



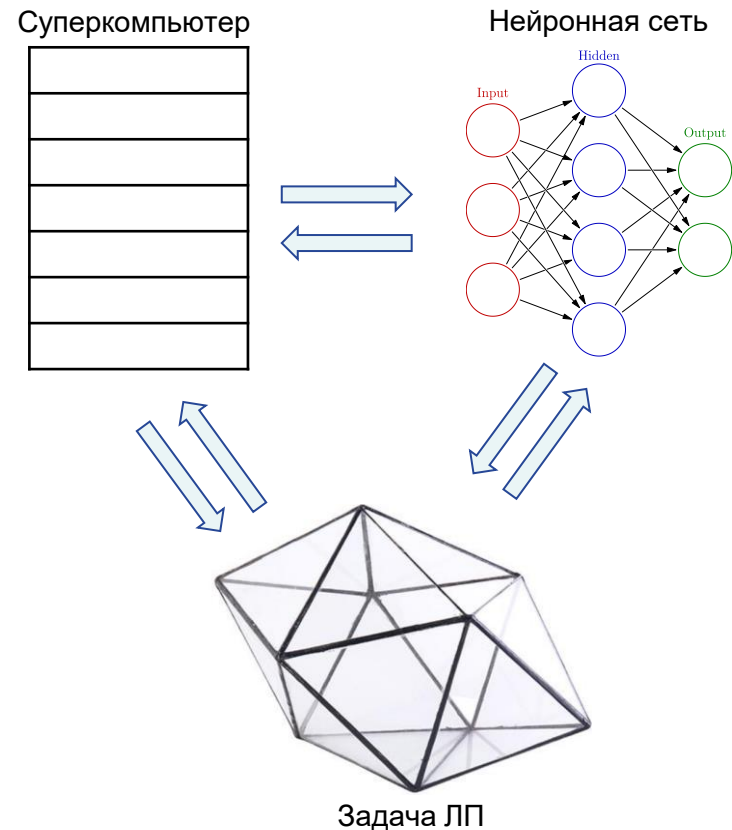
# FRaGenLP: генератор случайных задач линейного программирования для кластерных вычислительных систем

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

Южно-Уральский государственный университет  
(национальный исследовательский университет)

# Для чего нужен генератор случайных задач ЛП

- Проект по синтезу искусственной нейронной сети и суперкомпьютера для решения больших задач ЛП
- Генератор случайных задач ЛП нужен для генерации 70000 прецедентов для обучения искусственной нейронной сети



# Требования к генератору

---

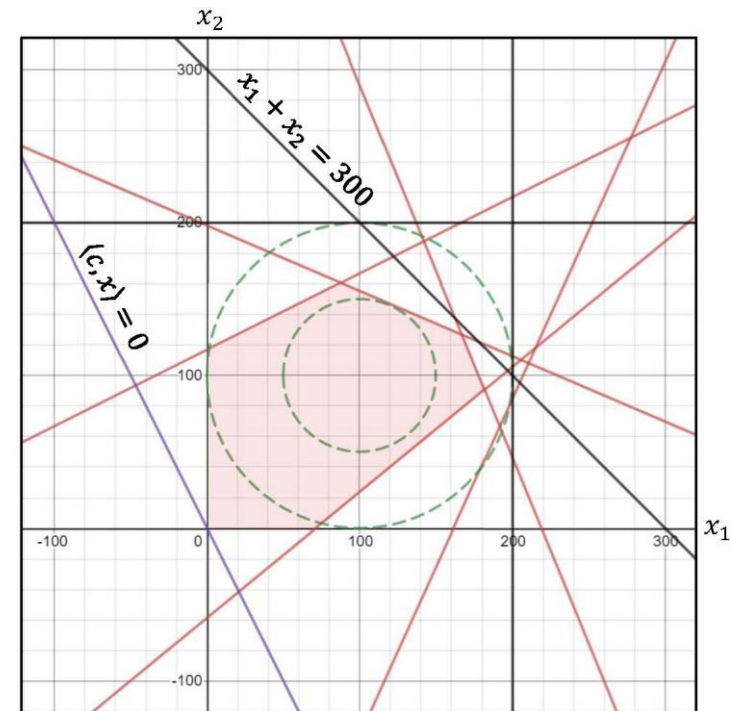
- Требования к генерируемым задачам ЛП
  - Случайные ограничения, влияющие на решение
  - Совместность
  - Ограниченность
  - Отсутствие нулевых коэффициентов в случайных ограничениях
- Параметры
  - Размерность
  - Общее количество ограничений
  - Количество случайных ограничений
- Параллельность алгоритма: генерация большого количества больших задач за минимальное время

