий
- Если через точку проходит 40 гиперплоскостей, необходимо проверить $2^{40} - 2 = 1\,099\,511\,627\,774$ линейных многообразий

Метод движения по ребрам ЕММ

- $I_{\boldsymbol{u}}$ – множество индексов всех гиперплоскостей H_i, проходящих через точку $\boldsymbol{u}$, являющуюся вершиной допустимого многогранника M: $|I_{\boldsymbol{u}}| \geq n$
- $\mathfrak{E}_{\boldsymbol{u}}$ – множество сочетаний $(n-1)$ элементов из $k = |I_{\boldsymbol{u}}|$ элементов всех собственных подмножеств множества $I_{\boldsymbol{u}}$
- Для всех $\mathcal{J} \in \mathfrak{E}_{\boldsymbol{u}}$ строим прямую

$$L \coloneqq \bigcap_{i \in \mathcal{J}} H_i$$

- Строим псевдопроекцию на L и получаем единичный вектор $\vec{\boldsymbol{d}}_L$
- Если $\vec{\boldsymbol{d}}_L$ выходит за пределы допустимого многогранника, переходим к следующей прямой
- Выбираем $\vec{\boldsymbol{d}}_L$, дающий максимальное приращение целевой функции

Алгоритм вычисления $\vec{d}$

1. $\boldsymbol{e_c} \coloneqq \frac{\boldsymbol{c}}{\|\boldsymbol{c}\|}$
2. $\vec{\boldsymbol{d}} = 0$
3. $f = -\infty$
4. **for** $\mathcal{J} \in \mathfrak{E}_{\boldsymbol{u}}$ **do**
5. $\quad L \coloneqq \bigcap_{i \in \mathcal{J}} H_i$
6. $\quad \boldsymbol{v} \coloneqq \boldsymbol{u} + \delta \boldsymbol{e_c}$
7. $\quad \boldsymbol{w} \coloneqq \boldsymbol{\eta}_L(\boldsymbol{v})$
8. $\quad \boldsymbol{d} \coloneqq \boldsymbol{w} - \boldsymbol{u}$
9. $\quad \boldsymbol{e_d} \coloneqq \frac{\boldsymbol{d}}{\|\boldsymbol{d}\|}$
10. $\quad \boldsymbol{u}' \coloneqq \boldsymbol{u} + \varepsilon \boldsymbol{e_d}$
11. $\quad$ **if** $\boldsymbol{u}' \notin M$ **then**
12. $\quad\quad$ **continue**
13. $\quad$ **end if**
14. $\quad$ **if** $\langle \boldsymbol{c}, \boldsymbol{u}' \rangle > f$ **then**
15. $\quad\quad \vec{\boldsymbol{d}} = \boldsymbol{d}$
16. $\quad\quad f = \langle \boldsymbol{c}, \boldsymbol{u}' \rangle$
17. $\quad$ **end if**
18. **end for**

Параллельная реализация SMM

- Использован параллельный программный каркас BSF: https://github.com/leonid-sokolinsky/BSF-skeleton
  - C++
  - MPI
- Исходные коды: https://github.com/leonid-sokolinsky/BSF-EMM

Характеристики вычислительного кластера «Торнадо ЮУрГУ»

