



О новом подходе к решению нестационарных задач линейного программирования на кластерных вычислительных системах

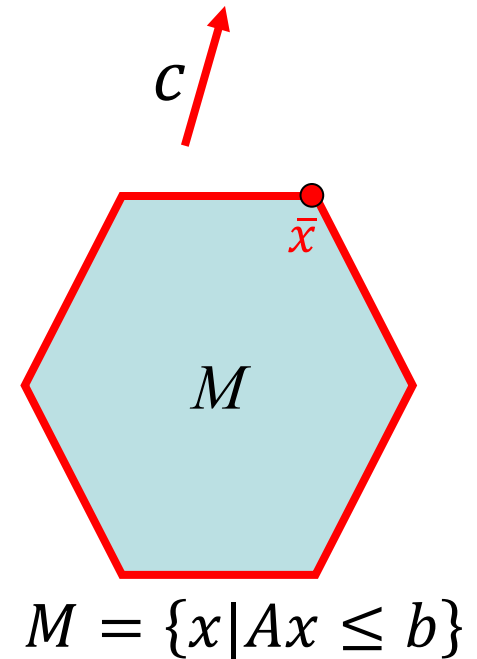
д.ф.-м.н., Л.Б. Соколинский,
к.ф.-м.н., И.М. Соколинская

Южно-Уральский государственный университет
(национальный исследовательский университет)

Задача линейного программирования (ЛП)

$$\bar{x} = \arg \max \{f(x) \mid Ax \leq b\}$$

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



Нестационарные задачи ЛП

- Целевая функция и/или ограничения изменяются в течение вычислительного процесса
- Примеры
 - выбор оптимальных стратегий в роботрейдинге
 - оптимальное управление летательными аппаратами
 - оптимальное управление технологическими процессами
 - логистические и транспортные задачи
 - оперативное планирование и управление производством продукции

Оптимизация в режиме реального времени

- Управление химическим производством
- Управление системой многоточечного впрыска топлива в ДВС
- Управление сотовыми сетями
- Автопилотирование
- Системы самонаведения ракет

Режим реального времени

- Решение задачи ЛП должно выполняться за определенное время
- Методы решения
 - Рассматривать каждое изменение как появление новой задачи оптимизации, которую необходимо решать с нуля
 - Непрерывно адаптировать решение к изменяющейся среде, повторно используя информацию, полученную в прошлом
 - Подход применим, если алгоритм достаточно быстро отслеживает траекторию движения оптимальной точки
 - В случае больших задач ЛП требует разработки масштабируемых методов и параллельных алгоритмов ЛП

Решатели

- Симплекс-метод
- Метод внутренних точек

Симплекс-метод

- Плохо распараллеливается на распределенной памяти (не более 16-32 процессорных узлов)
- При решении задач больших размерностей наблюдается потеря точности
- При изменении исходных данных необходимо начинать вычислительный процесс заново

Метод внутренних точек

- Адаптируется к динамическим изменениям исходных данных задачи
- Отсутствуют эффективные параллельные реализации для систем с распределенной памятью
- Во многих случаях необходимо находить начальную внутреннюю точку

Идея

Разработать новый метод решения нестационарных задач ЛП в реальном времени с использованием суперкомпьютерных и нейросетевых технологий

Метод поверхностного движения

1. Выбрать начальную точку u на поверхности многогранника
2. Вычислить на суперкомпьютере локальный образ допустимого многогранника в точке u
3. С помощью нейронной сети вычислить на поверхности многогранника направление d максимального увеличения $f(x)$
4. Двигаться по поверхности допустимого многогранника по направлению d так далеко, насколько это возможно
5. Полученную точку принять за новое приближение u и перейти на шаг 2
6. Закончить работу при $d = 0$

Выбор начальной точки: вычисление точки апекса

$$z = x' + \left(\eta + \max \left\{ \frac{b_i - \langle a_i, x' \rangle}{\langle a_i, e_c \rangle} \mid i \in \mathcal{J} \right\} \right) e_c$$