# Известные генераторы

<p>Charnes A. et al. On Generation of Test Problems for Linear Programming Codes // Communications of the ACM. 1974. Vol. 17, no. 10. P. 583–586.</p>	<ul style="list-style-type: none"><li>• Основа: задача ЛП с известным решением</li><li>• Задача модифицируется так, чтобы решение не менялось</li><li>• Последовательный алгоритм</li></ul>
<p>Arthur J.L., Friendewey J.O. GENGUB: A generator for linear programs with generalized upper bound constraints // Computers and Operations Research. Pergamon, 1993. Vol. 20, no. 6. P. 565–573.</p>	<ul style="list-style-type: none"><li>• Задается решение</li><li>• Строятся обобщенные верхние границы</li><li>• Варьируется разреженность коэффициентов</li><li>• Последовательный алгоритм</li></ul>
<p>Castillo E., Pruneda R.E., Esquivel Mó. Automatic generation of linear programming problems for computer aided instruction // International Journal of Mathematical Education in Science and Technology. Taylor &amp; Francis Group, 2001. Vol. 32, no. 2. P. 209–232.</p>	<ul style="list-style-type: none"><li>• Учебный характер</li><li>• Генерация различных типов задач ЛП<ul style="list-style-type: none"><li>• Ограниченные</li><li>• Неограниченные</li><li>• Единственное решение</li><li>• Множественное решение</li></ul></li><li>• Последовательный алгоритм</li></ul>

# Идея

- Использовать опорные ограничения (гарантируют замкнутость)
- Использовать две вложенные гиперсферы (гарантируют совместность)
- Постоянная целевая функция с убывающими коэффициентами (добавляются ограничения, способные повлиять на решение)



