Исследование фазы Quest алгоритма NSLP для решения нестационарных задач линейного программирования на кластерных вычислительных системах

д.ф.-м.н., Л.Б. Соколинский,
к.ф.-м.н., И.М. Соколинская

Южно-Уральский государственный университет
(национальный исследовательский университет)

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 17-07-00352а, Правительства РФ в соответствии с Постановлением №211 от 16.03.2013 г. (соглашение № 02.A03.21.0011) и Министерства образования и науки РФ (государственное задание 2.7905.2017/8.9).

Большие задачи ЛП

- Экономика
- Физика
- Логистика
- Составление расписаний
- Алгоритмическая торговля на биржах (нестационарный характер)

Нестационарная задача линейного программирования

$$\bar{x} = \text{arg max}\left\{\langle c^{(t)}, x\rangle | A^{(t)}x \leq b^{(t)}\right\}$$

- $x \in \mathbb{R}_n$
- $A^{(t)}$ – матрица $m \times n$
- $c^{(t)}, b^{(t)}$ – векторы размерности n
- $t \in \mathbb{R}_{\geq 0}$ – время

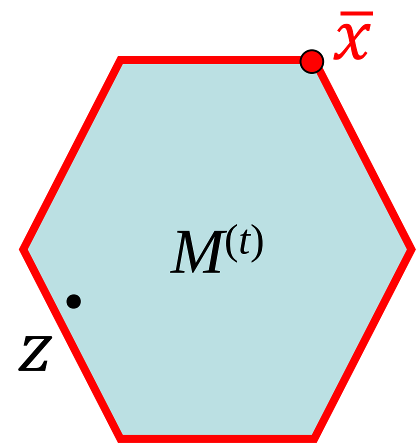
Известные методы

- Симплекс-метод
  – Потеря точности на больших задачах ЛП
  – Плохая масштабируемость (≤ 32 процессорных узлов)
  – Не работает на нестационарных задачах ЛП
- Метод внутренних точек
  + Самокорректируемость
  – Плохо распараллеливается

Идея алгоритма NSLP (Non-Stationary Linear Programming)

Фазы алгоритма:

- *Quest* – поиск точки $z \in M^{(t)}$
- *Targeting* – перемещение точки $\bar{z}$ таким образом, чтобы точное решение $\bar{x}$ задачи ЛП находилось в ее ε-окрестности

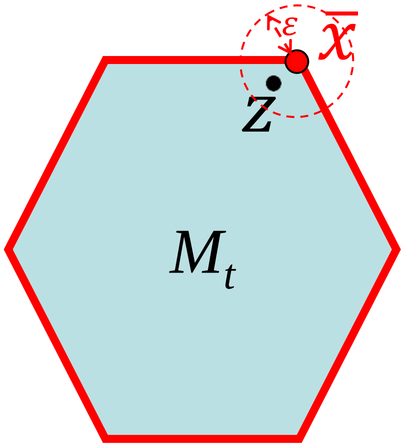


$$A^{(t)}x \leq b^{(t)} \Leftrightarrow x \in M^{(t)}$$

Идея алгоритма NSLP (Non Stationary Linear Programming)

Фазы алгоритма:

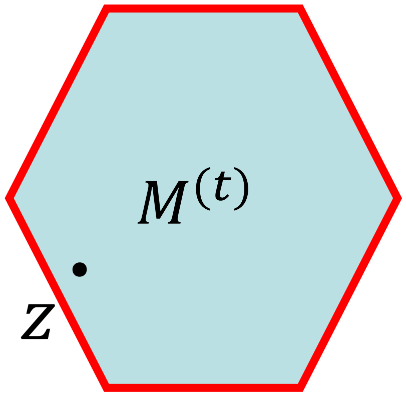
- *Quest* – поиск точки $z \in M^{(t)}$
- *Targeting* – перемещение точки z таким образом, чтобы точное решение $\bar{x}$ задачи ЛП находилось в ее ε-окрестности



$$A^{(t)}x \leq b^{(t)} \Leftrightarrow x \in M^{(t)}$$

Фаза *Quest* (поиск точки $\bar{z} \in M_t$)

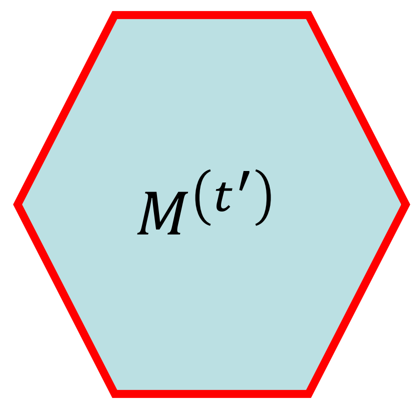
Нельзя просто решить систему неравенств $A^{(t)}x \le b^{(t)}$, так как пока мы ее решаем, многогранник M_t поменяет положение в пространстве.



$$A^{(t)}x \le b^{(t)} \Leftrightarrow x \in M^{(t)}$$

Фаза *Quest*
(поиск точки $\bar{z} \in M_t$)

Нельзя просто решить систему неравенств $A^{(t)}x \le b^{(t)}$, так как пока мы ее решаем, многогранник M_t поменяет положение в пространстве.



