


Нижегородский государственный университет им. Н.И. Лобачевского

Институт информационных технологий, математики и механики

Нижегородский государственный университет им. Н.И. Лобачевского

Институт информационных технологий, математики и механики

ПАРАЛЛЕЛЬНАЯ ОБРАБОТКА ГРАФОВ

Лекция 5-2. Переупорядочение разреженных матриц

Пирова А.Ю.
Кафедра ВВиСП

Решение СЛАУ с разреженной матрицей

- При выполнении численного моделирования конечно-элементными методами часто возникает задача решения системы линейных алгебраических уравнений (СЛАУ) с разреженной матрицей большого порядка.

- Пусть дана система линейных уравнений:

$$Ax = b$$

A – симметричная положительно определенная разреженная матрица,
b – плотный вектор, x – вектор неизвестных.

- Прямой метод решения (разложение Холецкого):

$$A = LL^T$$

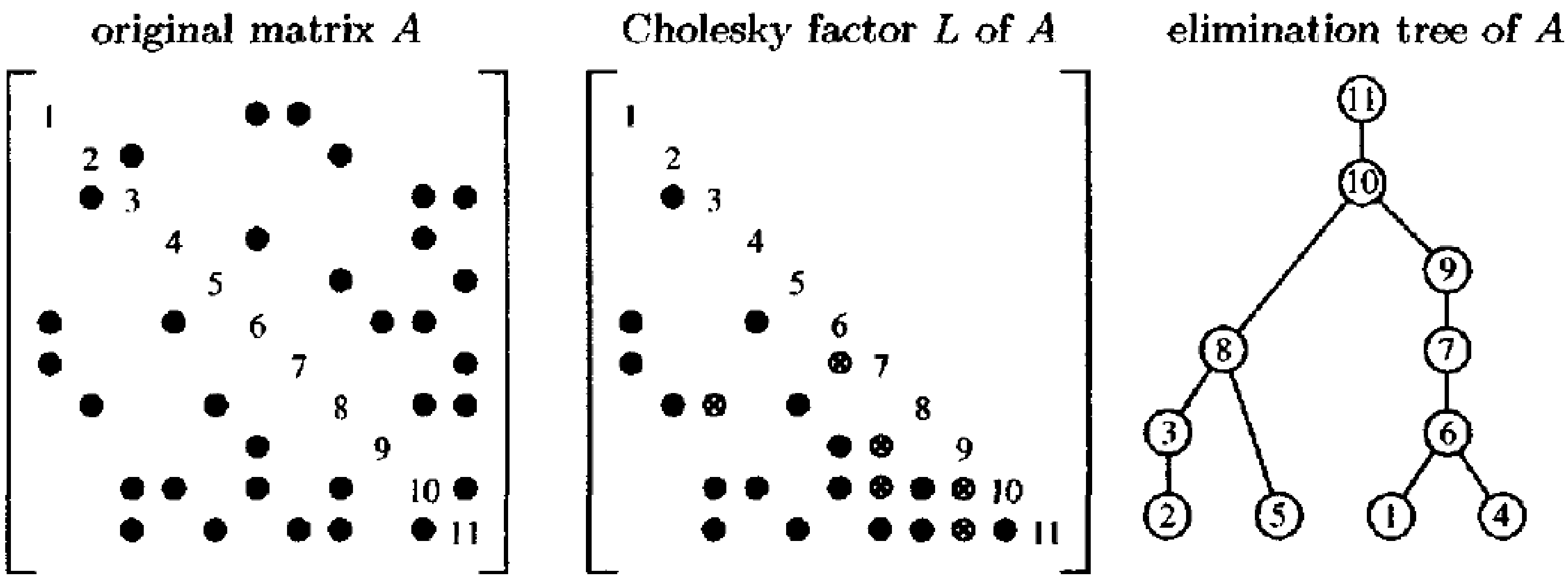
L – нижнетреугольная матрица, называемая *фактором* матрицы A

- Переход к решению треугольных систем

$$LDy = b, L^T x = y$$

Решение СЛАУ с разреженной матрицей. Заполнение

- При прямом решении СЛАУ с разреженной матрицей возникает проблема ***заполнения*** – существенного увеличения числа ненулевых элементов фактора матрицы.