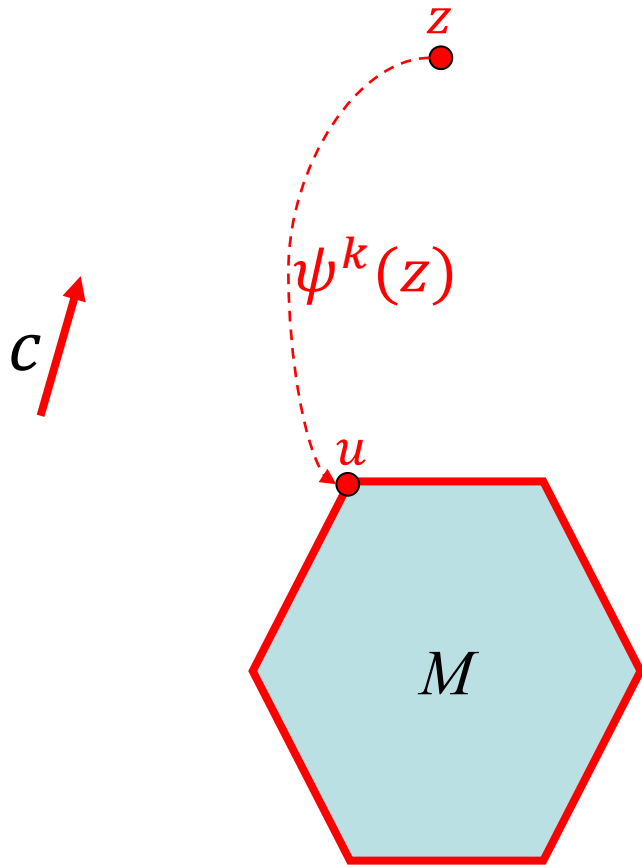
$\eta > 0$

c

$$\hat{H}_i = \{x \in \mathbb{R}^n \mid \langle a_i, x \rangle \leq b_i\}$$

$$\langle a_i, c \rangle > 0$$

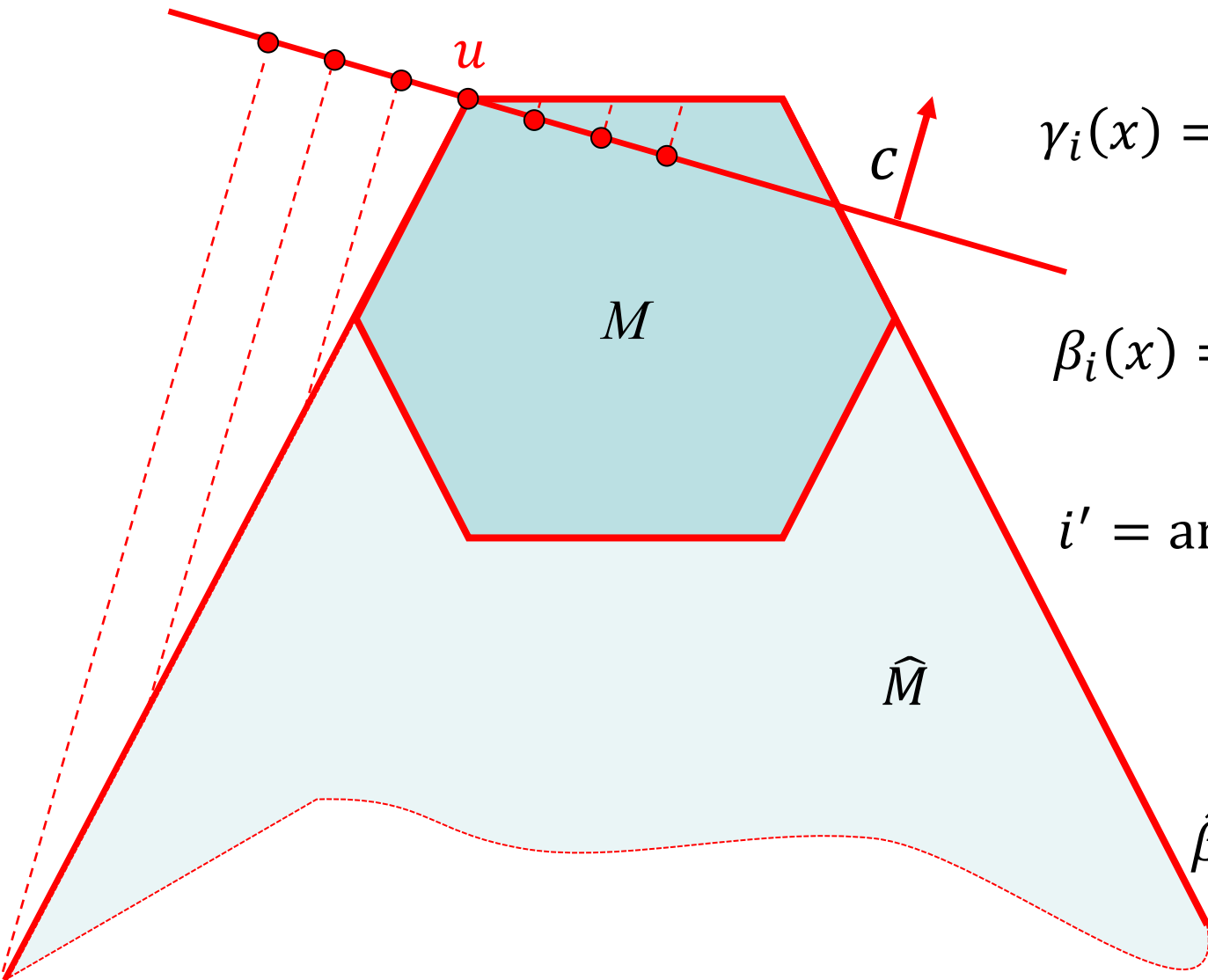
Выбор начальной точки: фейеровское отображение



$$\pi_i(x) = x - \frac{\langle a_i, x \rangle - b_i}{\|a_i\|^2} a_i$$

$$\psi(x) = \begin{cases} x, & \text{если } x \in M \\ \frac{1}{|\mathcal{J}_x|} \sum_{i \in \mathcal{J}_x} \pi_i(x), & \text{если } x \notin M \end{cases}$$

Построение локального образа



$$\gamma_i(x) = x - \frac{\langle a_i, x \rangle - b_i}{\langle a_i, c \rangle} c$$