$A^{(t')}x \le b^{(t')} \Leftrightarrow x \in M^{(t')}$

z

$A^{(t)}x \le b^{(t)} \Leftrightarrow x \in M^{(t)}$

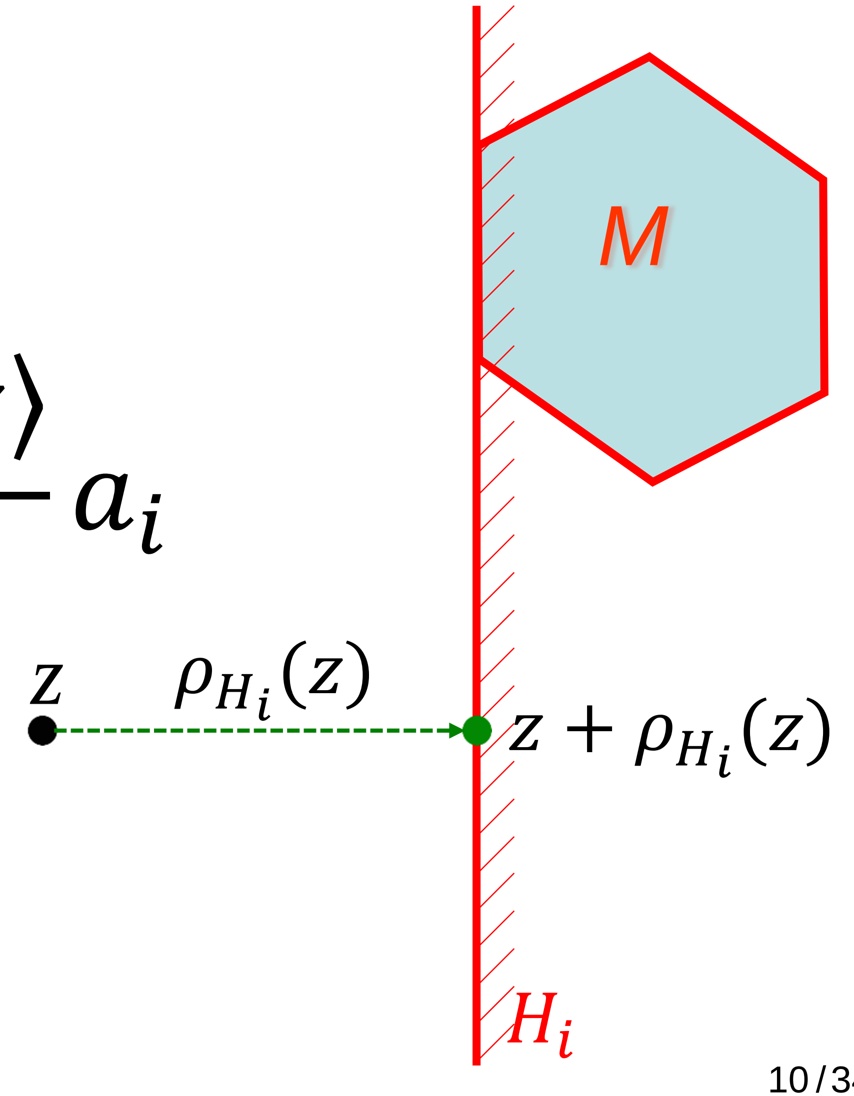
Алгоритм для фазы Quest

- Требования:
  - Высокая масштабируемость
  - Самокорректируемость
- Алгоритм Чиммино для неравенств
  - Проекционный
  - Итерационный

Вектор проекции на гиперплоскость H_i

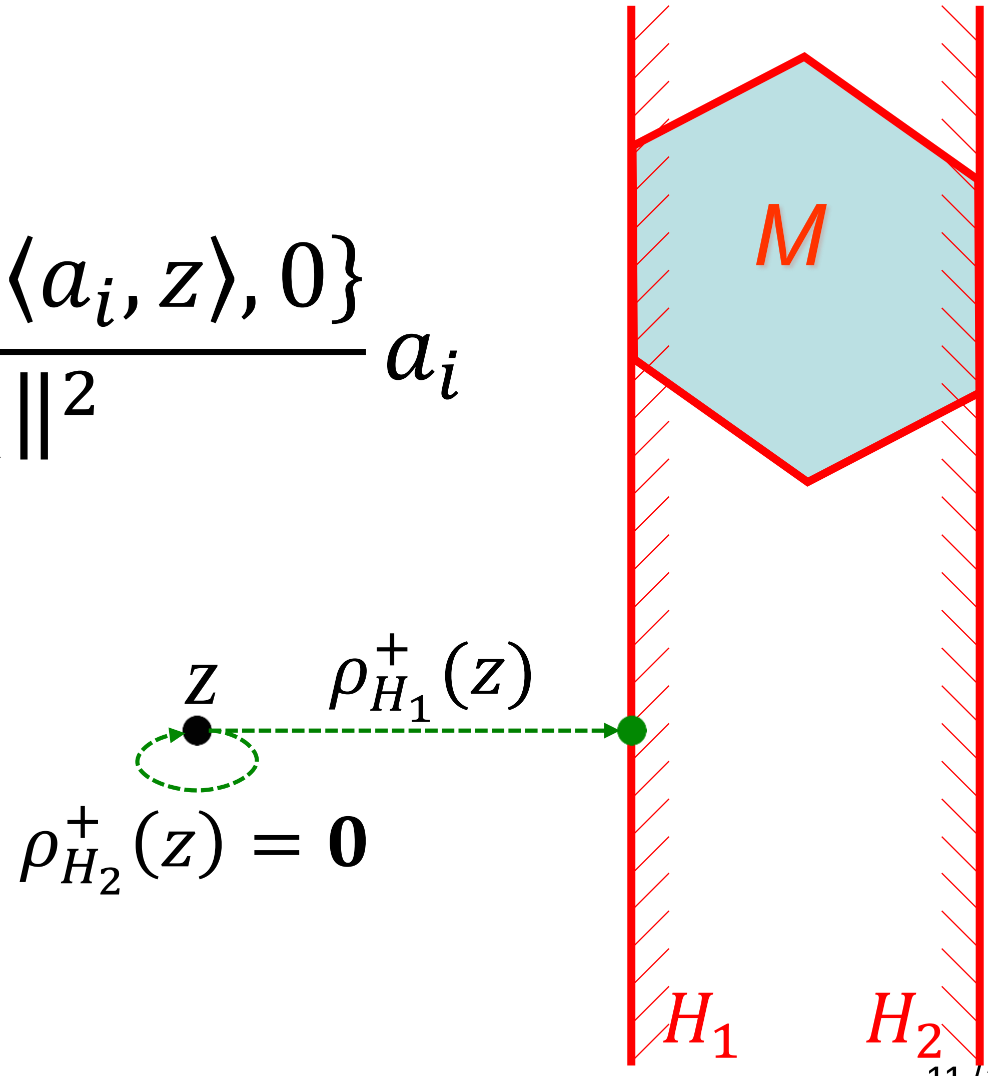
$$H_i: \langle a_i, x \rangle = b_i$$

$$\rho_{H_i}(z) = \frac{b_i - \langle a_i, z \rangle}{\|a_i\|^2} a_i$$



Положительная срезка вектора проекции на гиперплоскость H_i

$$\rho^+_{H_i}(z) = \frac{\min\{b_i - \langle a_i, z\rangle, 0\}}{\|a_i\|^2} a_i$$



