Предобработка СЦЛП с использованием почти параллельных строк

М. А. Симаков*

Работа посвящена новому методу предобработки задач СЦЛП, который уменьшает количество ненулевых элементов с помощью сокращения почти параллельных строк в матрице. Используется эвристика для эффективного поиска подходящих пар строк. Метод уменьшил количество ненулевых элементов в 30% задач MIPLIB 2017 Benchmark Set после предобработки HiGHS.

Ключевые слова: предобработка СЦЛП, дискретная оптимизация.

1. Введение

Предобработка является важным этапом решения задач СЦЛП.

$$z^\star = \min\{c^\top x \ : \ Ax = b,\ x \geqslant 0,\ x \in \mathbb{Z}^p \times \mathbb{R}^{d-p}\}, \quad p \in \{0, \ldots, d\} \ \text{(СЦЛП)}$$

Важным аспектом предобработки является уменьшение количества ненулевых элементов в матрице задачи ($\operatorname{nnz}(A)$) [2]. Одна из формулировок этой задачи представлена в [3]:

$$\operatorname*{argmin}_{\substack{T \in \mathbb{R}^{n \times n} \\ \det T \neq 0}} \operatorname{nnz}(TA) \tag{SP}$$

Задача SP является NP-сложной [4]. На практике применяются эвристические подходы. Например, можно выполнять последовательность сокращений $A_j \ \to \ A_j - \alpha A_i$. При хорошем выборе пары строк много ненулевых элементов сократится, и $\operatorname{nnz}(A)$ уменьшится. В работе [5] используются значения из пар столбцов для поиска хороших сокращений. Такой алгоритм генерирует большое количество кандидатов, и приходится искусственно ограничивать их список. Наш алгоритм использует большое подмножество столбцов для поиска хороших кандидатов.

* *Симаков Михаил Алексеевич* — студент ФПМИ МФТИ, e-mail: msimakov661@gmail.com, ORCID: 0009-0007-9412-343X.

Simakov Mikhail Alekseyevich — undergraduate student, Moscow Institute of Physics and Technology.

Опубликовано на условиях лицензии CC-BY 4.0
Published under CC-BY 4.0

2. Алгоритм

Далее $[k] = \{1, \ldots, k\}$, $A \in \mathbb{R}^{n \times d}$, $\operatorname{nnz}(A) = |\{(i,j) \,:\, A_{ij} \neq 0\}|$.

Выберем подмножество столбцов $C \subset [d]$, $\overline{C} = [d] \setminus C$. Назовём строки A_i, A_j почти параллельными относительно C, если $\exists\, \alpha \neq 0 \;:\; A_{iC} - \alpha A_{jC} = 0$. При замене $A_i x = b_i \;\to\; (A_i - \alpha A_j)x = b_i - \alpha b_j$, где $i \neq j$, решение задачи СЦЛП не изменится, а все ненулевые элементы в столбцах из C сократятся. Такие строки являются естественными кандидатами для сокращения.

Алгоритм 1 Поиск пар строк для сокращения

Вход: $A \in \mathbb{R}^{n \times d}$, разбиение столбцов на группы $C_1, \ldots, C_m$, $r \in [m]$
Выход: Кандидаты для сокращения

$a_i \leftarrow ()$, $\forall i \in [n]$
for all $\{k_1, \ldots, k_r\} \subset [m]$ **do**
  $C = \bigsqcup_{t=1}^{r} C_{k_t}$
  $\forall i$ α_i — первый ненулевой элемент в A_{iC} или 1, если $A_{iC} = 0$
  $A'_i \leftarrow A_i / \alpha_i$, $\forall i \in [n]$ ▷ нормируем элементы в столбцах C
  $G_0, \ldots, G_s$ — разбиение $\{i : a_i = ()\}$ по значениям A'_{iC}
  Пусть G_0 — группа строк таких, что $A_{iC} = 0$ $\forall i \in G_0$
  for all $g \in \{1, \ldots, s\}$ **do**
    $i^* \leftarrow \operatorname{argmin}\{\operatorname{nnz}(A_{i\overline{C}}) \,:\, i \in G_g\}$ ▷ эвристика
    for all $j \in G_g \setminus \{i^*\}$ **do**
      $a_j \leftarrow (i^*, \alpha_j / \alpha_{i^*})$ ▷ A_j и A_{i^*} – кандидаты для сокращения
    end for
  end for
end for
return a

При сокращении $A_j \;\to\; A_j - \alpha A_i$ изменение $\operatorname{nnz}(A)$ оценивается сверху: $\Delta = \operatorname{nnz}(A_j - \alpha A_i) - \operatorname{nnz}(A_j) \leqslant \operatorname{nnz}(A_{i\overline{C}}) - \operatorname{nnz}(A_{jC})$.

В каждой группе почти параллельных строк G_g алгоритм 1 выбирает одну строку $i^\star$, минимизирующую оценку на Δ, и формирует кандидатов на сокращение с ней. В итоге для каждой строки j алгоритм возвращает не более одного кандидата для сокращения. После этого предложенные сокращения, уменьшающие $\operatorname{nnz}(A)$, применяются одновременно. Алгоритм 1 имеет асимптотику $O(\binom{m}{r}(\operatorname{nnz}(A) + n \log n + mn))$ при использовании хеширования для разбиения на G_g.

Наша формулировка СЦЛП содержит только равенства в ограничениях. На практике задачи СЦЛП часто содержат неравенства. В таком случае кандидат для сокращения $i^\star$ выбирается среди равенств.