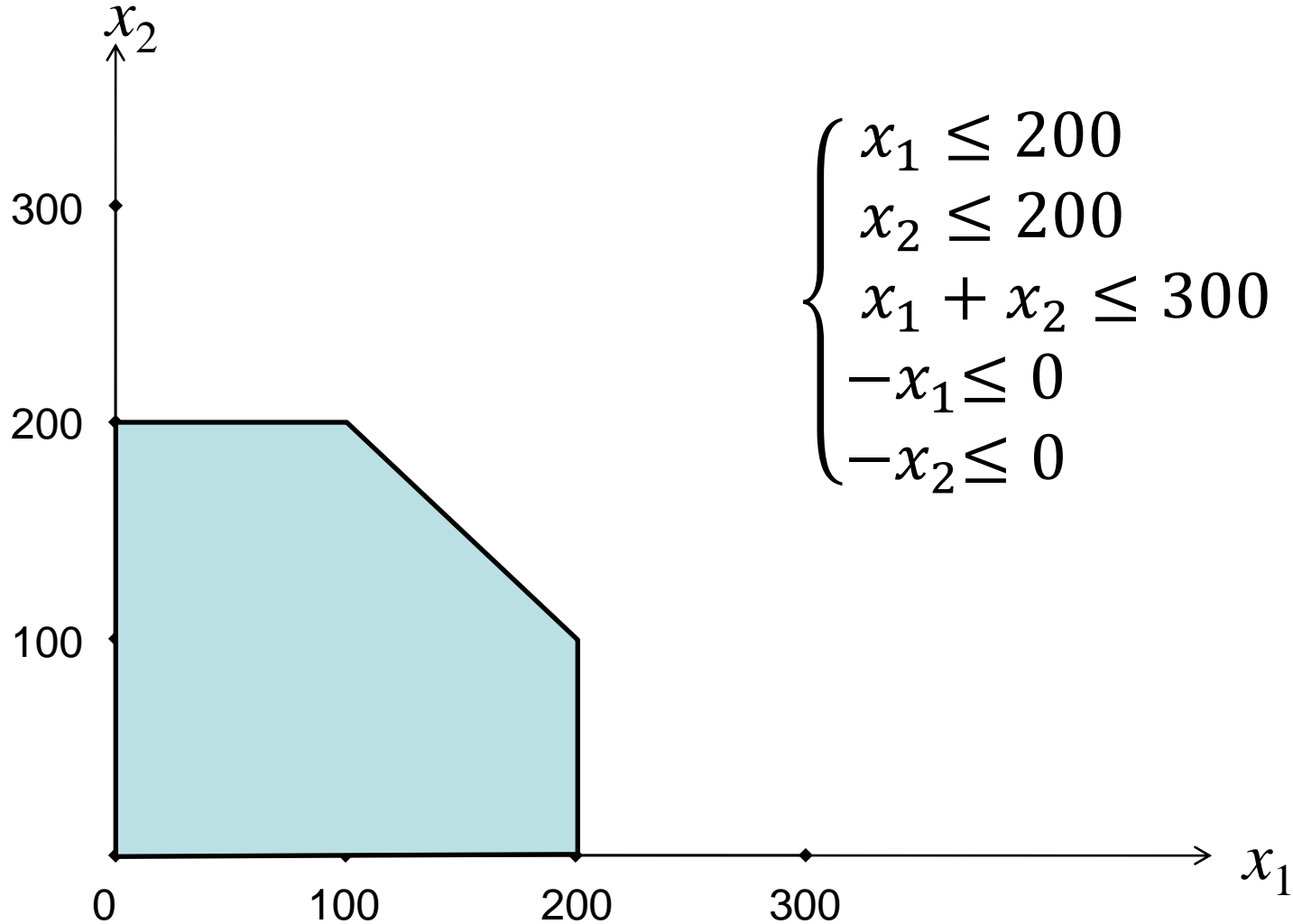
# Опорные ограничения

$$\left\{ \begin{array}{l} x_1 \leq \alpha \\ x_2 \leq \alpha \\ \vdots \\ x_n \leq \alpha \\ x_1 + x_2 + \dots + x_n \leq \alpha(n-1) + \frac{\alpha}{2} \\ -x_1 \leq 0 \\ -x_2 \leq 0 \\ \vdots \\ -x_n \leq 0 \end{array} \right.$$

Количество переменных:  $n$

Количество неравенств:  $m = 2n + 1$

# Опорные ограничения при $n = 2$ и $\alpha = 200$



# Целевая функция

---

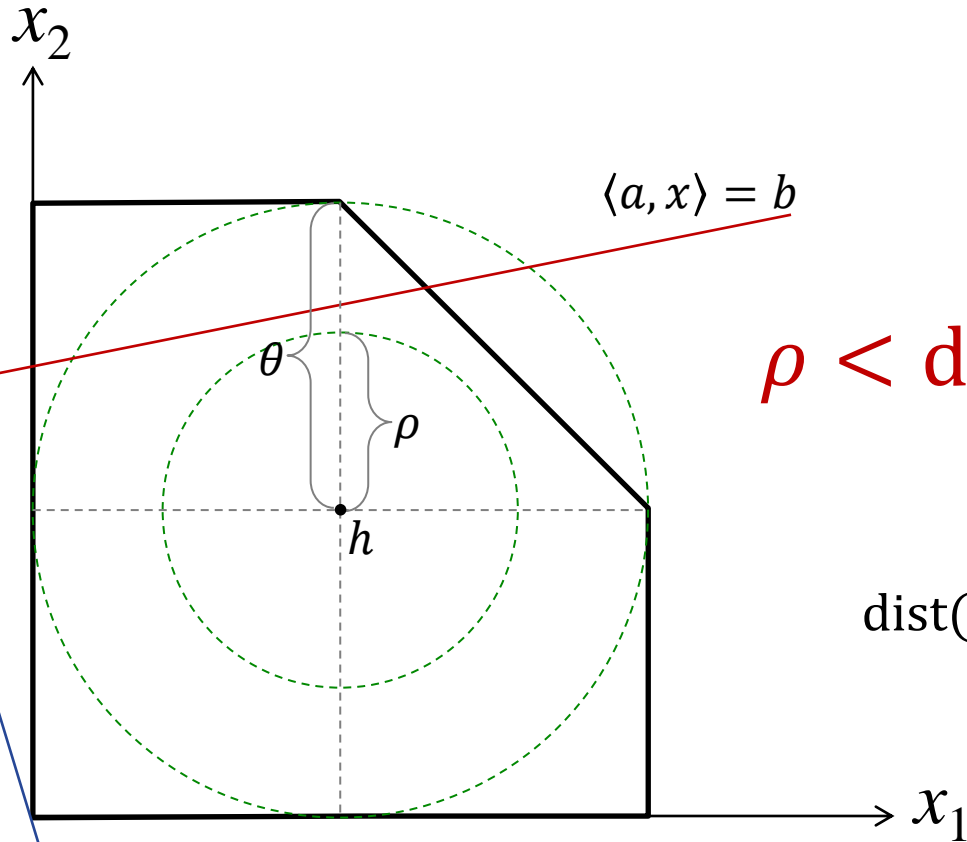
$$f(x) = \langle c, x \rangle = \frac{\alpha}{2} (nx_1 + (n-1)x_1 + \dots + 1)$$

