



Исследование масштабируемости апекс-метода для решения сверхбольших задач линейного программирования на кластерных вычислительных системах

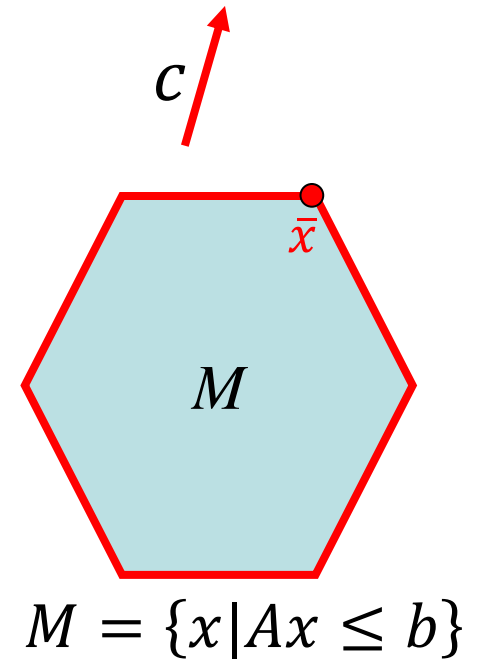
д.ф.-м.н., Л.Б. Соколинский,
к.ф.-м.н., И.М. Соколинская

Южно-Уральский государственный университет
(национальный исследовательский университет)

Задача линейного программирования

$$\bar{x} = \arg \max \{ \langle c, x \rangle \mid Ax \leq b \}$$

- $x \in \mathbb{R}_n$
- A – матрица $m \times n$
- c, b – векторы размерности n
- $\langle c, x \rangle$ – скалярное произведение



Апекс-метод

- Масштабируемый итерационный метод
- Может применяться к нестационарным задачам
- Построен по схеме предиктор-корректор
 - Quest – находит допустимую точку \tilde{x}
 - Target – строит по границе M путь до точки \bar{x} с максимальным значением целевой функции

Фаза Quest

$$\varphi_M: \mathbb{R}^n \rightarrow \mathbb{R}^n$$

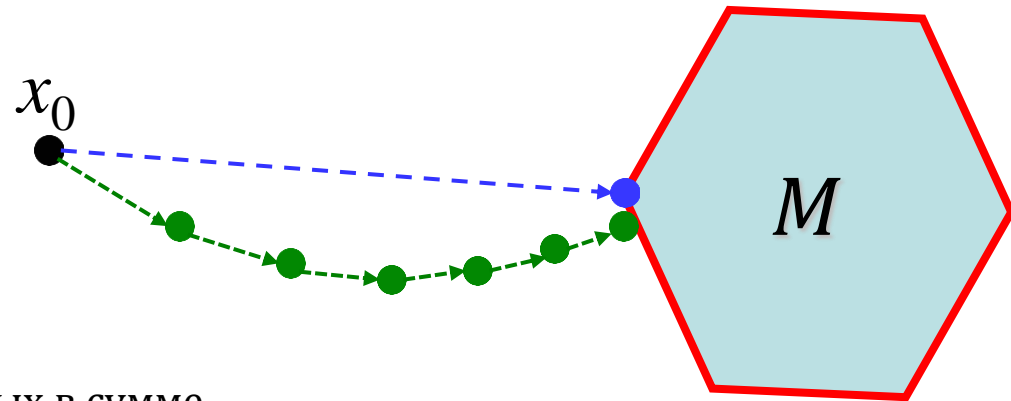
$$\rho_i^+(x) = \frac{\max\{\langle a_i, x \rangle - b_i, 0\}}{\|a_i\|^2} a_i$$

$$\varphi_M(x) = \frac{1}{h} \sum_{i=1}^m \rho_i^+(x)$$

h – количество ненулевых слагаемых в сумме

$$\varphi_M^{(k)}(x) = \underbrace{\varphi_M(\varphi_M(\dots \varphi_M(x) \dots))}_k$$