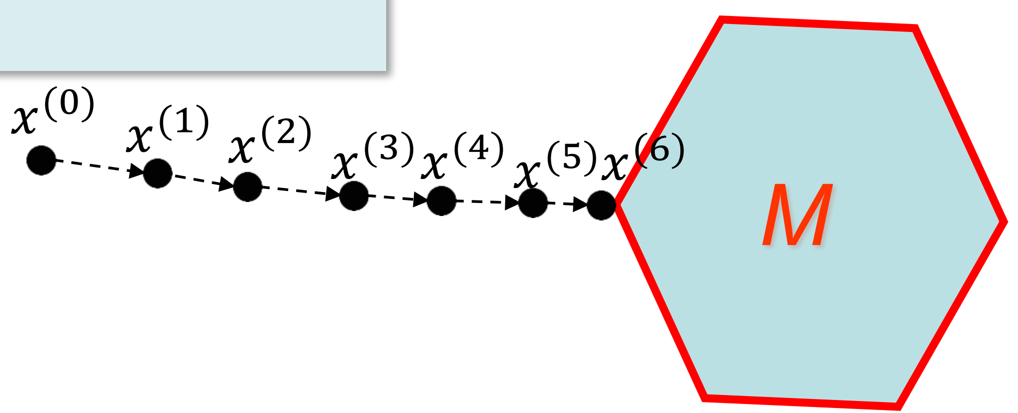
Проекционное отображение

$$\varphi(x) = \frac{1}{h}\sum_{i=1}^{m}\rho_{H_i}^{+}(x)$$

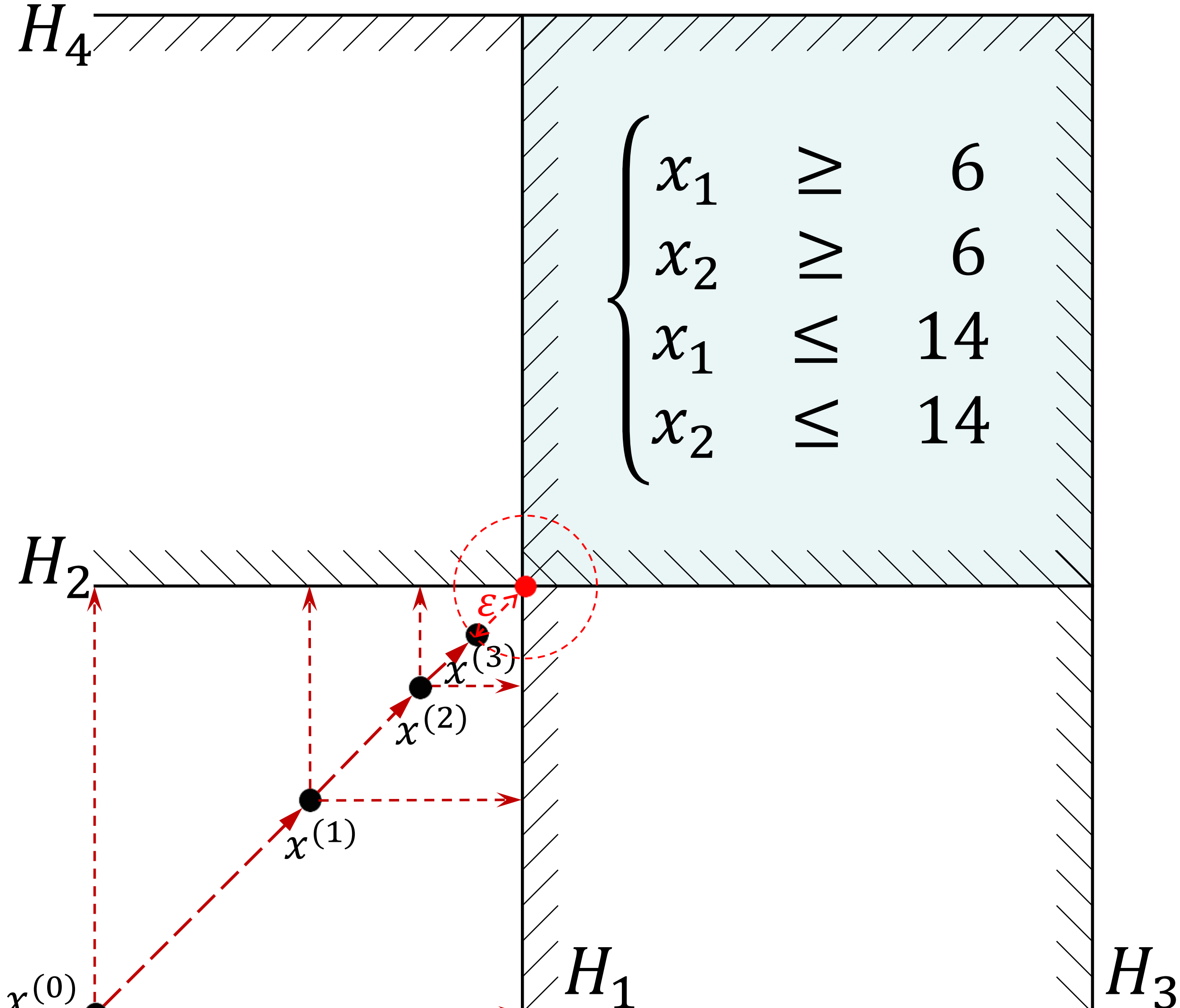
h – количество ненулевых слагаемых в сумме $\sum_{i=1}^{m}\rho_{H_i}^{+}(x)$

Итерационный алгоритм Чиммино для неравенств

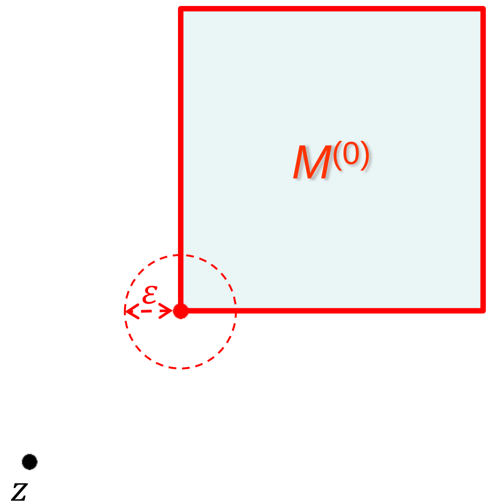
1. $x^{(0)} := \mathbf{0}$
2. $k := 0$
3. $x^{(k+1)} := x^{(k)} + \varphi^{(k)}\left(x^{(k)}\right)$
4. **if** $\left\|x^{(k+1)} - x^{(k)}\right\|^2 < \varepsilon$ **goto** 7
5. $k := k + 1$
6. **goto** 3
7. **stop**



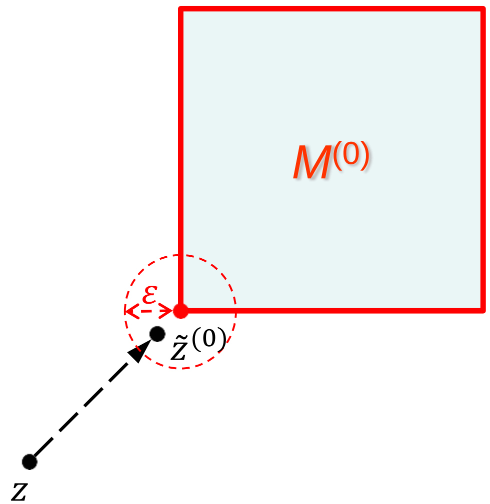
Как работает алгоритм



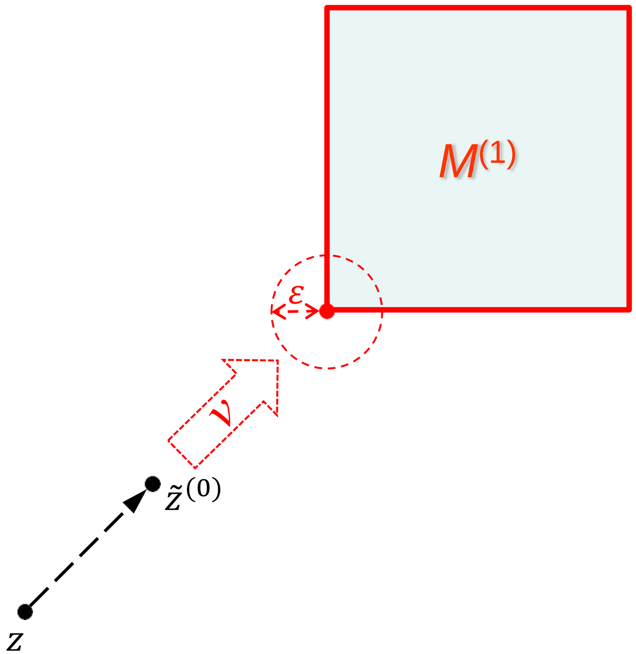
Алгоритм Чиммино для неравенств не работает в нестационарном случае