$$c = \left( \frac{\alpha}{2} nx_1, \frac{\alpha}{2} (n-1)x_1, \dots, \frac{\alpha}{2} \right)$$



# Допустимые случайные ограничения

$$\langle c, x \rangle = 0$$



$$\langle a, x \rangle = b$$

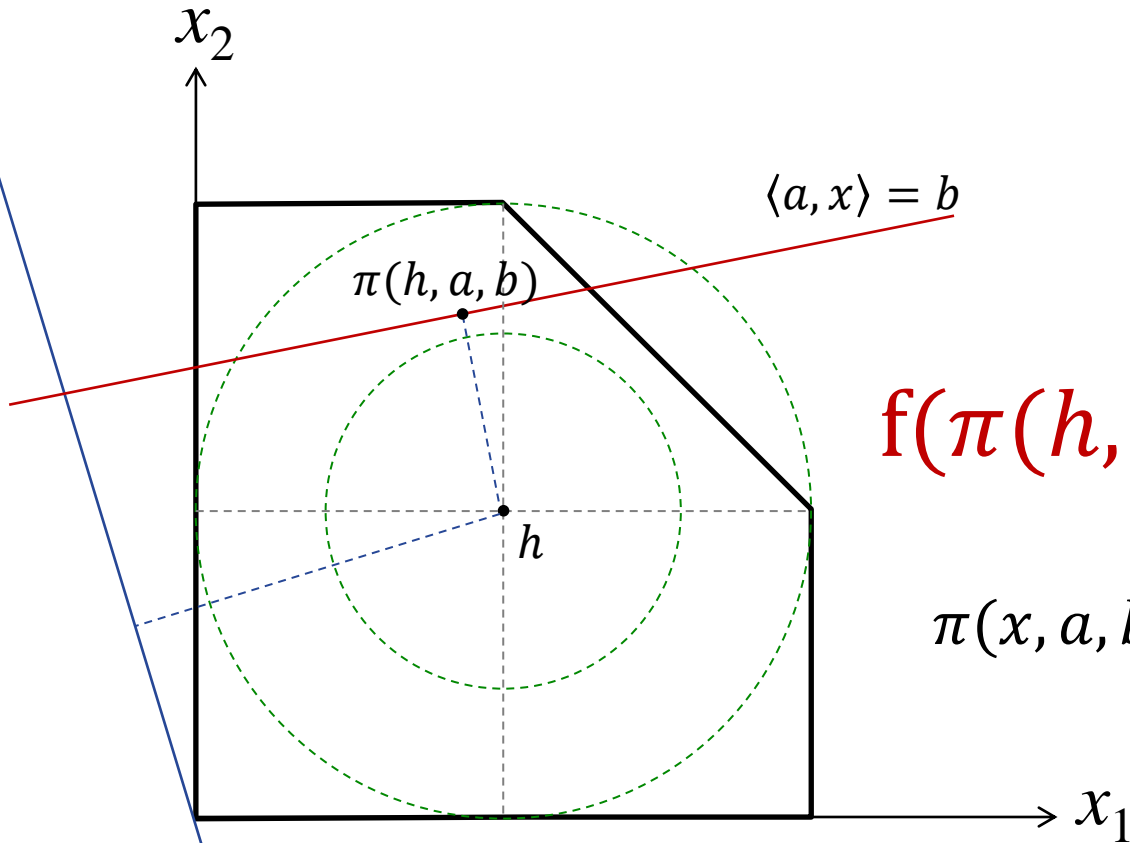
$$\rho < \text{dist}(a, b, h) \leq \theta$$