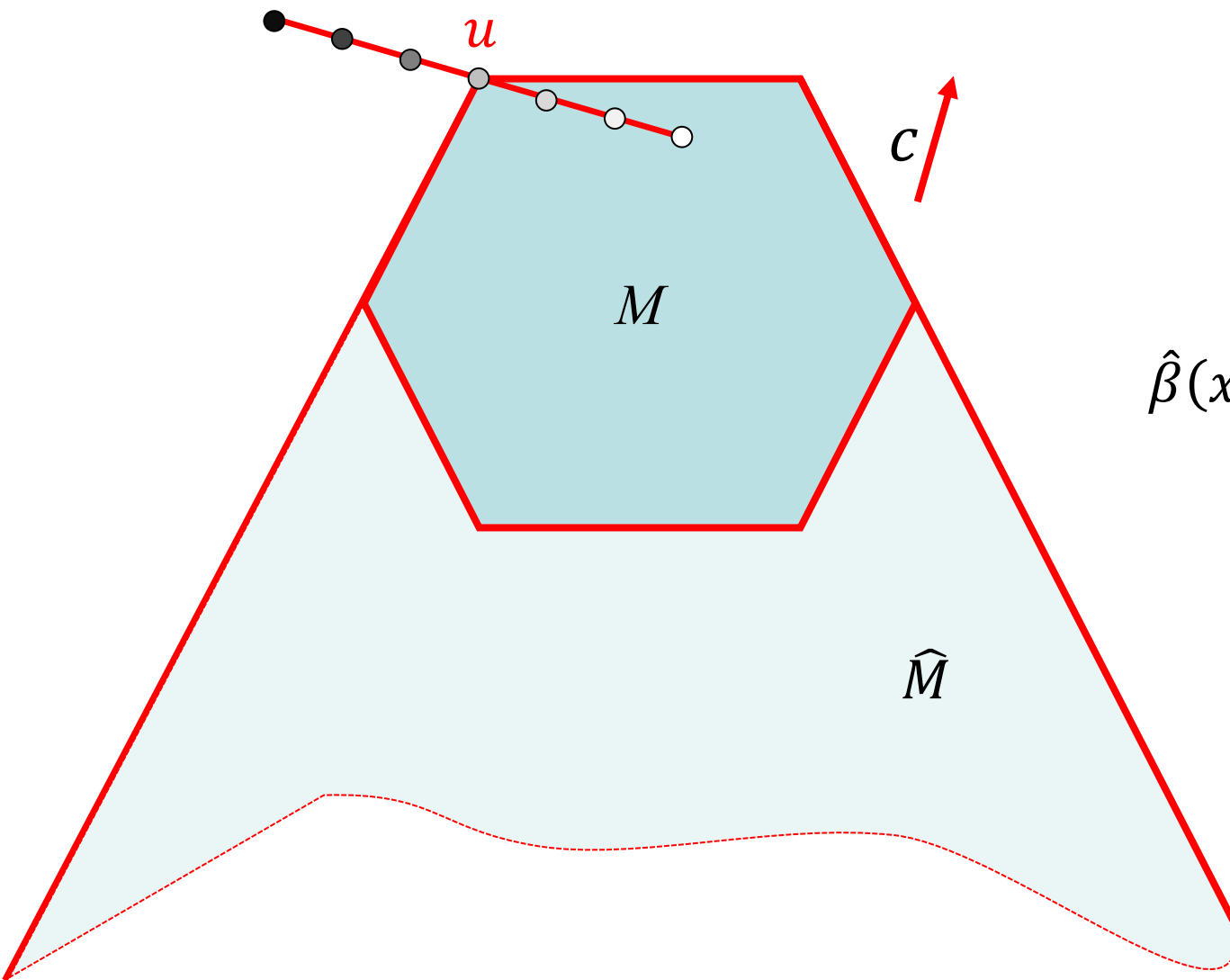
$$\beta_i(x) = -\frac{\langle a_i, x \rangle - b_i}{\langle a_i, c \rangle} \|c\|$$