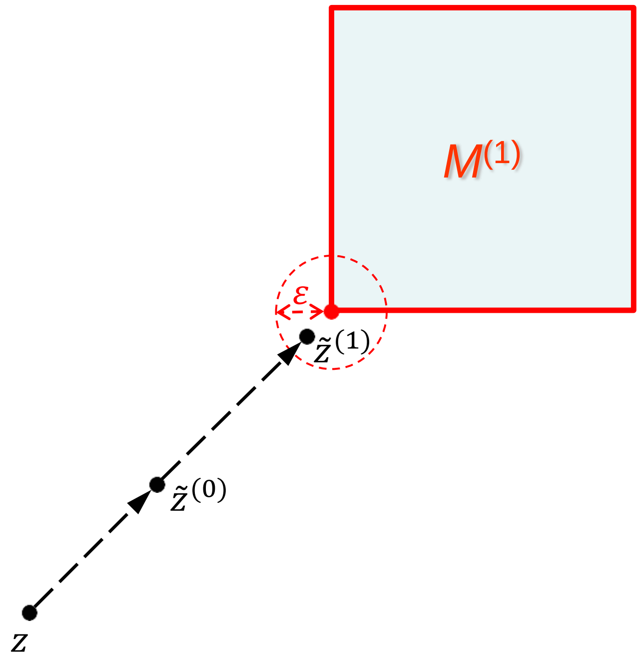
Алгоритм Чиммино для неравенств не работает в нестационарном случае



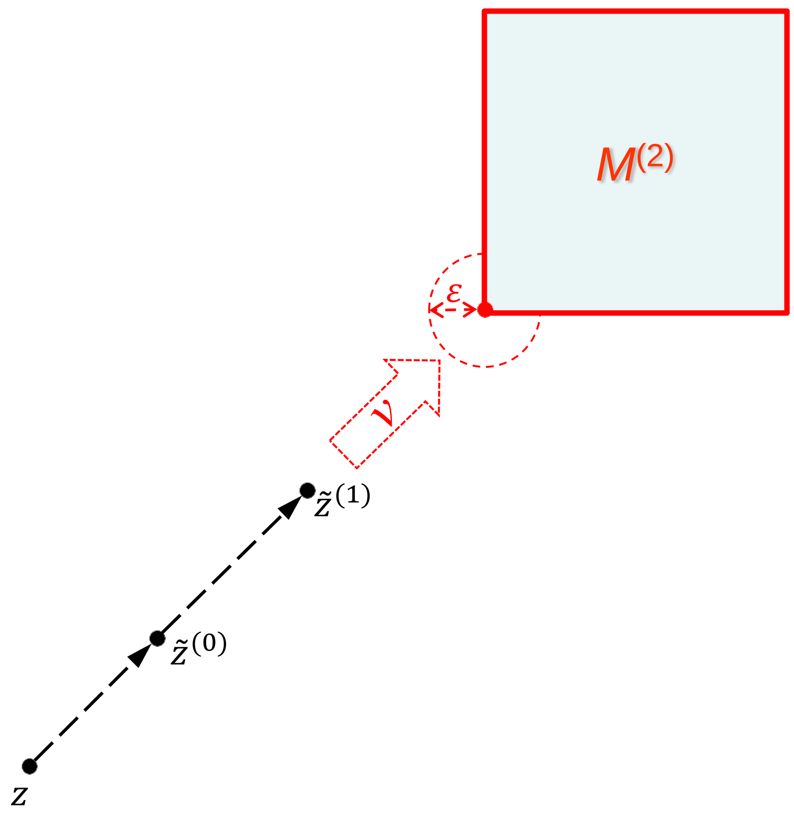
Алгоритм Чиммино для неравенств не работает в нестационарном случае



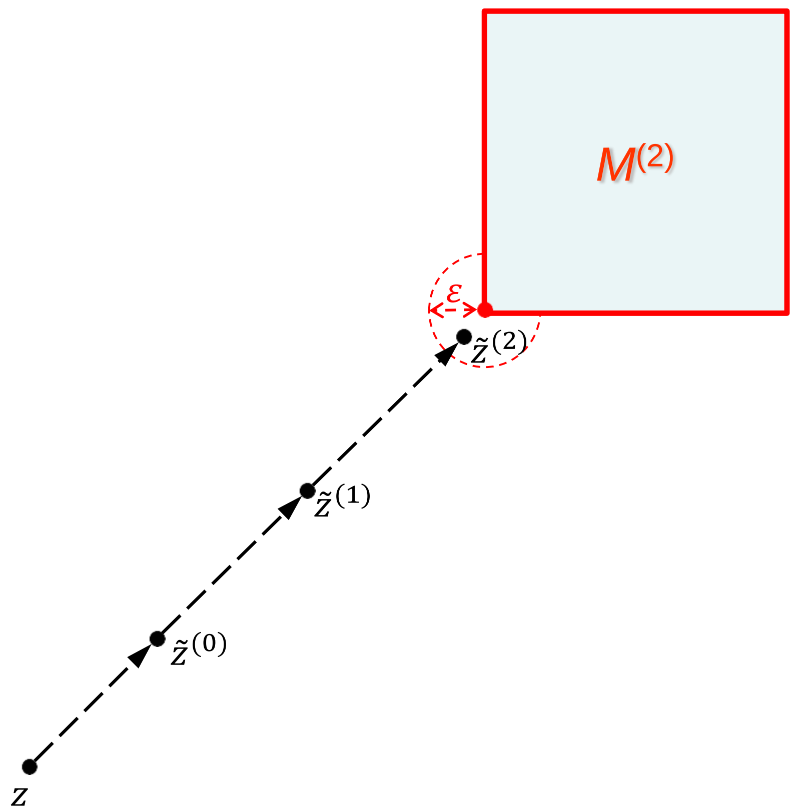
Алгоритм Чиммино для неравенств не работает в нестационарном случае



