

# Анализ реализаций симплекс-метода в решателях с открытым кодом

Федоренко Екатерина

Научный руководитель: Роланд Хильдебранд

Команда: Екатерина Федоренко, Денис Лейбман

Московский физико-технический институт

*fedorenko.es@phystech.edu*

21 апреля 2024 г.

# Содержание

## 1 Немного про решаемую задачу

- Симплекс-метод

## 2 Найденные решения

- Вычисления
- Остановка и зацикливание
- CHUZC

## 3 Литература

## 4 Цели

# Напоминание

## Стандартная форма линейной программы

$$\min_{x \geq 0} c^T x : Ax = b, \quad (1)$$

где  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$ ,  $c \in \mathbb{R}^n$ ,  $0 < \text{rk } A = m < n$ .

В процессе работы симплекс-метода поддерживаются два дизъюнктных множества индексов столбцов  $B \subset [n]$ ,  $|B| = m$  и  $N = [n] \setminus B$  – базисные и небазисные индексы, а также невырожденная подматрица  $A_{*B}$  – матрица базисных столбцов.

# Итерация прямого симплекс-метода

- ① CHUZC: выбрать некоторым способом индекс  $q$  в строке  $\hat{c}_N$  текущих стоимостей – входящий элемент.
- ② FTRAN: получить опорный  $q$ -ый столбец  $\hat{A}_{*q} = A_{*B}^{-1} A_{*q}$ .
- ③ CHUZR: выбрать некоторым образом индекс строки  $p$  – выходящий элемент, положить  $\alpha = \hat{b}_p / \hat{A}_{pq}$ . Обновляем столбец  $\hat{b}$ :  $\hat{b} \leftarrow \hat{b} - \alpha \hat{A}_{*q}$ .
- ④ BTRAN: получить строку  $\pi_p^T = e_p^T A_{*B}^{-1}$ .
- ⑤ PRICE: получить опорную  $p$ -ю строку  $\hat{A}_{p*} = \pi_p^T A_{*N}$ . Обновить стоимость:  $\hat{c}^T \leftarrow \hat{c}^T - \hat{c}_q \hat{a}_p^T$ .
- ⑥ Либо заново провести LU-разложение для  $A_{*B}^{-1}$ , либо обновить текущее разложение, дописав новую матрицу в список факторов.

# Изучаемый решатель



# Вычисления

$B^{-1}$  : INVERT

$$B = LU$$

LU - разложение выполняется с помощью метода Марковеца для разреженных матриц.

$B^{-1}$  : UPDATE

$$\bar{B} = B + (a_q - Be_p)e_p^T = B(I + (\hat{a}_q - e_p)e_p^T) = BE$$

E - это матрица, которая будет легко обратима

Теперь при обновлении матрицы B(замена столбца) надо будет обратить только E

# Двойственное вырождение

- Если некоторые небазисные двойственные значения  $c_N^T - c_B^T B^{-1} N$  нулевые, то вершина является двойственным вырожденной
- В этой вершине итерация двойственного симплекс метода может не привести к увеличению двойственного аргумента.
- Так может произойти зацикливание или остановка

## Изменение стоимости

- Добавим небольшие значения к некоторым значениям коэффициентам с.
- Тогда в худшем случае небазисные двойственные значения принимают небольшие положительные значения.
- Итерация двойственного симплекс метода будет хотя бы на немного увеличивать значение
- При достижении оптимума нужно убрать добавленные значения
- После этого могут потребоваться дополнительные итерации так как потеряна оптимальность

## CHUZC разреженность

Даже для задач, у которых матрица является гипер разреженной, вектор базисных стоимостей может быть полным. Вектор, приведенных затрат, заданный формулой:

$$\hat{c}_N^T = c_N^T - c_B^T B^{-1} N$$

тоже может быть полным.

Только если строка  $e_p^T B^{-1} N$  разрежена, количество стоимостей, которые изменяются на каждой итерации невелико. Тогда можем повысить эффективность CHUZC

# Эффективный CHUZC

Будем поддерживать набор набор  $C$ , состоящий из небольшого числа переменных  $s$ , который гарантированно включает переменную с наибольшей отрицательной приведенной стоимостью

В начале итерации инициализируем переменные:

- $g$  - самая отрицательная приведенная стоимость
- $h$  - самая низкая приведенная стоимость тех небазисных переменных, которых нет в  $C$ .

# Эффективный CHUZC

- Если  $h < g$ , то повторно инициализируем  $C$ , выполнив полный CHUZC: просмотреть все небазисные переменные и найти  $s + 1$  с наименьшими приведенными стоимостями. Сохранить значения с наименьшими приведенными затратами  $s$  в  $C$ .
- Найти и удалить переменную с наилучшей сниженной стоимостью из  $C$ .
- При обновлении приведенных затрат – сформировать набор  $D$  переменных (не входящих в  $C$ ) с наименьшими  $s$  приведенными затратами, которые изменились

# Эффективный CHUZC

- Обновлять  $h$  каждый раз, когда переменная, не включенная в  $D$ , имеет меньшую приведенную стоимость, чем текущее значение  $h$ .
- Обновить значение  $C$ , чтобы оно соответствовало наименьшим приведенным затратам  $s$  в  $C \cup D$ , установить значение  $g$  на наименьшую приведенную стоимость в результирующем  $C$  и обновить значение  $h$ , если переменная, которая не включена в  $C$ , имеет меньшую приведенную стоимость, чем текущее значение  $h$ .

# Литература



Rump, Siegfried (2005)

High precision evaluation of nonlinear functions



Uwe H. Suhl and Leena Suhl (1990)

Computing Sparse LU Factorizations for Large-Scale Linear Programming Bases

*Informs Journal on Computing* 2(4), 325 – 335.

## Текущие цели

- Разобрать оставшиеся решатели: Lpsolve и COIN-OR  
(Соответственно, Денис разбирает SCIP и GLPK).
- Составить итоговую сравнительную таблицу со списком решений проблем в разных решателях с их описаниями.