$$i' = \operatorname{argmin}\{\beta_i(x) \mid i \in \mathcal{I}\}$$

$$\hat{\gamma}(x) = \gamma_{i'}(x)$$

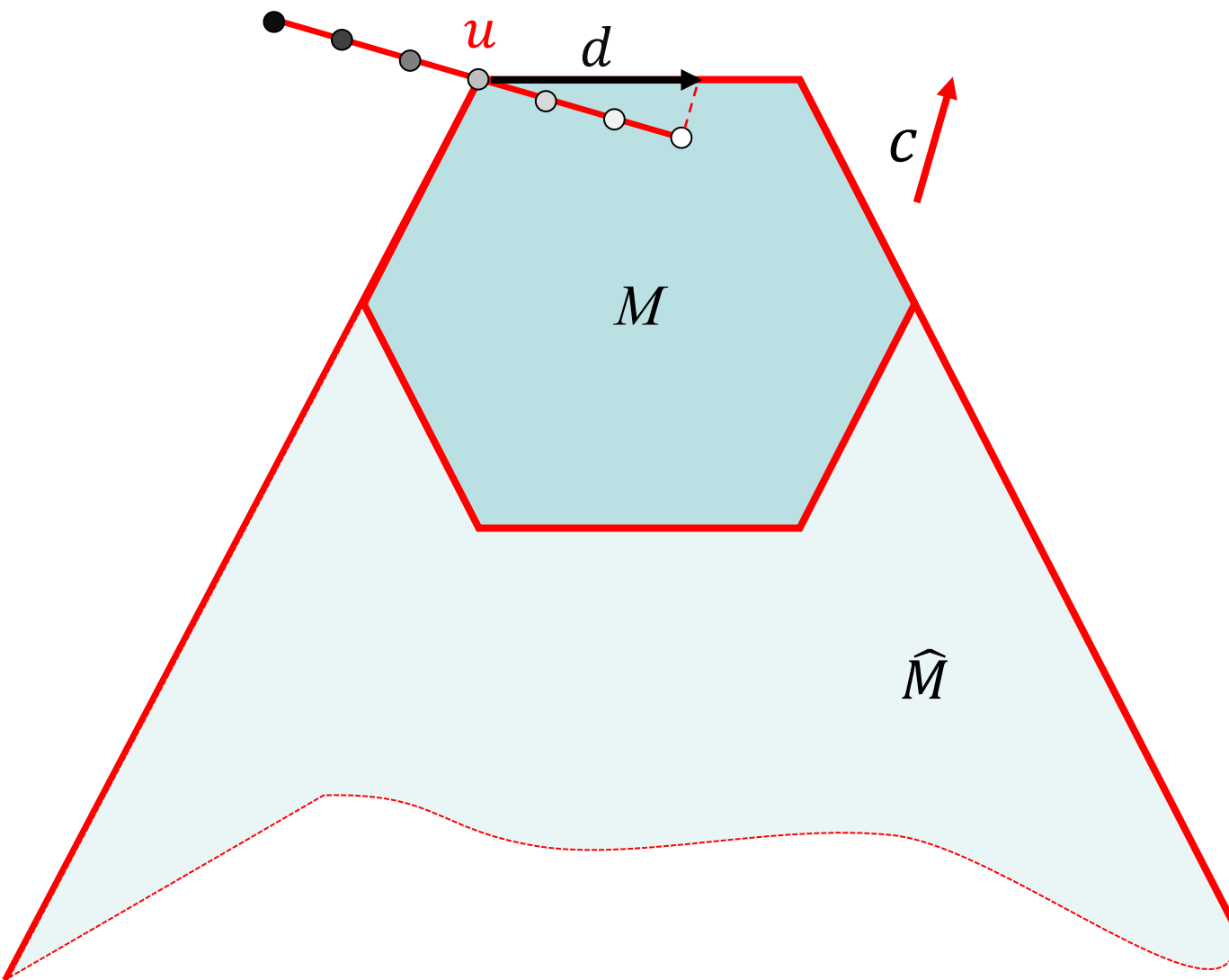
$$\hat{\beta}(x) = \frac{\langle c, \hat{\gamma}(x) - x \rangle}{\|c\|^2}$$