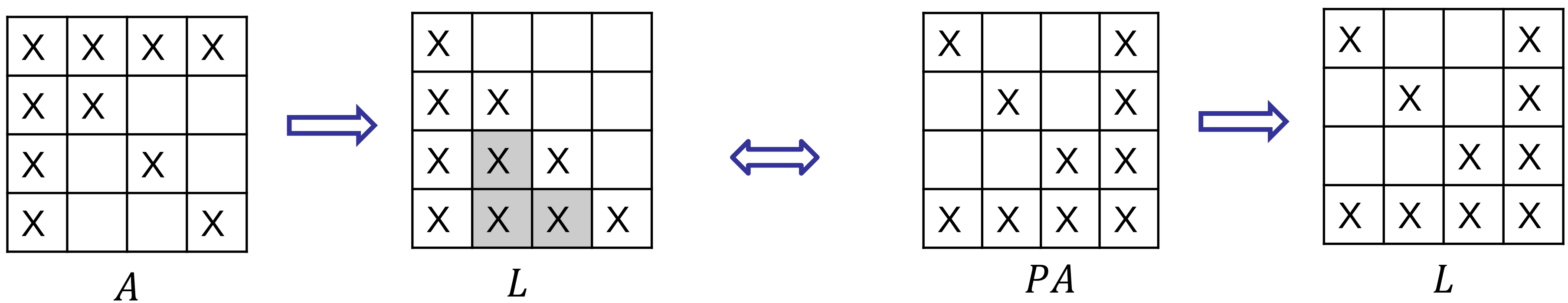
Теорема. Для разложения Холецкого $LL^T = A$ если $a_{ij} \neq 0$, то $l_{ij} \neq 0$.
Для $i < j < k$, если $l_{ji} \neq 0$ и $l_{ki} \neq 0$, то $l_{kj} \neq 0$.

Davis T. A. Direct methods for sparse linear systems. – SIAM, 2006. Pp. 38-41.

Решение СЛАУ с разреженной матрицей. Перестановка

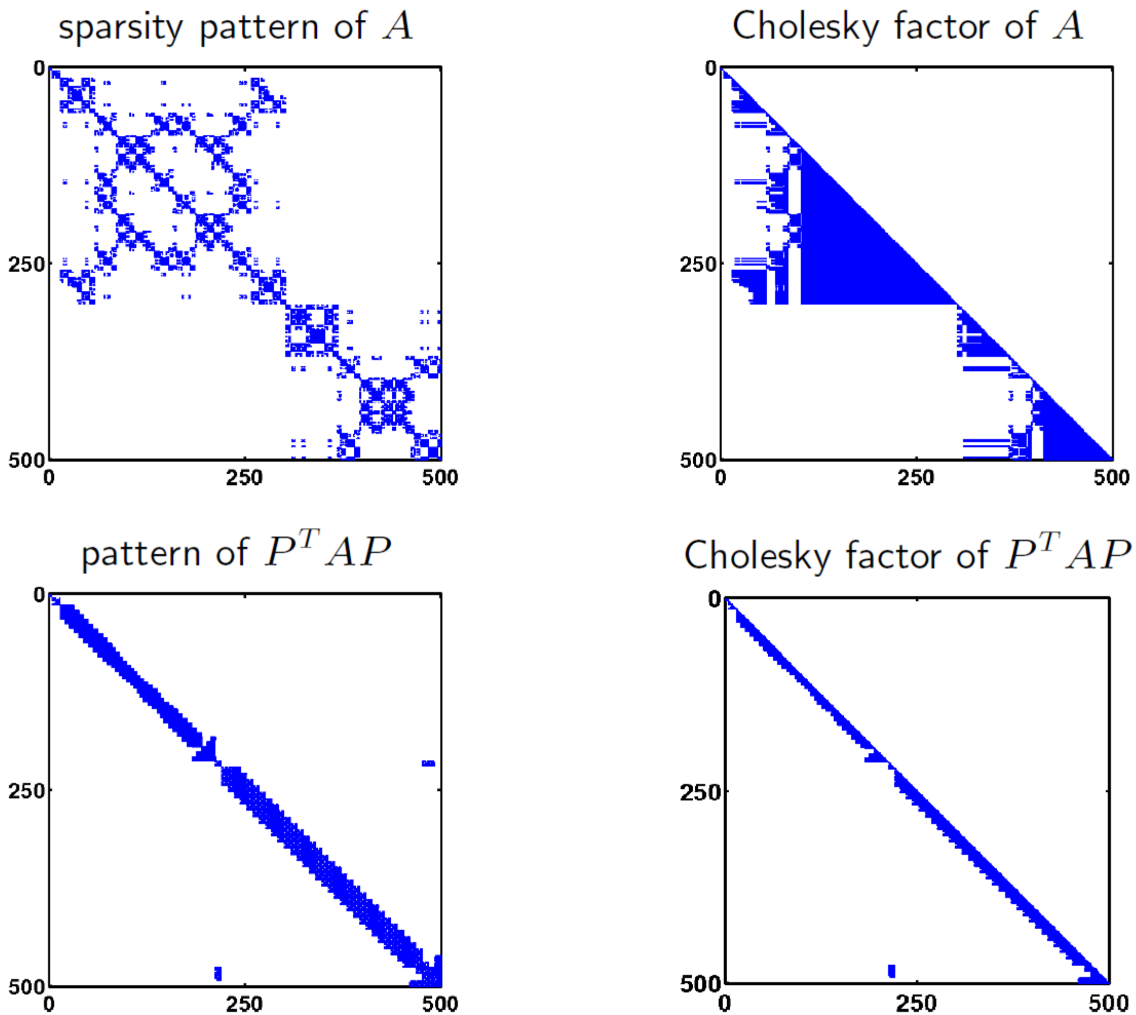
- Уменьшить заполнение можно с помощью симметричной перестановки:

$$Ax = b \iff (PAP^T)(Px) = Pb,\ P\text{– матрица перестановки}$$



- Задача нахождения оптимальной перестановки, минимизирующей заполнение фактора, **NP-трудная** (*Yannakakis*, 1981), решается эвристическими методами.

Пример



sparsity pattern of A
Cholesky factor of A
pattern of P^TAP
Cholesky factor of P^TAP
0
250
500

Решение СЛАУ с разреженной матрицей

- **Алгоритм решения** разреженной СЛАУ:
  1. Фаза анализа – предобработка матрицы, подготовка данных:
     1. Нахождение перестановки, минимизирующей размер фактора
        $$A' = PAP^T, b' = Pb$$
     2. Символьная факторизация – определение расположения ненулевых элементов в матрице L, выделение памяти.
  2. Численная фаза – численное разложение матрицы A', нахождение L;
  3. Обратный ход – решение треугольных систем
     $$Ly = Pb, L^T x = y$$
  4. Постобработка решения
     1. Обратная перестановка вектора x
     2. [Итерационное уточнение]

Алгоритмы и программные средства для переупорядочения разреженных матриц

Переупорядочение разреженной матрицы

Метод вложенных сечений (ND)

Комбинированные подходы

Метод минимальной степени (MD)

автоматические вложенные сечения

спектральный

координатный

многоуровневый

METIS / ParMETIS / mt-METIS,
Scotch / PT-Scotch,
коммерческий решатель Intel MKL Pardiso

множественный

приближенный

столбцовый приближенный

открытые решатели
SuiteSparse, MUMPS

- Открыта проблема рационального использования вычислительных ресурсов для параллельного метода вложенных сечений.

МЕТОД МИНИМАЛЬНОЙ СТЕПЕНИ

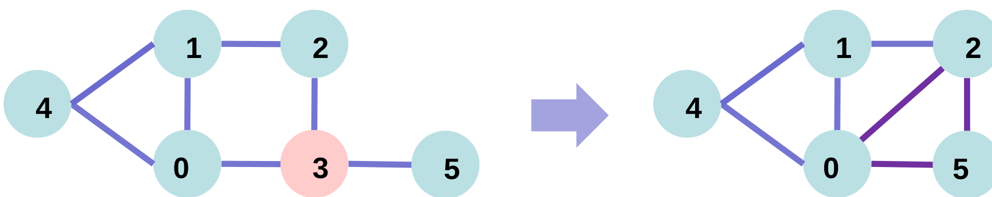
Метод минимальной степени

- Моделируем процесс исключения Гаусса
- *Исключение* вершины v из графа:
  - Вершина исключается вместе с исходящими из нее ребрами,
  - Множество смежных с v вершин становится кликой:

$$G' = (V\backslash v, E\backslash \mathrm{E}_v \cup F), F = \{(u,w): u,w \in Adj(v), u \neq w\}$$