$$\pi_M(x) = \lim_{k \rightarrow \infty} \varphi_M^{(k)}(x)$$

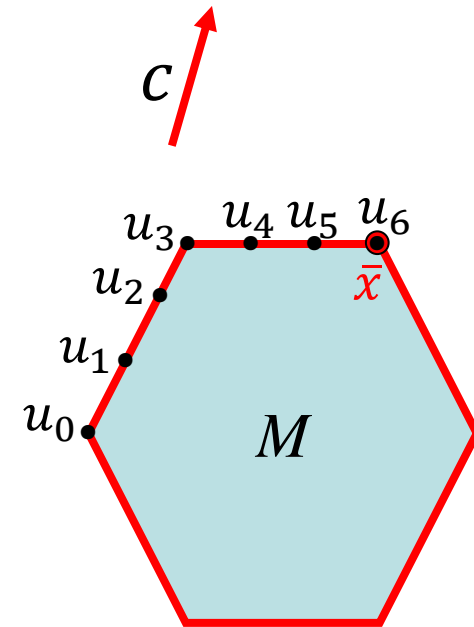


---> проектирование

---> псевдопроектирование

Фаза Target

1. Построение начального приближения u_0 на границе многогранника
2. Построение последовательности граничных точек u_1, u_2, u_3, \dots , сходящейся к точному решению \bar{x}



