



VaLiPro: валидатор решений задач линейного программирования для кластерных вычислительных систем

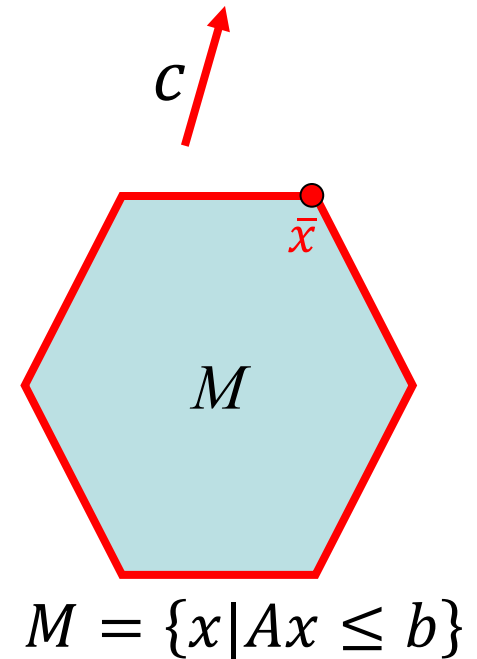
д.ф.-м.н., Л.Б. Соколинский,
к.ф.-м.н., И.М. Соколинская

Южно-Уральский государственный университет
(национальный исследовательский университет)