3. Результаты

В экспериментах использован вариант алгоритма 1 с независимой нормировкой строк в каждой группе столбцов C_i, без согласования масштабов между группами. Реализация на C++ была применена к 240 задачам MIPLIB 2017 Benchmark Set [1] после предобработки HiGHS v1.7.2 со стандартными параметрами [6]. Использовался 8 ядерный процессор Apple M1 с 8ГБ RAM, $m = 4$, $r = 2$ и разбиение столбцов $C_k = \{i : i \equiv k \mod m\}$, $k \in [m]$.

Даже после предобработки HiGHS, включающей уменьшение $\operatorname{nnz}(A)$, предложенный алгоритм дополнительно уменьшил $\operatorname{nnz}(A)$ в 30% задач. На этих задачах $\operatorname{nnz}(A)$ в среднем упал на 2%. На 4 задачах уменьшение составило больше 10%. На задаче `neos-5093327-huahum` число ненулевых элементов уменьшилось на 31.75% за 8 мс дополнительной предобработки.

Время работы на всех задачах не превышало $10^{-6} \operatorname{nnz}(A)$ секунд.

Список литературы

[1] A. Gleixner, G. Hendel, G. Gamrath, T. Achterberg, M. Bastubbe, T. Berthold, P. M. Christophel, K. Jarck, T. Koch, J. Linderoth, M. Lübbecke, H. D. Mittelmann, D. Ozyurt, T. K. Ralphs, D. Salvagnin, Y. Shinano, "MIPLIB 2017: Data-Driven Compilation of the 6th Mixed-Integer Programming Library", *Mathematical Programming Computation*, **13** (2021), 443–490. DOI: 10.1007/s12532-020-00194-3.

[2] T. Achterberg, R. E. Bixby, Z. Gu, E. Rothberg, D. Weninger, "Presolve Reductions in Mixed Integer Programming", *INFORMS Journal on Computing*, **32**:2 (2020), 473–506. DOI: 10.1287/ijoc.2018.0857.

[3] S. F. Chang, S. T. McCormick, "A hierarchical algorithm for making sparse matrices sparser", *Mathematical Programming*, **56** (1992), 1–30. DOI: 10.1007/BF01580890.

[4] Alan J. Hoffman, S. T. McCormick, "A Fast Algorithm that makes Matrices Optimally Sparse", *Progress in Combinatorial Optimization*, eds. WILLIAM R. PULLEYBLANK, Academic Press, 1984, 185–196. DOI: 10.1016/B978-0-12-566780-7.50017-9.

[5] P. Gemander, W.-K. Chen, D. Weninger, L. Gottwald, A. Gleixner, A. Martin, "Two-row and two-column mixed-integer presolve using hashing-based pairing methods", *EURO Journal on Computational Optimization*, **8**:3–4 (2020), 205–240. DOI: 10.1007/s13675-020-00129-6.

[6] Q. Huangfu, J. A. J. Hall, “Parallelizing the dual revised simplex method”, *Mathematical Programming Computation*, **10**:1 (2018), 119–142. DOI: 10.1007/s12532-017-0130-5 .

Статья поступила 9 ноября 2026 г.

MILP presolve using near-parallel rows

M. A. Simakov

We propose a new MILP presolve method that reduces nonzeros using near-parallel rows in the constraint matrix. A heuristic efficiently identifies promising row pairs. The method reduced nonzeros in 30% of instances in the MIPLIB 2017 Benchmark Set after HiGHS presolve.

Keywords: MILP presolve, discrete optimization.

References

[1] A. Gleixner, G. Hendel, G. Gamrath, T. Achterberg, M. Bastubbe, T. Berthold, P. M. Christophel, K. Jarck, T. Koch, J. Linderoth, M. Lübbecke, H. D. Mittelmann, D. Ozyurt, T. K. Ralphs, D. Salvagnin, Y. Shinano, “MIPLIB 2017: Data-Driven Compilation of the 6th Mixed-Integer Programming Library”, *Mathematical Programming Computation*, **13** (2021), 443–490. DOI: 10.1007/s12532-020-00194-3.

[2] T. Achterberg, R. E. Bixby, Z. Gu, E. Rothberg, D. Weninger, “Presolve Reductions in Mixed Integer Programming”, *INFORMS Journal on Computing*, **32**:2 (2020), 473–506. DOI: 10.1287/ijoc.2018.0857.

[3] S. F. Chang, S. T. McCormick, “A hierarchical algorithm for making sparse matrices sparser”, *Mathematical Programming*, **56** (1992), 1–30. DOI: 10.1007/BF01580890.

[4] Alan J. Hoffman, S. T. McCormick, “A Fast Algorithm that makes Matrices Optimally Sparse”, *Progress in Combinatorial Optimization*, eds. WILLIAM R. PULLEYBLANK, Academic Press, 1984, 185–196. DOI: 10.1016/B978-0-12-566780-7.50017-9.

[5] P. Gemander, W.-K. Chen, D. Weninger, L. Gottwald, A. Gleixner, A. Martin, “Two-row and two-column mixed-integer presolve using hashing-based pairing methods”, *EURO Journal on Computational Optimization*, **8**:3–4 (2020), 205–240. DOI: 10.1007/s13675-020-00129-6.

[6] Q. Huangfu, J. A. J. Hall, “Parallelizing the dual revised simplex method”, *Mathematical Programming Computation*, **10**:1 (2018), 119–142. DOI: 10.1007/s12532-017-0130-5.

Received on November 9, 2026