$Adj(v)$ – множество вершин, смежных вершине v

E_v – множество ребер, исходящих из вершины v



Метод минимальной степени

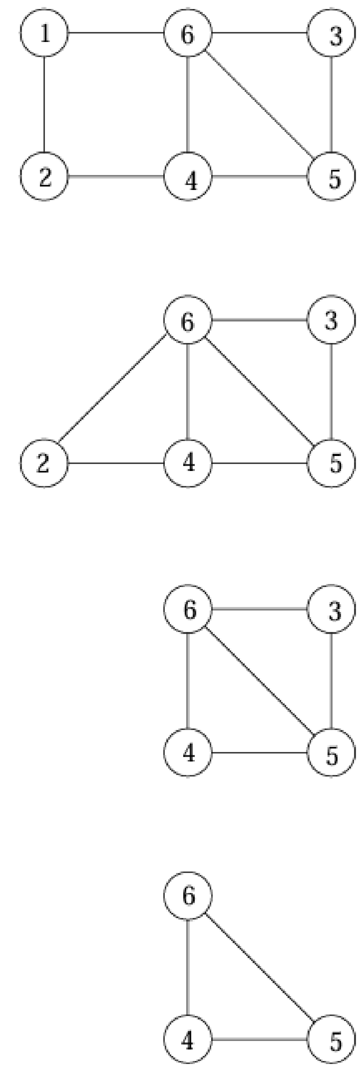
- Авторы: Тинней, Уолкер (1969 г.), модификации Джордж, Лю (1980 г.), Лю (1985 г.), Амстей, Дэвис и Дафф (1996 г.), Дэвис (2004 г.)

- Базовый алгоритм:

Пока не перенумерованы все вершины графа G:

1. В текущем графе G выбрать вершину с минимальной степенью v_i и присвоить ей очередной номер. Если таких вершин несколько, выбрать v_i произвольно.
2. Исключить вершину v_i из графа G. Обновить граф.

Вычислительная сложность $O(|V|^2|E|)$



Модификации метода минимальной степени

- **Множественный** метод минимальной степени, Лю, 1986 г.:
  - Одновременное исключение вершин с одинаковой степенью,
  - Неполное обновление степеней (incomplete degree update). Если выполняется соотношение: $Adj(u) \cup \{u\} \subset Adj(v) \cup \{v\}$, то степень вершины v не нужно вычислять, пока не будет исключена вершина u.

  Вычислительная сложность $O(|V|^2|E|)$
- **Приближенный** метод минимальной степени, Амстой и др., 1996 г.:
  - Оценка степени вершины вместо точного вычисления,
  - Одновременное исключение вершин с одинаковыми списками смежности

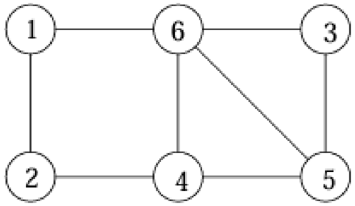
  Вычислительная сложность $O(|V||E|)$

Приближенный метод минимальной степени

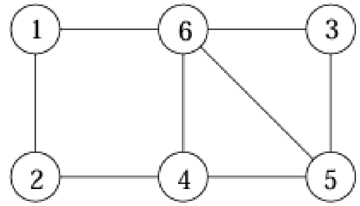
Термины:

- *Граф исключения* G_k – граф после исключения k вершин.
- *Фактор-граф (quotient graph)* $\mathcal{G}_k = (V^k, \overline{V^k}, E^k, \overline{E^k})$ – граф, в котором хранится информация об исключенных и не исключенных вершинах:
  V^k - неисключенные вершины графа G_k,
  $\overline{V^k}$ - исключенные вершины (супервершины),
  E^k - ребра между вершинами из V^k,
  $\overline{E^k}$ - ребра между вершинами из V^k и V^k.
- Достижимое множество $Reach(v)$ – множество вершин из V^k, в которые существует путь из v через вершины из V^k или $\overline{V^k}$