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

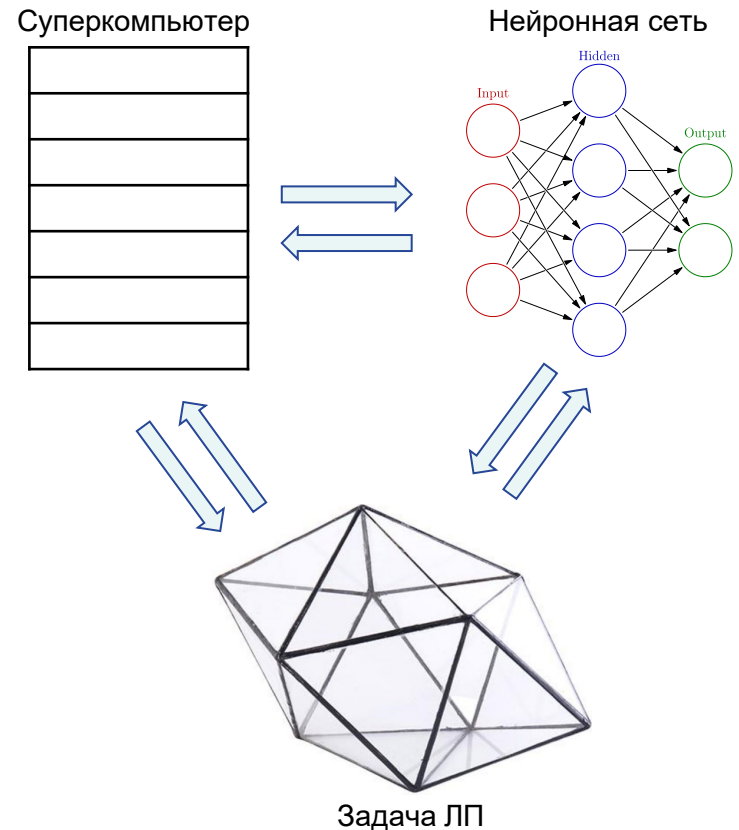


Большие задачи ЛП

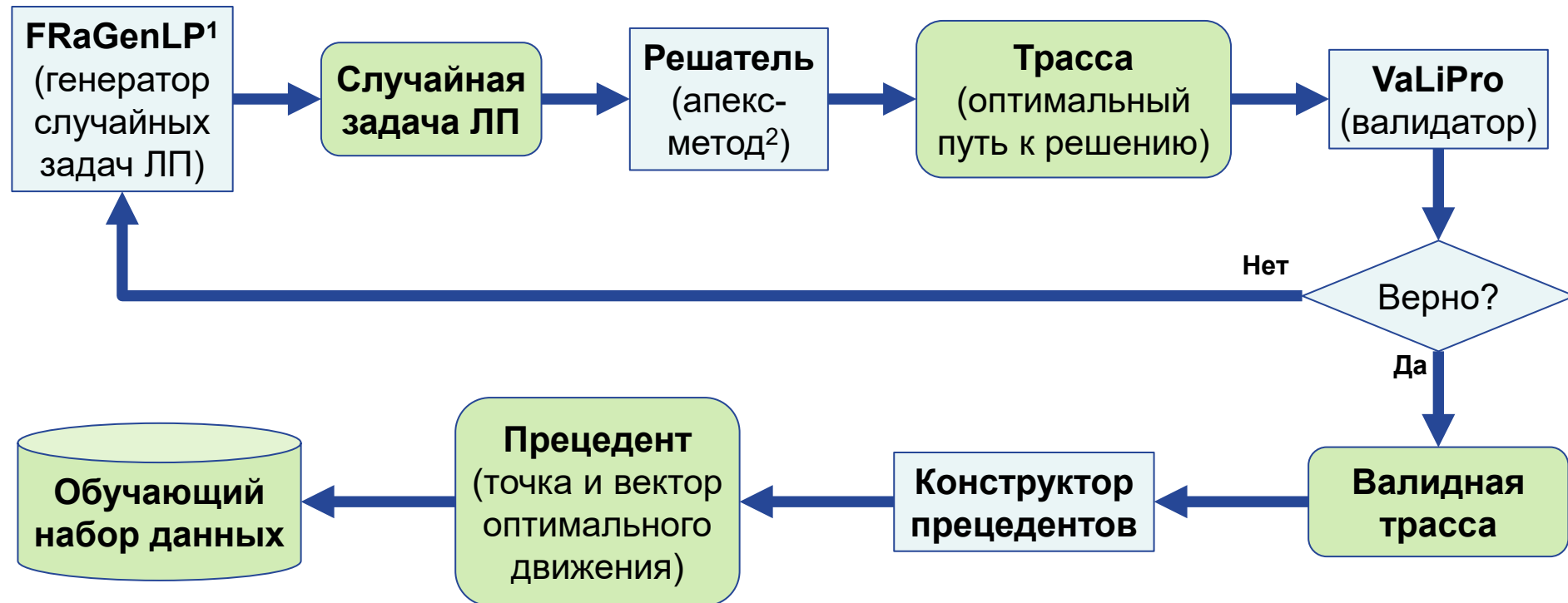
- Экономика
- Физика
- Логистика
- Составление расписаний
- Алгоритмическая торговля на биржах
(нестационарный характер)

Для чего нужен валидатор случайных задач ЛП

- Проект по синтезу искусственной нейронной сети и суперкомпьютера для решения больших задач ЛП



Построение обучающего набора данных



¹ Sokolinsky L.B., Sokolinskaya I.M. FRaGenLP: A Generator of Random Linear Programming Problems for Cluster Computing Systems // Parallel Computational Technologies. PCT 2021. Communications in Computer and Information Science, vol. 1437. 164-177.

² Sokolinsky L.B., Sokolinskaya I.M. Scalable Method for Linear Optimization of Industrial Processes // Proceedings - 2020 Global Smart Industry Conference, GloSIC 2020. IEEE, 2020. P. 20–26. Article number 9267854.