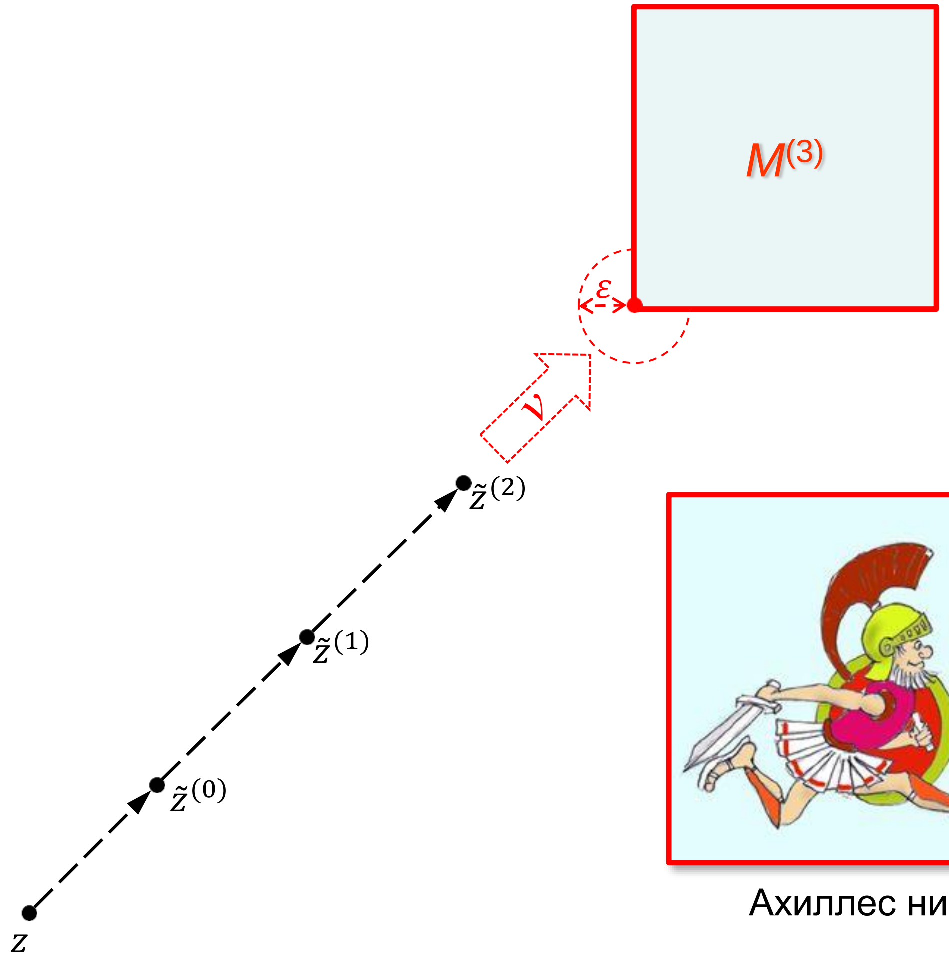
Алгоритм Чиммино для неравенств не работает в нестационарном случае

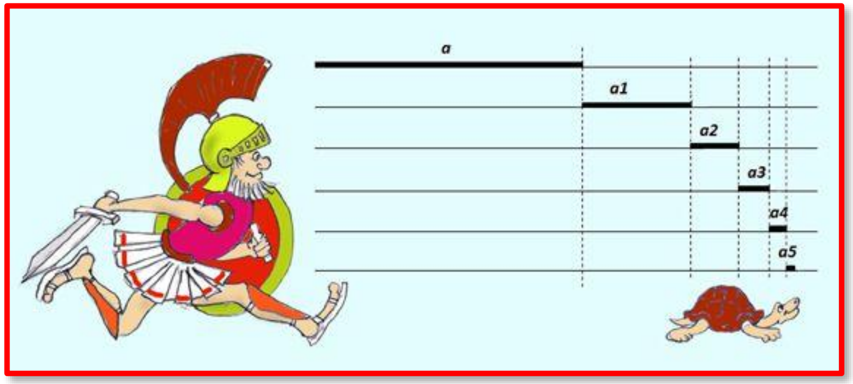


Алгоритм Чиммино для неравенств не работает в нестационарном случае



Алгоритм Чиммино для неравенств не работает в нестационарном случае





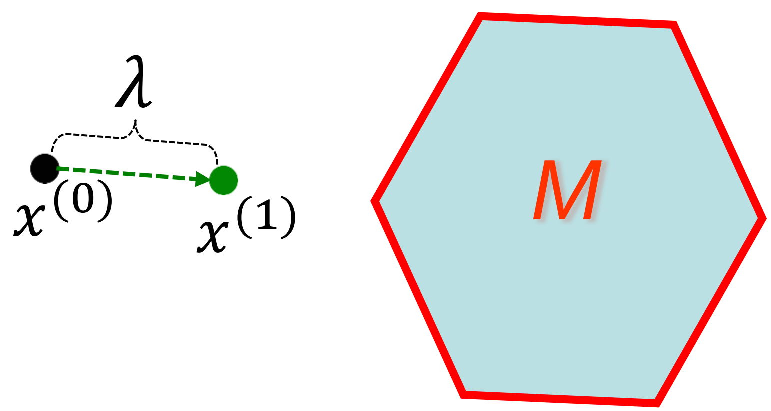
Ахиллес никогда не догонит черепаху!

Модификация проекционного отображения

$$\varphi(x) = \frac{1}{h}\sum_{i=1}^{m}\rho_{H_i}^{+}(x) \quad \Rightarrow \quad \psi(x) = \lambda\frac{\varphi(x)}{\|\varphi(x)\|}$$

$$\lambda > 0$$

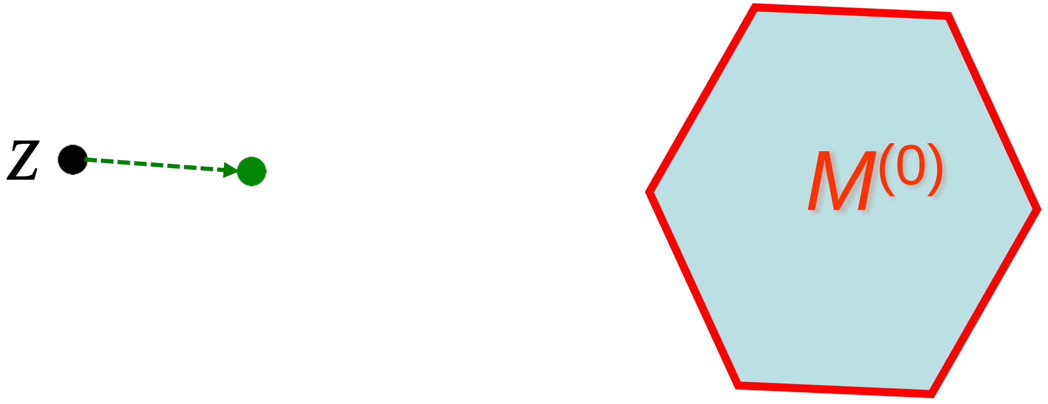
$$x^{(1)} = x^{(0)} + \psi\left(x^{(0)}\right)$$