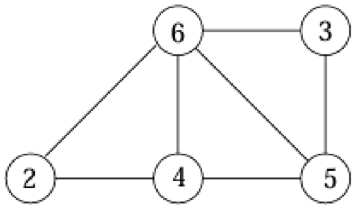
Пример



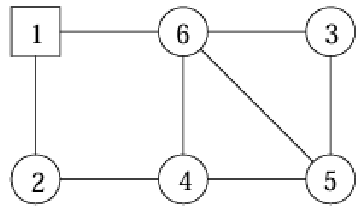
0



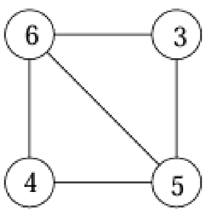
$reach\,(1) = \{2, 6\}$



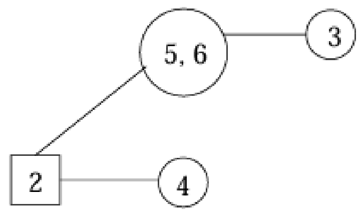
1



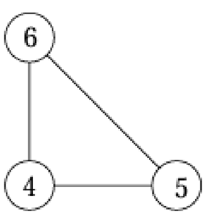
$reach\,(2) = \{4, 6\}$



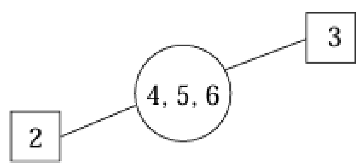
2



$reach\,(3) = \{5, 6\}$



3



$reach\,(4{,}5{,}6) = \{\}$

Приближенный метод минимальной степени

Обозначения:

- $\mathcal{A}_i$ - неисключенные вершины, смежные с неисключенной i
- $\mathcal{E}_i$ - исключенные вершины, смежные с неисключенной i

$$\mathcal{A}_i \equiv \{j : (i,j) \in E\} \subseteq V,$$

$$\mathcal{E}_i \equiv \{e : (i,e) \in \overline{E}\} \subseteq \overline{V},$$

$$\mathrm{Adj}_{\mathcal{G}}(i) \equiv \mathcal{A}_i \cup \mathcal{E}_i \subseteq V \cup \overline{V}.$$

- $\mathcal{L}_i$ - неисключенные вершины, смежные с исключенной e:

$$\mathcal{L}_e \equiv \mathrm{Adj}_{\mathcal{G}}(e) = \{i : (i,e) \in \overline{E}\} \subseteq V.$$

- *Внешняя степень* вершины используется вместо истинной степени:

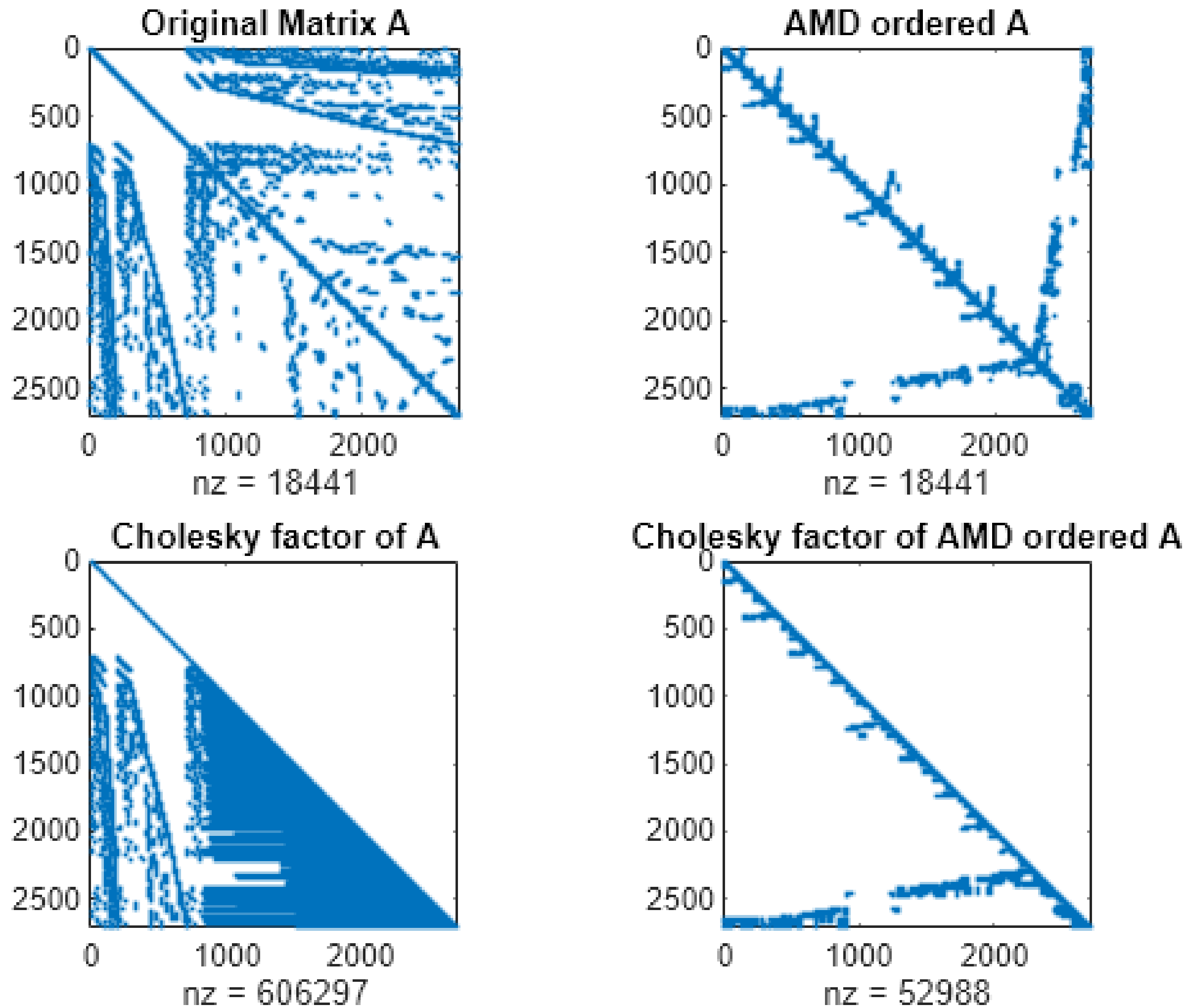
$$d_i = |\mathrm{Adj}_G(i) \setminus \mathbf{i}| = |\mathcal{A}_i \setminus \mathbf{i}| + \left| \left(\bigcup_{e \in \mathcal{E}_i} \mathcal{L}_e \right) \setminus \mathbf{i} \right|,$$