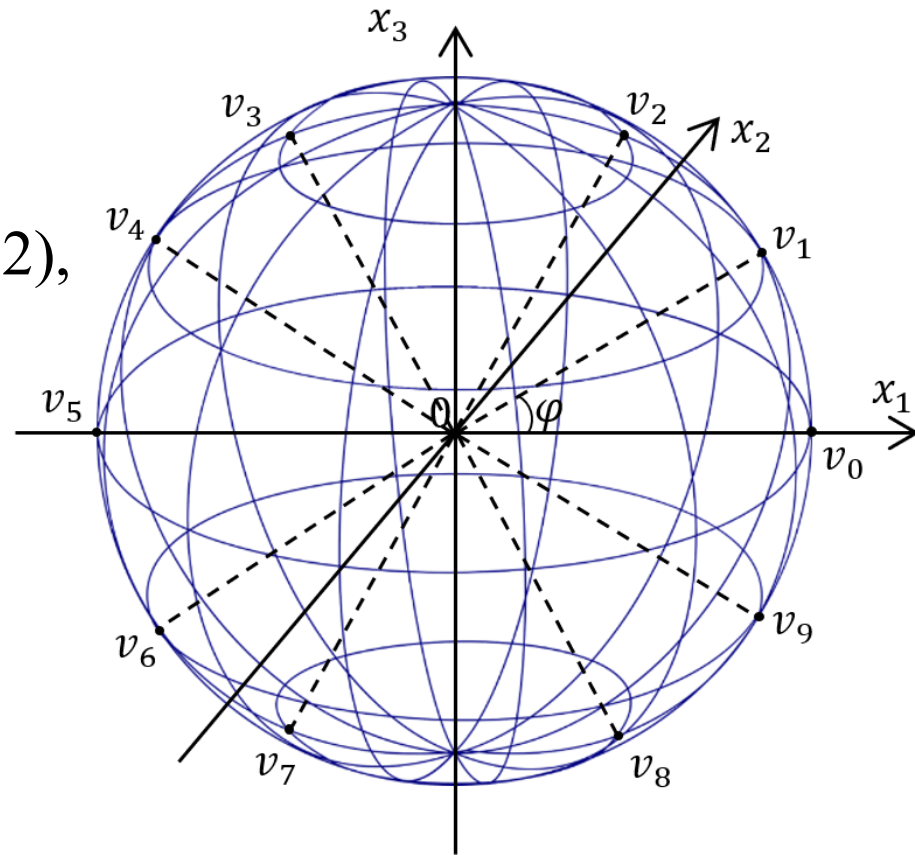
Метод валидации VaLiPro (Validator of Linear Program)

$$v_1 = \rho \cos \phi_1,$$

$$v_j = \rho \cos \phi_j \prod_{i=1}^{j-1} \sin \phi_i \quad (j = 2, \dots, n-2),$$

$$v_{n-1} = \rho \sin \theta \prod_{i=1}^{n-2} \sin \phi_i,$$

$$v_n = \rho \cos \theta \prod_{i=1}^{n-2} \sin \phi_i,$$



Простейший алгоритм

```
1: input d
2:  $\varphi := \pi/d$ 
3: for  $j_{n-1} = 0 \dots (2d - 1)$  do begin
4:    $\theta := j_{n-1}\varphi$ 
5:   for  $j_{n-2} = 0 \dots d$  do begin
6:      $\phi_{n-2} := j_{(n-2)}\varphi$ 
7:     ...
8:     for  $j_2 = 0 \dots d$  do begin
9:        $\phi_2 := j_2\varphi$ 
10:      for  $j_1 = 0 \dots d$  do begin
11:         $\phi_1 := j_1\varphi$ 
12:         $\Pi := 1$ 
13:         $v_1 := \rho \cos(\phi_1)$ 
14:        for  $l = 2 \dots n - 2$  do begin
15:           $\Pi := \sin(\phi_{l-1})\Pi$ 
16:           $v_l := \rho \cos(\phi_l) \Pi$ 
17:        end
18:         $v_{n-1} := \rho \sin(\theta) \Pi$ 
19:         $v_n := \rho \cos(\theta) \Pi$ 
20:        output v
21:      end
22:    end
23:  ...
24: end
25: end
26: stop
```

- Число вложенных циклов зависит от размерности
- Появление дубликатов: для размерности $n=4$ при количестве параллелей $d=5$ алгоритм порождает более 50% дубликатов (189/360)

Проблема дубликатов

$$v_j = \rho \cos \phi_j \prod_{i=1}^{j-1} \sin \phi_i \quad (j = 2, \dots, n-2)$$

- Дубликаты порождаются в итерациях, когда $\phi_i = 0$ или $\phi_i = \pi$

Алгоритм без дубликатов

```
1: input  $d, \rho$ 
2:  $\varphi := \pi/d$ 
3: for  $j_{n-1} = 0 \dots (2d - 1)$  do begin
4:    $\theta := j_{n-1}\varphi$ 
5:   for  $j_{n-2} = 1 \dots (d - 1)$  do begin
6:      $\phi_{n-2} := j_{n-2}\varphi$ 
7:     ...
8:     for  $j_2 = 1 \dots (d - 1)$  do begin
9:        $\phi_2 := j_2\varphi$ 
10:      for  $j_1 = 1 \dots (d - 1)$  do begin
11:         $\phi_1 := j_1\varphi$ 
12:         $\Pi := 1$ 
13:         $v_1 := \rho \cos(\phi_1)$ 
14:        for  $l = 2 \dots n - 2$  do begin
15:           $\Pi := \sin(\phi_{l-1})\Pi$ 
16:           $v_l := \rho \cos(\phi_l)\Pi$ 
17:        end
18:         $v_{n-1} := \rho \sin(\theta)\Pi$ 
19:         $v_n := \rho \cos(\theta)\Pi$ 
20:        output  $v$ 
21:      end
22:    end
23:    ...
24:  end
25: end
26: stop
```