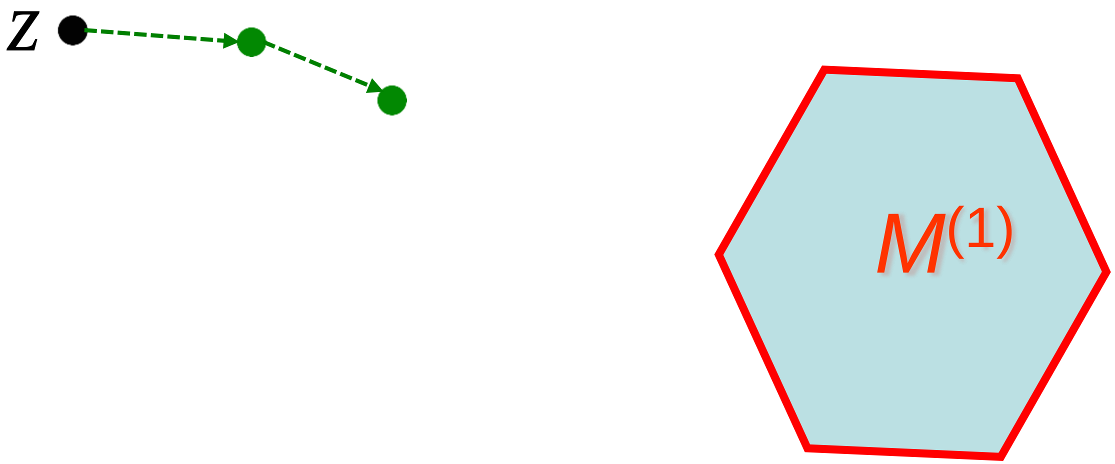
Модифицированный алгоритм

1. $x^{(0)} := \mathbf{0}$
2. $k := 0$
3. $x^{(k+1)} := x^{(k)} + \psi^{(k)}\left(x^{(k)}\right)$
4. **if** $x^{(k+1)} \in M^{(k)}$ **goto** 7
5. $k := k + 1$
6. **goto** 3
7. **stop**