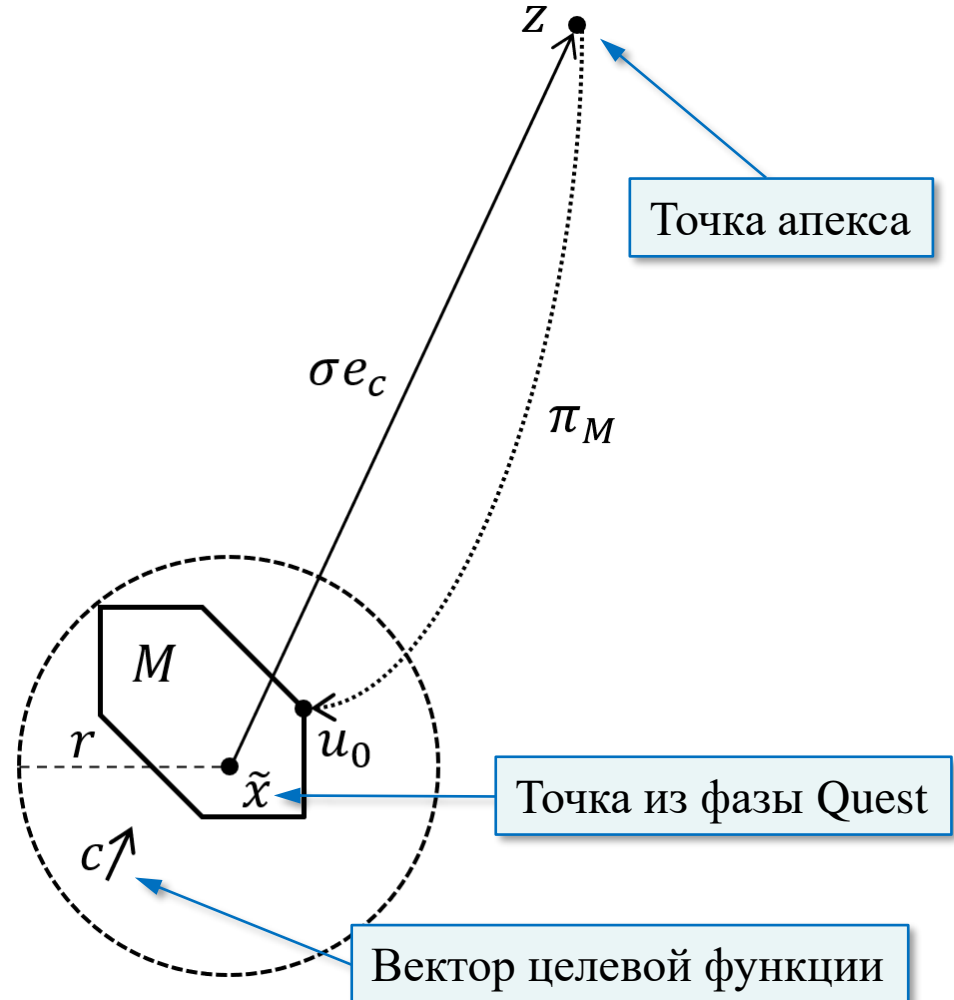
Построение начального приближения u_0

$$\sigma \gg r$$

$$e_c = \frac{c}{\|c\|}$$

$$z = \tilde{x} + \sigma e_c.$$

$$u_0 = \pi_M(z)$$



Построение последовательности

$$u_1, u_2, u_3, \dots \rightarrow \bar{x}$$

$$\delta > 0$$

$$e_c = \frac{c}{\|c\|}$$

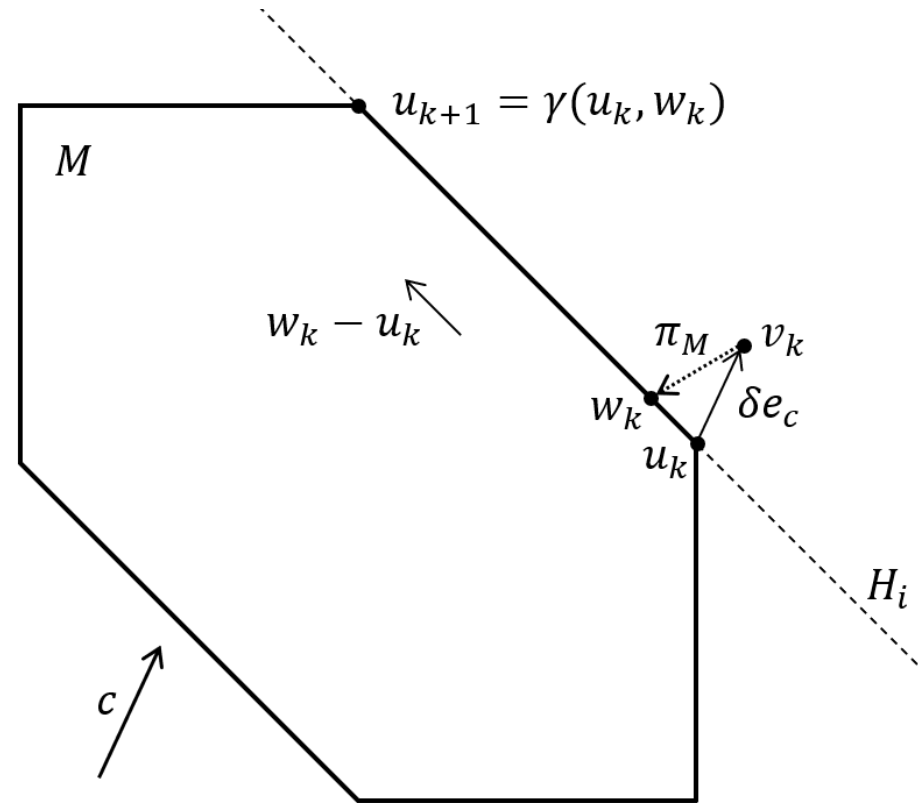
$$v_k = u_k + \delta e_c$$

$$w_k = \pi_M(v_k)$$

$$\langle c, w_k \rangle < \langle c, u_k \rangle \Rightarrow u_k \approx \bar{x}$$

$$L_{u_k w_k} = \{x \in \mathbb{R}^n \mid x = u_k + \eta(w_k - u_k), \eta \in \mathbb{R}_{\geq 0}\}$$