- Вместе с дубликатами теряется 7% значимых точек
- Число вложенных циклов зависит от размерности

Функция g

(вычисление точки v по ее номеру k)

```
1: function  $g(k)$  begin
2:    $u_{n-1} := \lfloor k/(d-1)^{n-2} \rfloor$ 
3:    $u_n := u_{n-1}$ 
4:    $k := k \bmod (d-1)^{n-2}$ 
5:   for  $j = (n-3) \dots 0$  do begin
6:      $u_j := \lfloor k/(d-1)^j \rfloor + 1$ 
7:      $k := k \bmod (d-1)^j$ 
8:   end
9:    $\Pi := 1$ 
10:   $\varphi := \pi/d$ 
11:   $v_1 := \rho \cos(u_1\varphi)$ 
12:  for  $j = 2 \dots (n-2)$  do begin
13:     $\Pi = \Pi \sin(u_{j-1}\varphi)$ 
14:     $v_j = \rho \cos(u_j\varphi) \Pi$ 
15:  end
16:   $\Pi = \Pi \sin(u_{n-2}\varphi)$ 
17:   $v_{n-1} := \rho \sin(u_{n-1}\varphi) \Pi$ 
18:   $v_n := \rho \cos(u_n\varphi) \Pi$ 
19:  return  $v$ 
20: end
```

Итоговый алгоритм VaLiPro

```
1: input  $n, A, b, c, d, \rho, \varepsilon, \tilde{x}$ 

2:    $\varphi := \pi/d$ 

3:   for  $k = 0 \dots 2d(d - 1)^{n-2} - 1$  do begin

4:      $v := g(k)$ 

5:     if  $Av \leq b \wedge \langle c, v \rangle > \langle c, \tilde{x} \rangle + \varepsilon$  goto 9

6:   end

7:   output "Solution"  $\tilde{x}$  "is correct."

8:   goto 10

9:   output "Solution"  $\tilde{x}$  "is incorrect."

10: stop
```

Параллельный алгоритм валидации решения задачи ЛП

Мастер	L -й рабочий ($L=0, \dots, L-1$)
M1:	W1: input $n, A, b, c, d, \rho, \varepsilon, \tilde{x}$
M2:	W2: $W_l := [lK/L, \dots, (l+1)K/L - 1]$
M3:	W3: $Z_l := \text{Map}(f_{\tilde{x}}, W_l)$
M4:	W4: $s_l := \text{Reduce}(\wedge, Z_l)$
M5: $\text{RecvFromWorkers}(s_1, \dots, s_L)$	W5: $\text{SendToMaster}(s_l)$
M6: $s := \text{Reduce}(\wedge, [s_1, \dots, s_L])$	W6:
M7: if $s = \text{false}$ goto M10	W7:
M8: output "Solution" \tilde{x} "is correct."	W8:
M9: goto M11	W9:
M10: output "Solution" \tilde{x} "is incorrect."	W10:
M11: stop	W11: stop

$$W = [0, \dots, K-1] \quad (K = 2d(d-1)^{n-2})$$

$$Z = \text{Map}(f_x, W) = [f_{\tilde{x}}(w_0), \dots, f_{\tilde{x}}(w_{K-1})]$$

$$f_{\tilde{x}}(w) = \begin{cases} \text{true} & | A \cdot g(w) \leq b \wedge \langle c, g(w) \rangle \leq \langle c, \tilde{x} \rangle \\ \text{false} & | A \cdot g(w) > b \vee \langle c, g(w) \rangle > \langle c, \tilde{x} \rangle \end{cases}$$

$$s = \text{Reduce}(\wedge, Z) = z_0 \wedge \dots \wedge z_{K-1}$$

Реализация

- C++

- Модель параллельных вычислений 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]