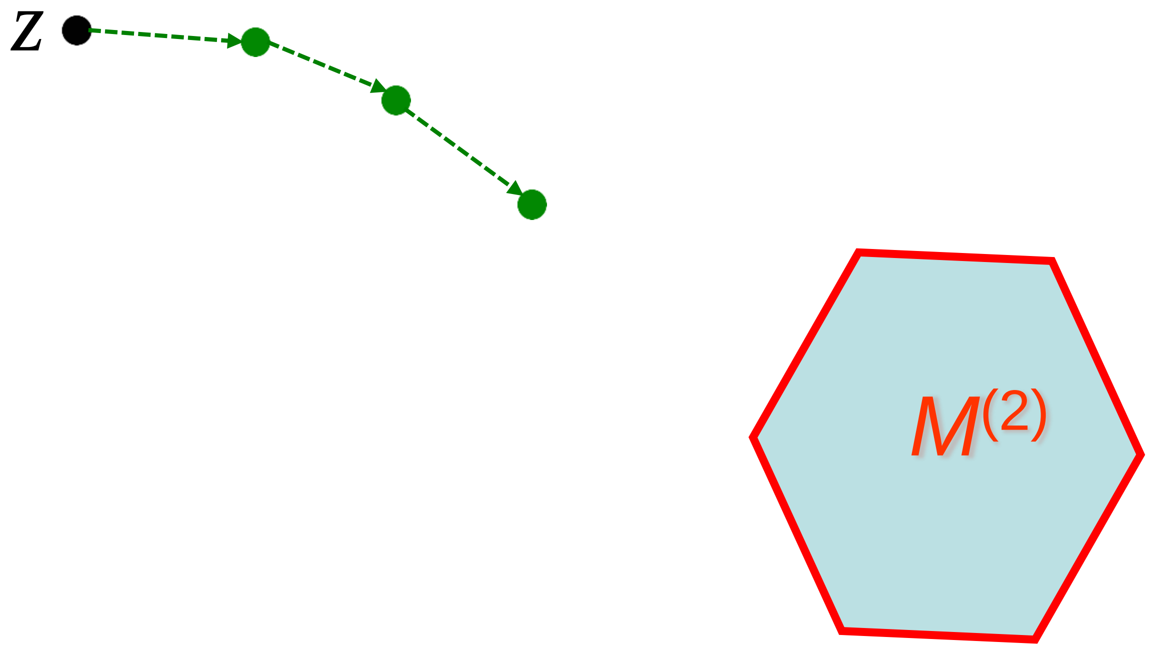
Работа модифицированного алгоритма



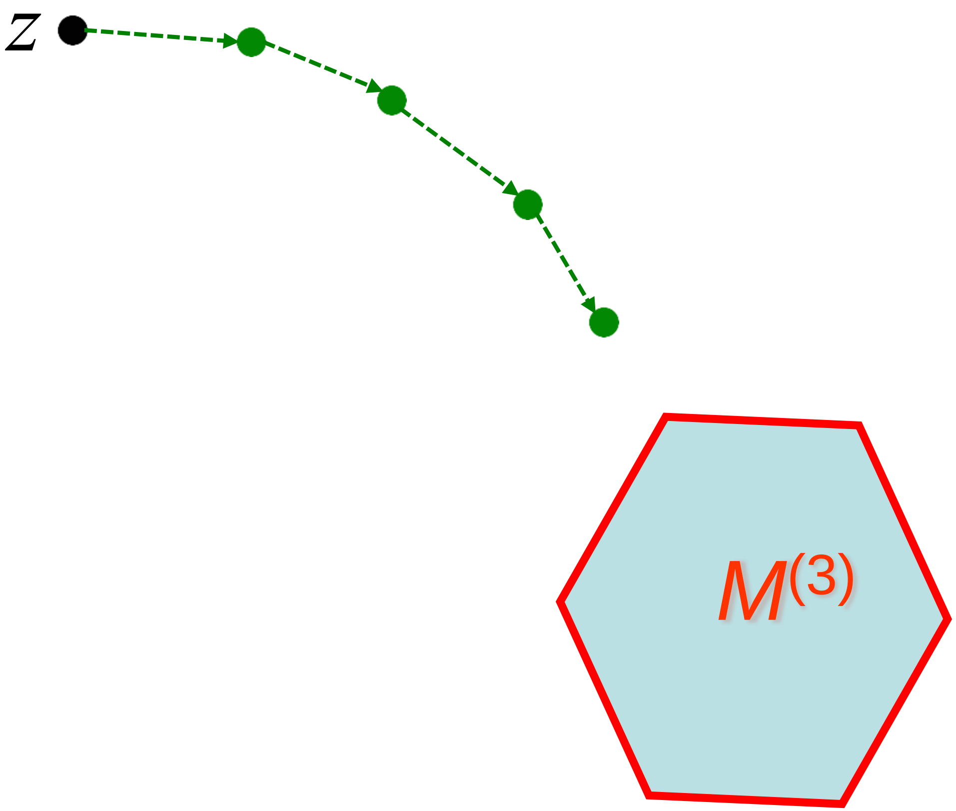
Работа модифицированного алгоритма



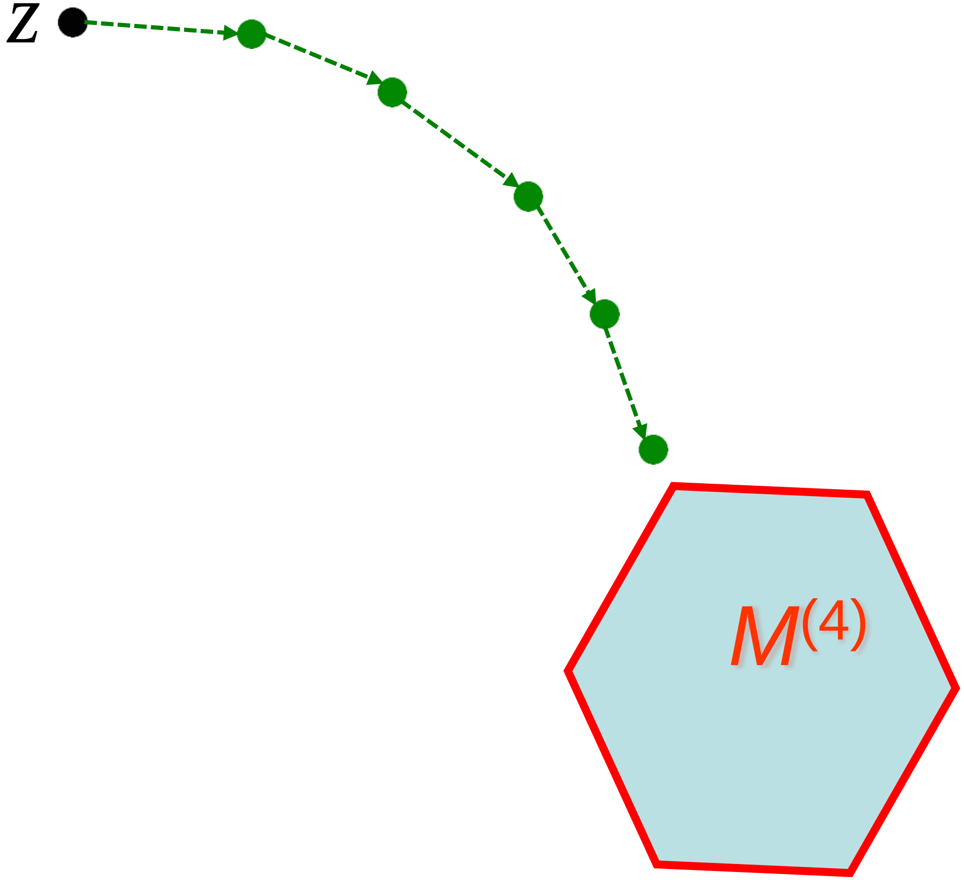
Работа модифицированного алгоритма



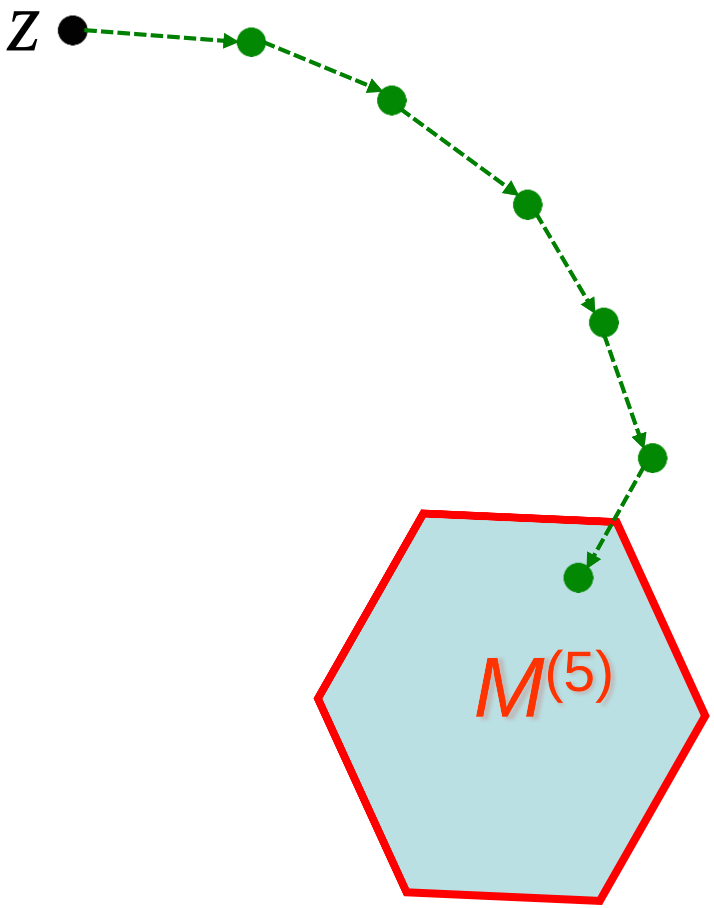
Работа модифицированного алгоритма



Работа модифицированного алгоритма



Работа модифицированного алгоритма



Ахиллес догнал черепаху!

Тестовый пример

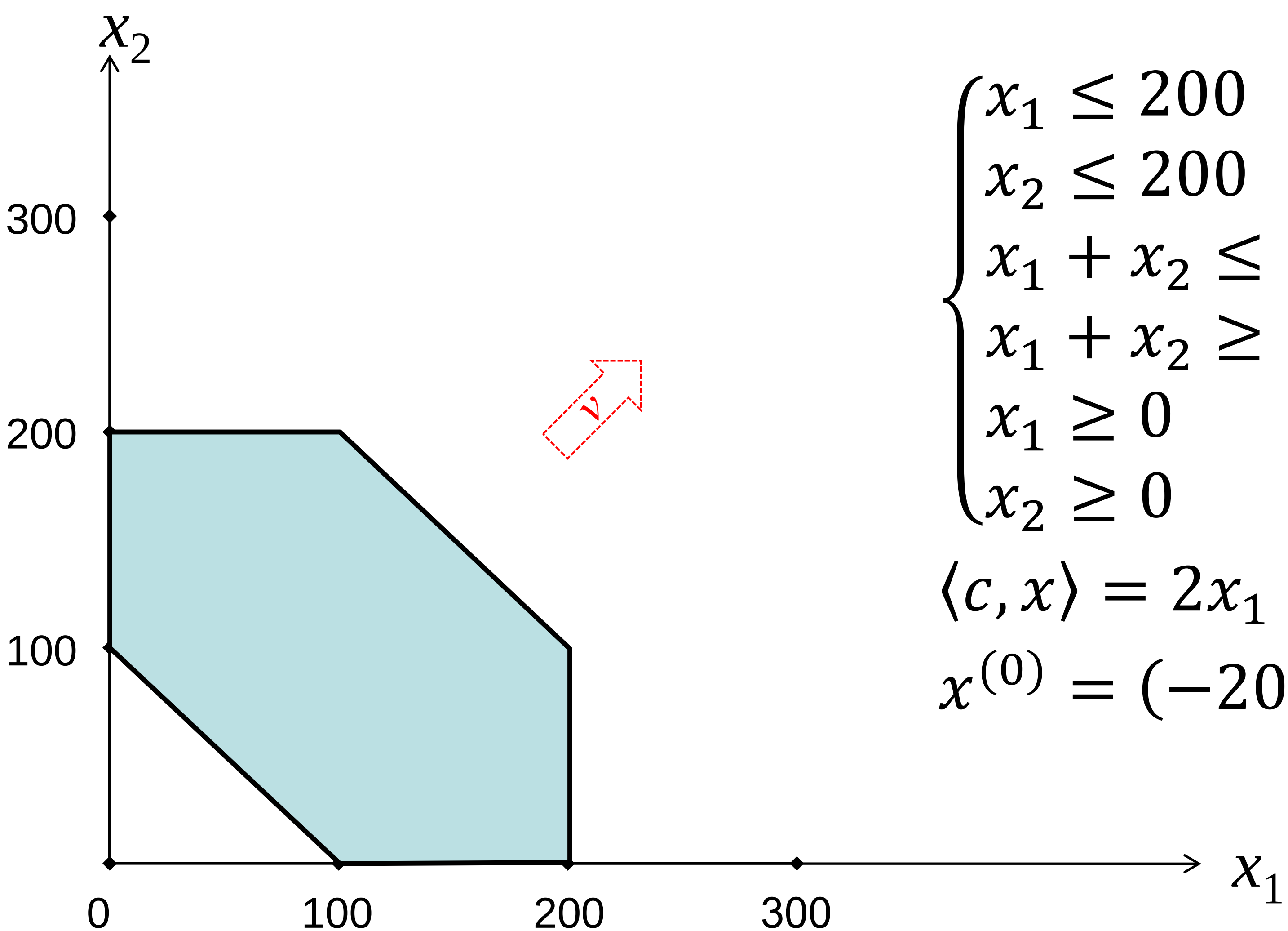
$$
\begin{cases}
x_0 & & & & \le & 200 \\
& x_1 & & & \le & 200 \\
& & \ddots & & \cdots & \cdots \\
& & & x_{n-1} & \le & 200 \\
x_0 & + \; x_1 & \cdots & + \; x_{n-1} & \le & 200(n-1)+100 \\
x_0 & + \; x_1 & \cdots & + \; x_{n-1} & \le & -100 \\
-x_0 & & & & \le & 0 \\
& -x_1 & & & \le & 0 \\
& & \ddots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