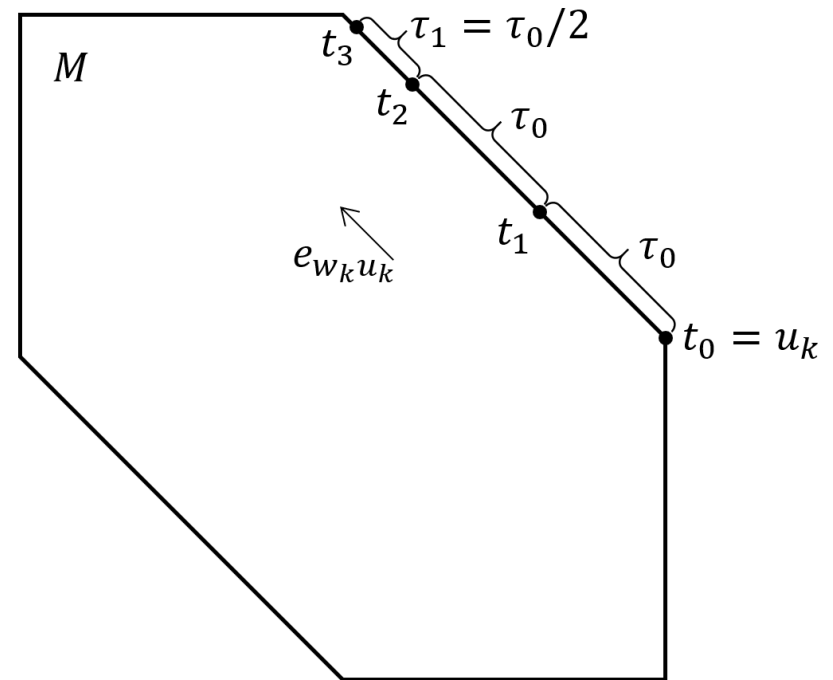
$$u_{k+1} = \gamma(u_k, w_k) = \operatorname{argmax}\{\|x - u_k\| \mid x \in L_{u_k w_k} \cap M\}$$



Вычисление $\gamma(u_k, w_k)$

Алгоритм 3.

```
1:  input  $u_k, w_k$ 
2:   $e_{u_k w_k} := (w_k - u_k) / \|w_k - u_k\|$ 
3:   $\tau_0 := \mu; t_0 := u_k$ 
4:   $j := 0; i := 0$ 
5:  if  $\tau_i < \varepsilon$  go to 13
6:   $t_{j+1} := t_j + \tau_i e_{u_k w_k}$ 
7:  if  $t_{j+1} \notin M$  go to 10
8:   $j := j + 1$ 
9:  go to 6
10:  $\tau_{i+1} := \tau_i / 2$ 
11:  $i := i + 1$ 
12: go to 5
13: output  $u_{k+1} = t_j; \text{ stop}$ 
```



Масштабируемая задача ЛП

$$\left\{ \begin{array}{cccccc}
 x_0 & & & & \leq & 200 \\
 & x_1 & & & \leq & 200 \\
 & & \ddots & & \vdots & \dots \\
 & & & x_{n-1} & \leq & 200 \\
 x_0 & + & x_1 & \dots & + & x_{n-1} & \leq & 200(n-1) + 100 \\
 x_0 & + & x_1 & \dots & + & x_{n-1} & \geq & 100 \\
 x_0 & & & & \geq & & & 0 \\
 & x_1 & & & \geq & & & 0 \\
 & & \ddots & & \vdots & & & \dots \\
 & & & x_{n-1} & \geq & & & 0
 \end{array} \right.$$