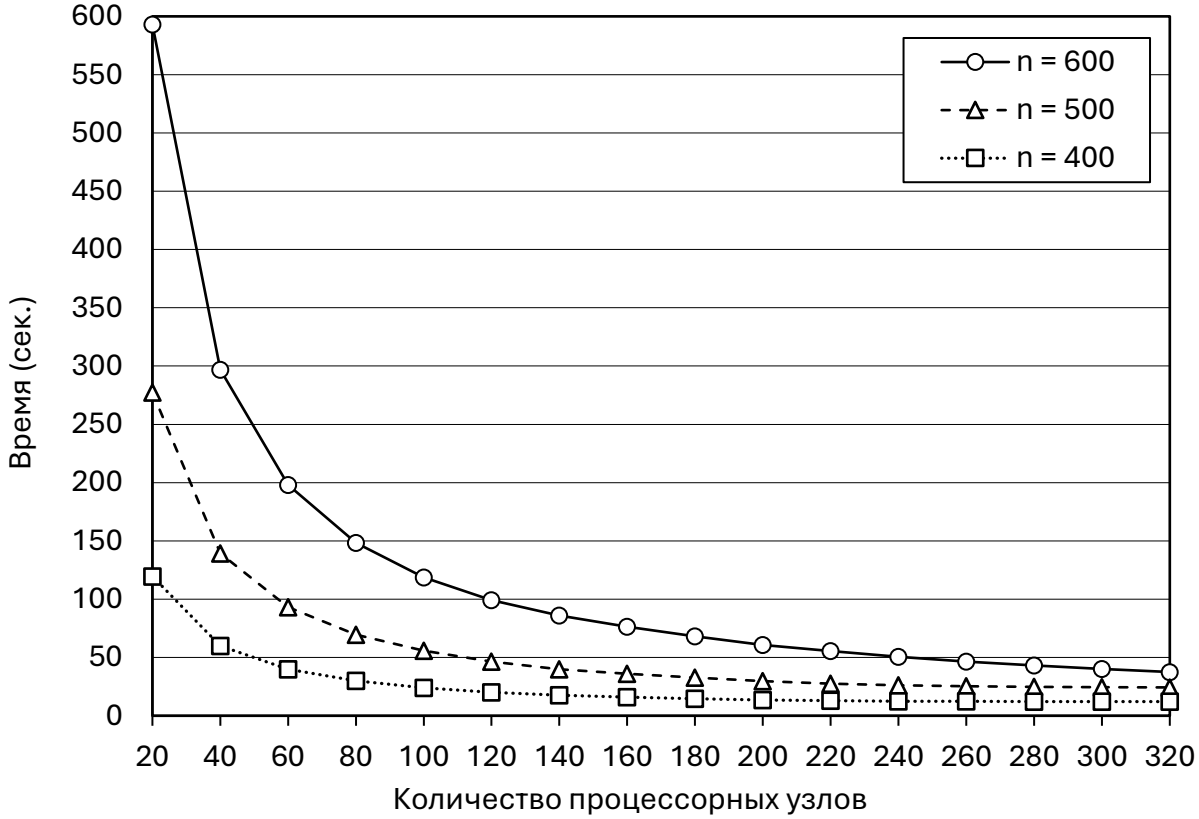
Количество процессорных узлов	480
Процессоры	Intel Xeon X5680 (6 cores 3.33 GHz)
Количество процессоров в узле	2
Оперативная память узла	24 GB DDR3
Соединительная сеть	InfniBand QDR (40 Gbit/s)
Операционная система	Linux CentOS

Тестовый набор задач

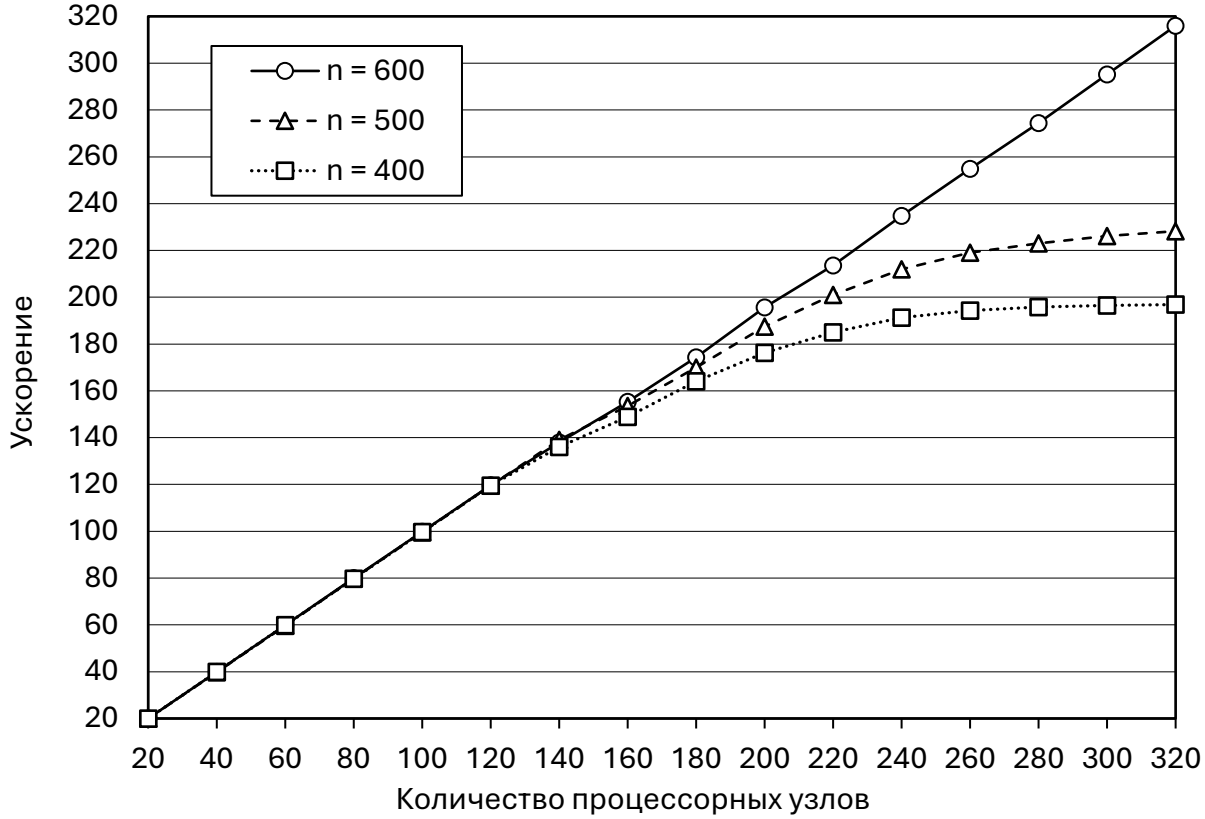
- Генератор случайных задач ЛП https://github.com/leonid-sokolinsky/BSF-LPP-Generator
- Набор сгенерированных задач: https://github.com/leonid-sokolinsky/Random-LP-Problems