$$C_{\max}(x) = nx_0 + (n-1)x_1 + \cdots + 2x_{n-2} + x_{n-1}$$

Количество переменных: n

Количество неравенств: $m = 2n + 2$

Тестовый пример при n=2



$$\begin{cases} x_1 \le 200 \\ x_2 \le 200 \\ x_1 + x_2 \le 300 \\ x_1 + x_2 \ge 100 \\ x_1 \ge 0 \\ x_2 \ge 0 \end{cases}$$

$$\langle c, x \rangle = 2x_1 + 1x_2$$

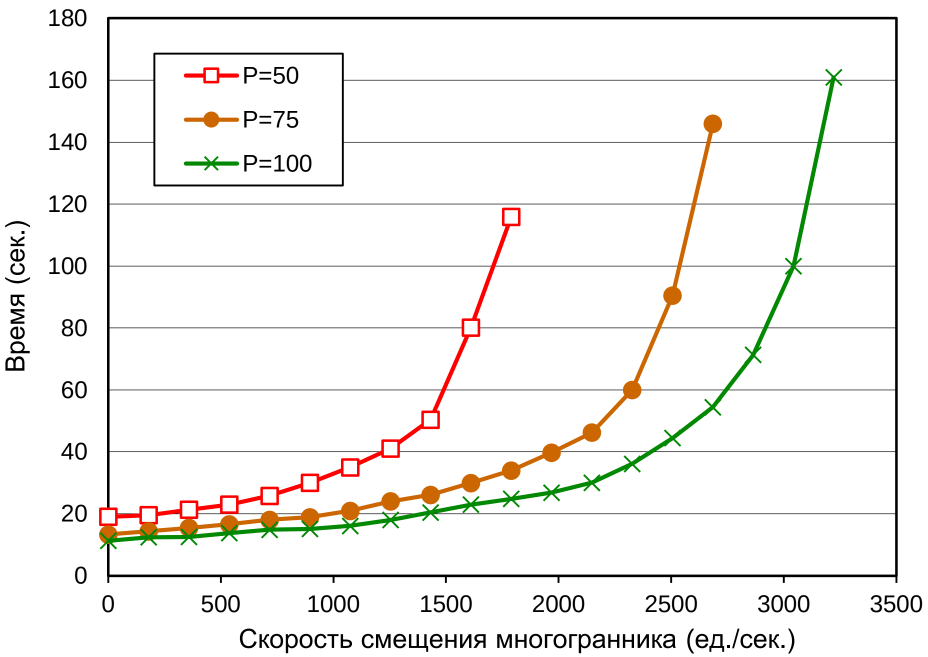
$$x^{(0)} = (-200, -200)$$

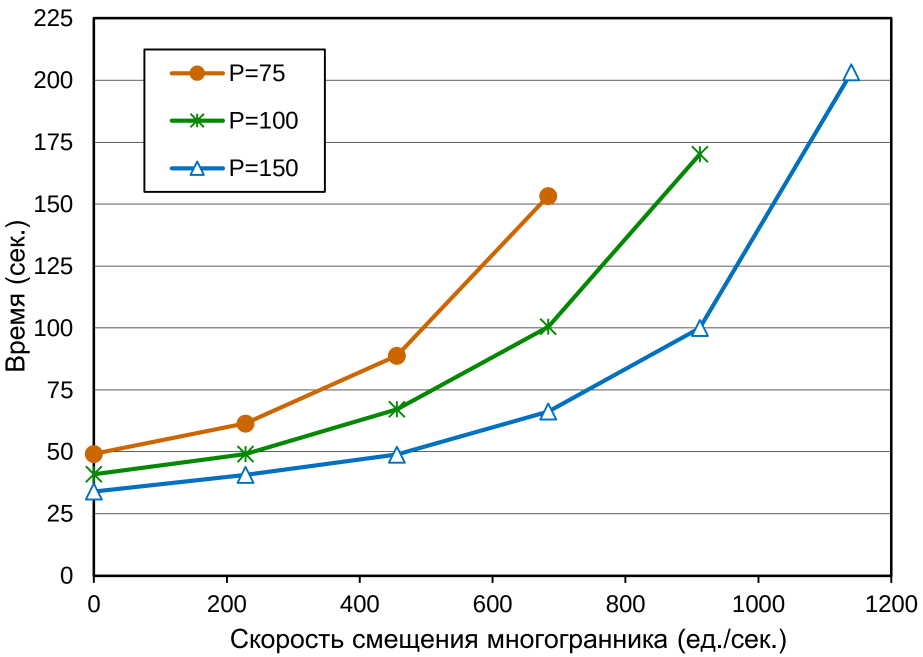
Суперкомпьютер «Торнадо ЮУрГУ»

Количество узлов:	384
Тип процессоров:	2 x Intel Xeon X5680 (12 ядер по 3.33 ГГц; 2 потока на ядро)
Оперативная память узла:	24 Гб
Тип сопроцессора:	Intel Xeon Phi SE10X: (61 ядро по 1.1 ГГц; 4 потока на ядро)
Память сопроцессора:	8 Гб
Тип системной сети:	InfiniBand QDR
Тип управляющей сети:	Gigabit Ethernet
Операционная система:	Linux CentOS 6.2

Вычислительные эксперименты

P – количество процессорных узлов





Количество переменных: 32 000
Количество неравенств: 64 002

Количество переменных: 54 000
Количество неравенств: 108 002

Спасибо за внимание!