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

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

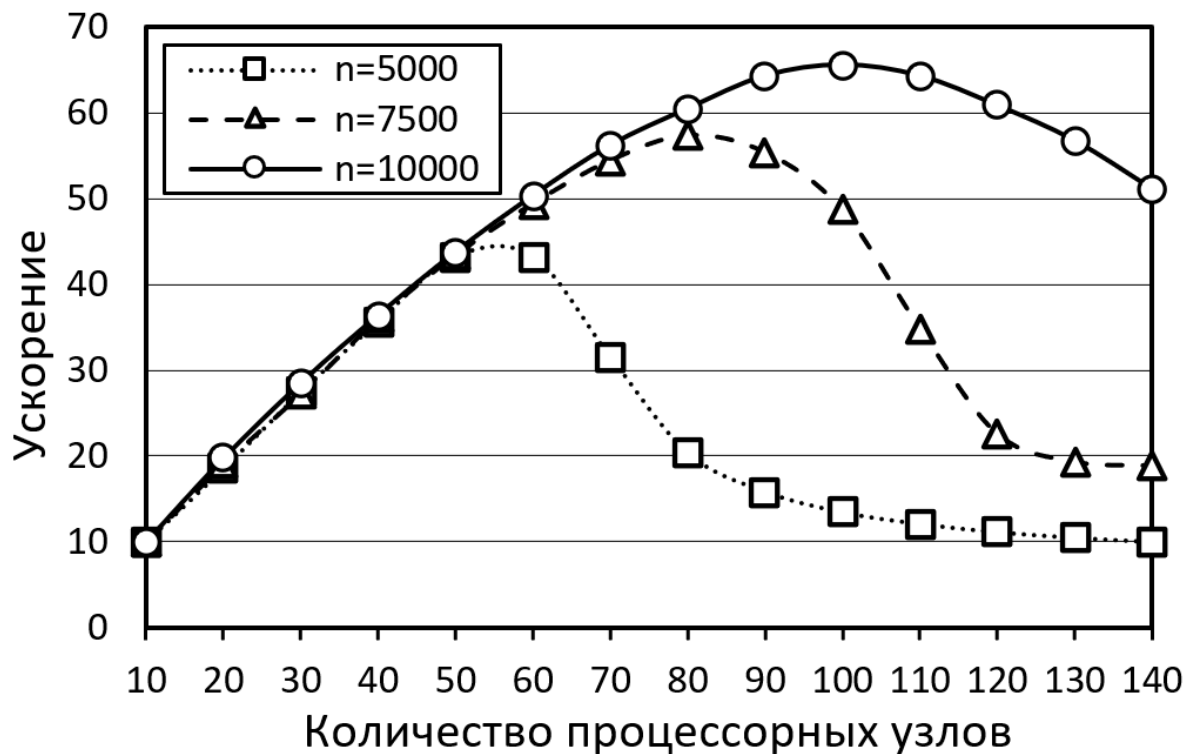
Параллельная реализация

- 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»

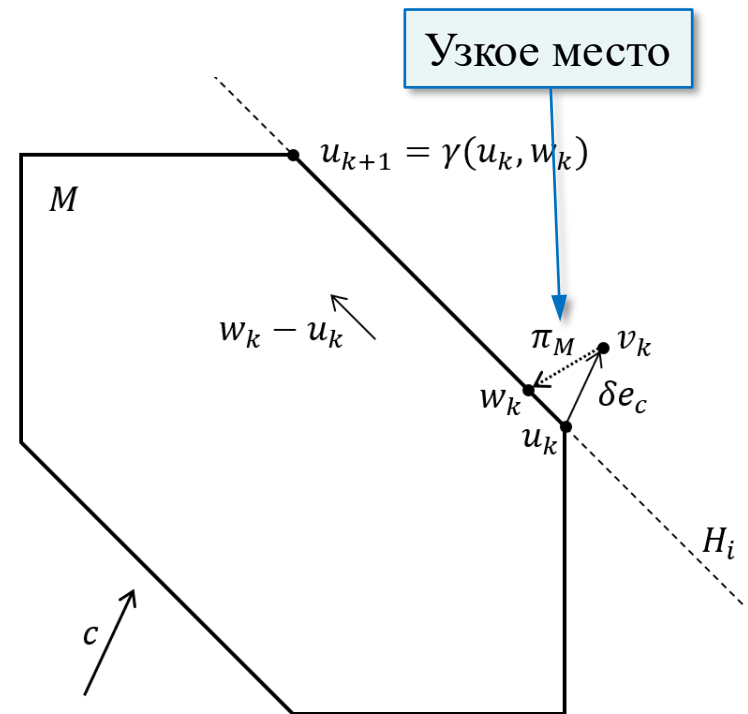
Количество процессорных узлов	480
Процессоры	Intel Xeon X5680 (6 cores 3.33 GHz)
Количество процессоров в узле	2
Оперативная память узла	24 GB DDR3
Соединительная сеть	InfniBand QDR (40 Gbit/s)
Операционная система	Linux CentOS

Вычислительные эксперименты



Узкое место

- 99% времени приходится на вычисление псевдопроекций
- Вычисление одного приближения u_k для задачи размерности 10000 на 100 процессорных узлах составило 44 минуты



Кому нужен такой алгоритм?

- Алгоритм предполагается использовать для обучения искусственной нейронной сети, которая будет решать задачи линейного программирования большой размерности

Спасибо за внимание!