Локальный образ

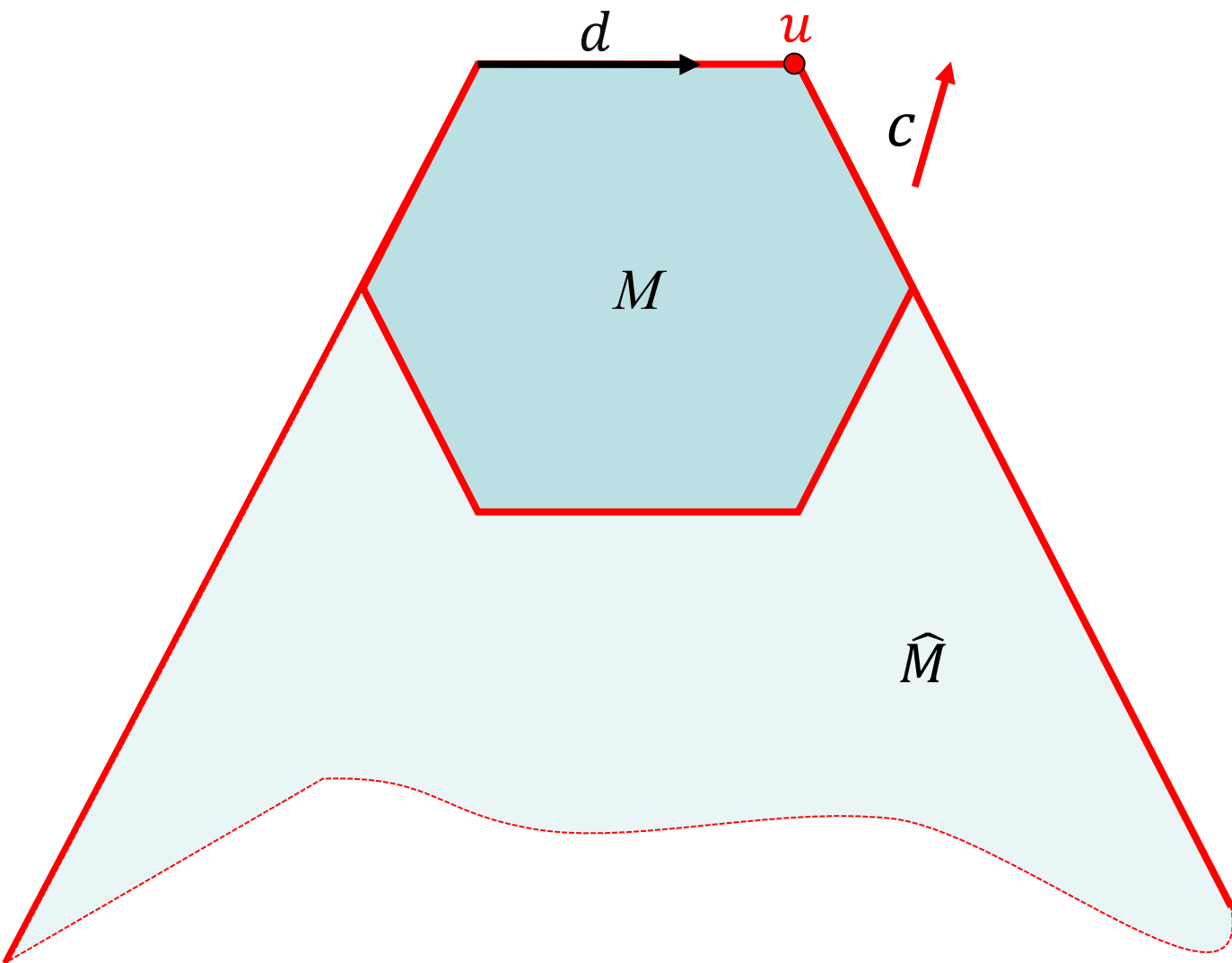


$$\hat{\beta}(x) = \frac{\langle c, \hat{\gamma}(x) - x \rangle}{\|c\|^2}$$

Нейронная сеть вычисляет направление движения



Получаем следующее приближение



Обучающее множество

- Обучающее множество может быть построено с помощью апекс-метода

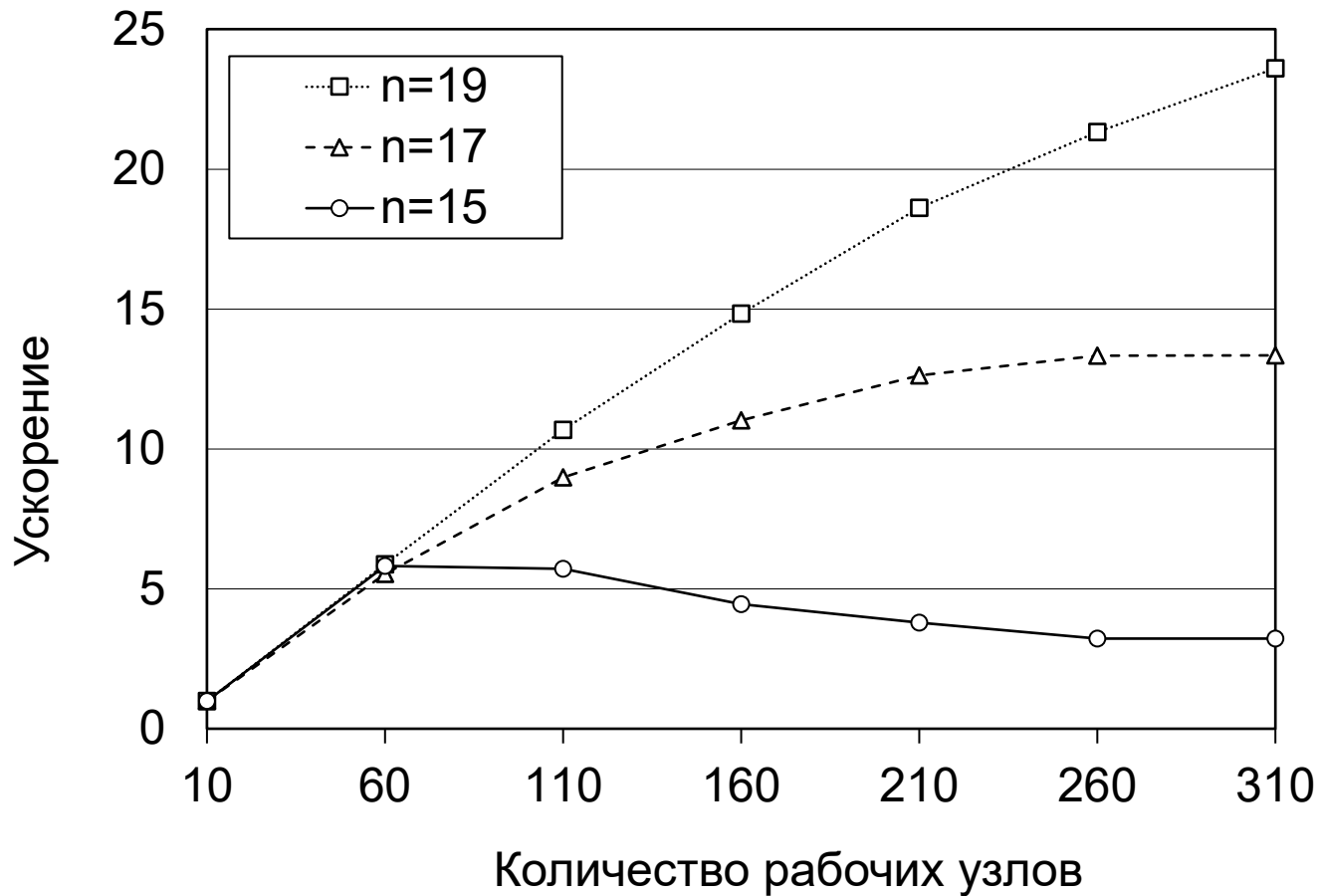
[Соколинский Л.Б., Соколинская И.М. О новой версии апекс-метода для решения задач линейного программирования // Вестник Южно-Уральского государственного университета. Серия: Вычислительная математика и информатика. 2023. Т. 12, № 2. С. 5–46. DOI: <https://doi.org/10.14529/cmse230201>]

Суперкомпьютер «Торнадо ЮУрГУ»