$$\text{dist}(a, b, h) = \frac{|\langle a, h \rangle - b|}{\|a\|}$$

0

# Допустимые случайные ограничения

$$\langle c, x \rangle = 0$$



$$f(\pi(h, a, b)) > f(h)$$

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

# Допустимые случайные ограничения

$$\langle c, x \rangle = 0$$

 $x_2$ 

$$\langle a', x \rangle = b'$$

$$\langle a'', x \rangle = b'$$

 $h$ 

$$\left\| \frac{a'}{\|a'\|} - \frac{a''}{\|a''\|} \right\| > L_{max}$$

 $x_1$ 

0

# Допустимые случайные ограничения

$$\langle c, x \rangle = 0$$

 $x_2$ 

$$\langle a', x \rangle = b'$$

$$\left| \frac{b'}{\|a'\|} - \frac{b''}{\|a''\|} \right| > S_{min}$$

 $h$ 

$$\langle a'', x \rangle = b''$$

 $x_1$ 

0

```

M1: input  $n, d, \alpha, \theta, \rho, S_{min}, L_{max}, a_{max}, b_{max}$ 
M2:  $k := 0$ 
M3:  $A_S := []$ 
M4:  $B_S := []$ 
M5:  $AddSupport(A_S, B_S)$ 
M6: for  $j = n \dots 1$  do  $c_j := \theta \cdot j$ 
M7:  $A_R := []$ 
M8:  $B_R := []$ 
M9: if  $d > 0$  goto M19
M10: output  $A_S, B_S, c$ 
M11: goto M37
M12:
M13:
M14:
M15:
M16:
M17:
M18:
M19:  $RecvFromAllWorkers(a^{(1)}, b^{(1)}, \dots, a^{(L)}, b^{(L)})$ 
M20: for  $l = 1 \dots L$  do begin
M21:    $is\_like := false$ 
M22:   for all  $(\bar{a}, \bar{b}) \in (A_R, B_R)$  do begin
M23:     if  $like(a^{(l)}, b^{(l)}, \bar{a}, \bar{b})$  then begin
M24:        $is\_like := true$ 
M25:       goto M28
M26:     end
M27:   end
M28: if  $is\_like$  continue
M29:  $A_R := A_R \# a^{(l)}$ 
M30:  $B_R := B_R \# b^{(l)}$ 
M31:  $k := k + 1$ 
M32: if  $k = d$  goto M34
M33: end
M34:  $SendToAllWorkers(k)$ 
M35: if  $k < d$  goto M19
M36: output  $A_S, B_S, A_R, B_R, c$ 
M37: stop

```

```

W1: input  $n, d, \alpha, \theta, \rho, S_{min}, L_{max}, a_{max}, b_{max}$ 
W2: if  $d \leq 0$  goto W37
W3:  $A_S := []$ 
W4:  $B_S := []$ 
W5:  $AddSupport(A_S, B_S)$ 
W6: for  $j = 1 \dots n$  do begin
W7:    $a_j^{(l)} := rsign() \cdot rand(0, a_{max})$ 
W8: end
W9:  $b^{(l)} := rsign() \cdot rand(0, b_{max})$ 
W10: if  $\langle a^{(l)}, h_{ctr} \rangle \leq b^{(l)}$  goto W13
W11: for  $j = 1 \dots n$  do  $a_j := -a_j$ 
W12:  $b^{(l)} := -b^{(l)}$ 
W13: if  $dist(a^{(l)}, b^{(l)}, h_{ctr}) < \rho$  goto W6
W14: if  $dist(a^{(l)}, b^{(l)}, h_{ctr}) > \theta$  goto W6
W15: if  $f(\pi(h_{ctr}, a^{(l)}, b^{(l)})) > f(h_{ctr})$  goto W6
W16: for all  $(\bar{a}, \bar{b}) \in (A_S, B_S)$  do begin
W17:   if  $like(a^{(l)}, b^{(l)}, \bar{a}, \bar{b})$  goto W6
W18: end
W19:  $SendToMaster(a^{(l)}, b^{(l)})$ 
W20:
W21:
W22:
W23:
W24:
W25:
W26:
W27:
W28:
W29:
W30:
W31:
W32:
W33:
W34:  $RecvFromMaster(k)$ 
W35: if  $k < d$  goto W6
W36:
W37: stop