Масштабируемость метода ЕММ

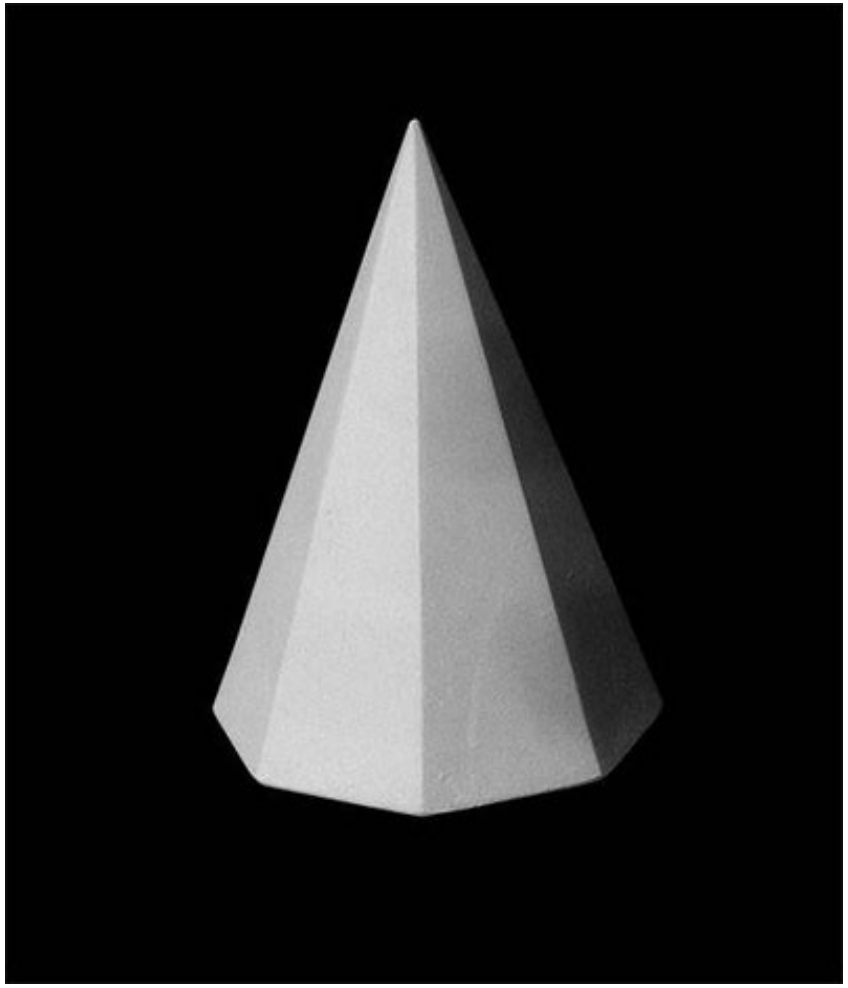
Время



Ускорение



Проблема комбинаторного взрыва



- Через точку проходит $2n$ гиперплоскостей
- Количество ребер при $n = 20$ равно 137 846 528 820
- Количество ребер при $n = 30$ равно 118 264 581 564 861 424

Направления будущих исследований

- Стохастический метод ЕММ
- Использование искусственных нейронных сетей

Спасибо за внимание!