- Параллельный BSF-каркас

[Sokolinsky L.B. BSF-skeleton: A template for parallelization of iterative numerical algorithms on cluster computing systems // MethodsX. 2021. Vol. 8. Article 101437]

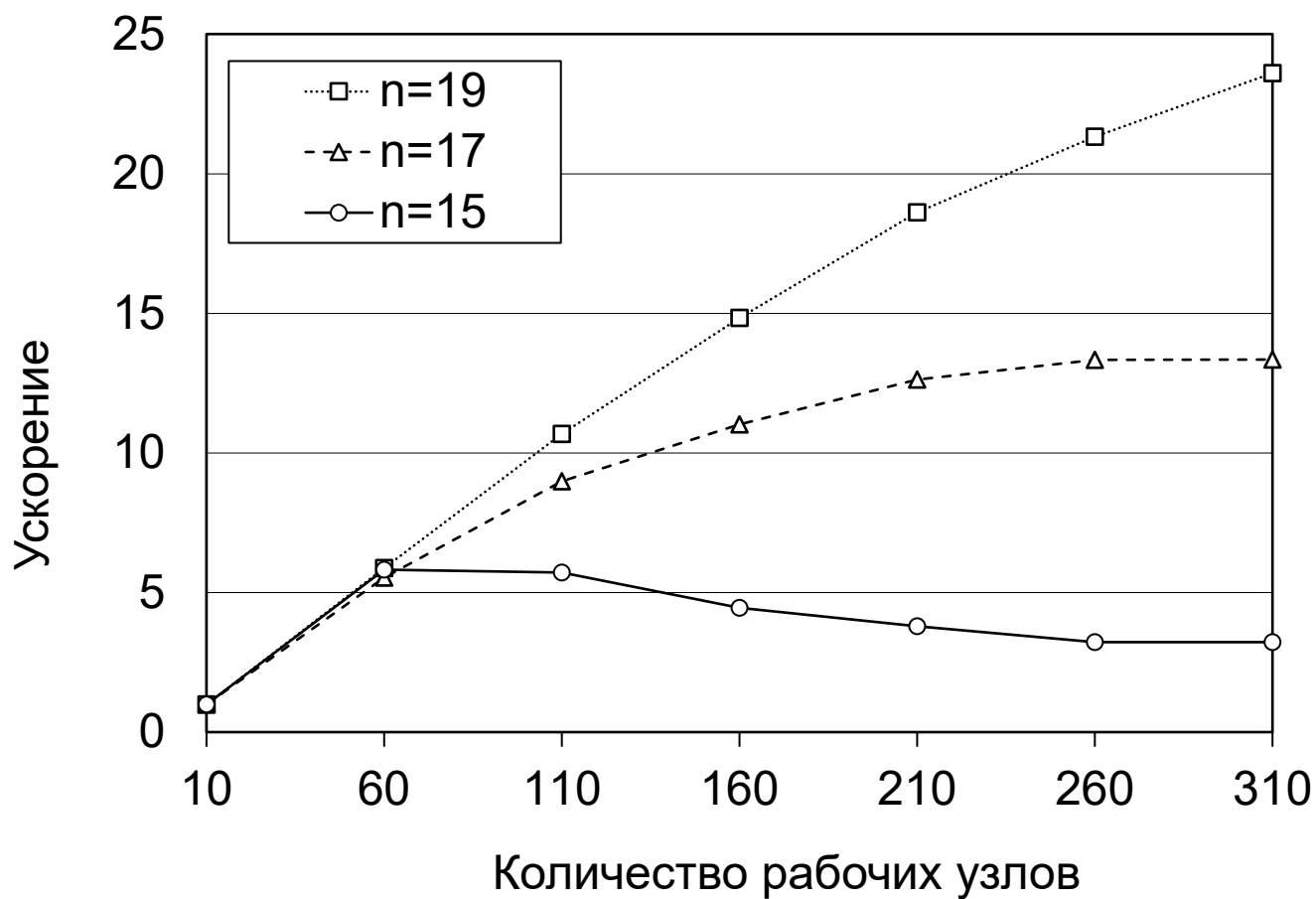
- <https://github.com/leonid-sokolinsky/BSF-LPP-Validator>

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



Спасибо за внимание!