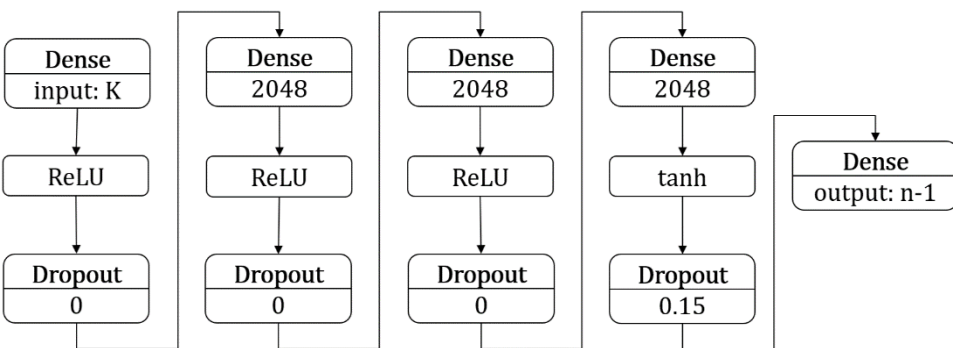


Количество узлов:	384
Тип процессоров:	2 x Intel Xeon X5680 (12 ядер по 3.33 ГГц; 2 потока на ядро)
Оперативная память узла:	24 Гб
Тип сопроцессора:	Intel Xeon Phi SE10X: (61 ядро по 1.1 ГГц; 4 потока на ядро)
Память сопроцессора:	8 Гб
Тип системной сети:	InfiniBand QDR
Тип управляющей сети:	Gigabit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

Построение локального образа задачи ЛП



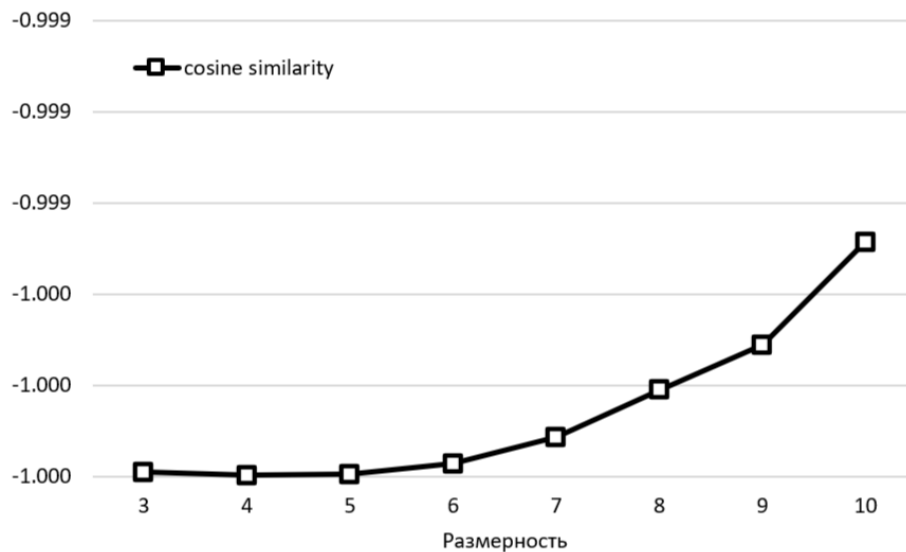
Нейронная сеть



NVIDIA Tesla V100

$$\frac{\sum_{k=1}^n (\alpha_k \cdot y_k)}{\sqrt{\sum_{k=1}^n \alpha_k^2} \cdot \sqrt{\sum_{k=1}^n y_k^2}}$$

Cosine Similarity



Ольховский Н. А. 2023. Исследование структуры рецептивного поля в визуальном методе решения задачи линейного программирования. PREPRINTS.RU. <https://doi.org/10.24108/preprints-3112771>

Спасибо за внимание!