```

# Параллельный алгоритм (мастер-рабочие)

---

# Параллельная реализация

---

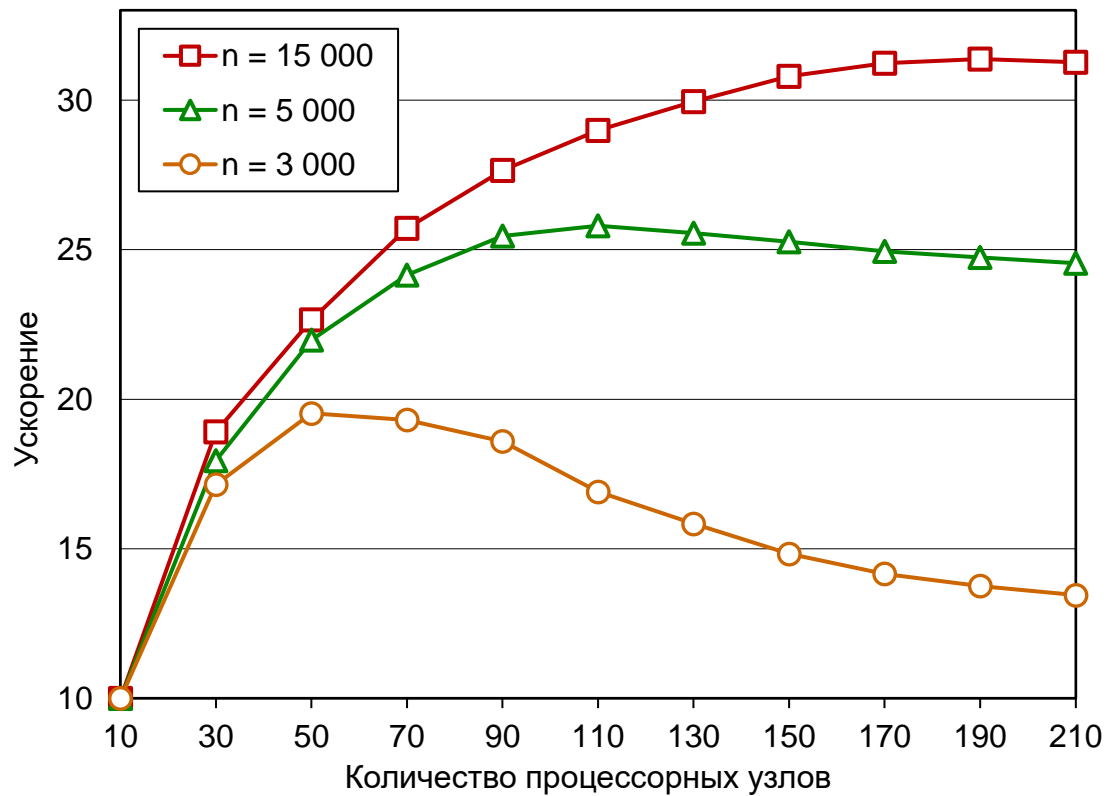
- C++
- MPI & OpenMP
- Параллельный программный каркас BSF
  - Sokolinsky L.B. BSF: A parallel computation model for scalability estimation of iterative numerical algorithms on cluster computing systems // Journal of Parallel and Distributed Computing. 2021. Vol. 149. P. 193–206.
  - <https://github.com/leonid-sokolinsky/BSF-skeleton>

# 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

# Масштабируемость





# Время генерации задачи

---

- Время генерации случайной задачи ЛП с 31501 ограничениями на конфигурации из одного узла-мастера и **1 узла-рабочего: 12 минут**
- Время генерации такой же задачи на конфигурации из одного узла-мастера и **170 узлов-рабочих: 22 секунды**

---

Спасибо за внимание!