Приближенный метод минимальной степени

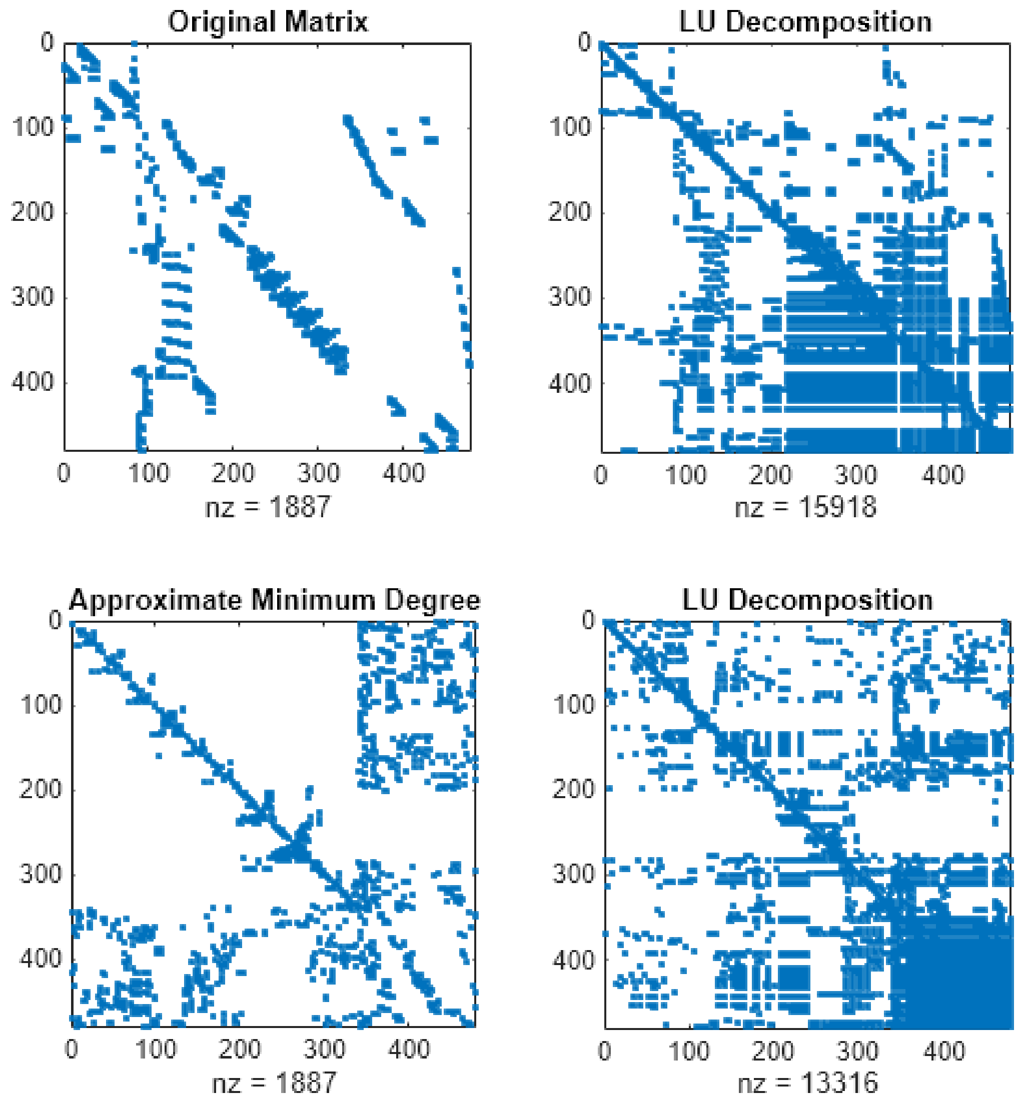
Algorithm 1 (Minimum degree algorithm, based on quotient graph)

$V = \{1...n\}$
$\overline{V} = \emptyset$
for $i = 1$ **to** n **do**
  $\mathcal{A}_i = \{j : a_{ij} \neq 0 \text{ and } i \neq j\}$
  $\mathcal{E}_i = \emptyset$
  $d_i = |\mathcal{A}_i|$
  $\mathbf{i} = \{i\}$
end for
$k = 1$
while $k \leq n$ **do**
  mass elimination:
  select variable $p \in V$ that minimizes d_p
  $\mathcal{L}_p = \left(\mathcal{A}_p \cup \bigcup_{e \in \mathcal{E}_p} \mathcal{L}_e\right) \setminus \mathbf{p}$
  for each $\mathbf{i} \in \mathcal{L}_p$ **do**
    remove redundant entries:
    $\mathcal{A}_i = (\mathcal{A}_i \setminus \mathcal{L}_p) \setminus \mathbf{p}$
    element absorption:
    $\mathcal{E}_i = (\mathcal{E}_i \setminus \mathcal{E}_p) \cup \{p\}$
    compute external degree:
    $d_i = |\mathcal{A}_i \setminus \mathbf{i}| + \left|\left(\bigcup_{e \in \mathcal{E}_i} \mathcal{L}_e\right) \setminus \mathbf{i}\right|$
  end for
  supervariable detection, pairs found via hash function:
  for each pair $\mathbf{i}$ and $\mathbf{j} \in \mathcal{L}_p$ **do**
    if $\mathbf{i}$ and $\mathbf{j}$ are indistinguishable **then**
      remove the supervariable $\mathbf{j}$*:*
      $\mathbf{i} = \mathbf{i} \cup \mathbf{j}$
      $d_i = d_i - |\mathbf{j}|$
      $V = V \setminus \{j\}$
      $\mathcal{A}_j = \emptyset$
      $\mathcal{E}_j = \emptyset$
    end if
  end for
  convert variable p to element p:
  $\overline{V} = (\overline{V} \cup \{p\}) \setminus \mathcal{E}_p$
  $V = V \setminus \{p\}$
  $\mathcal{A}_p = \emptyset$
  $\mathcal{E}_p = \emptyset$
  $k = k + |\mathbf{p}|$
end while

Примеры применения AMD перестановки

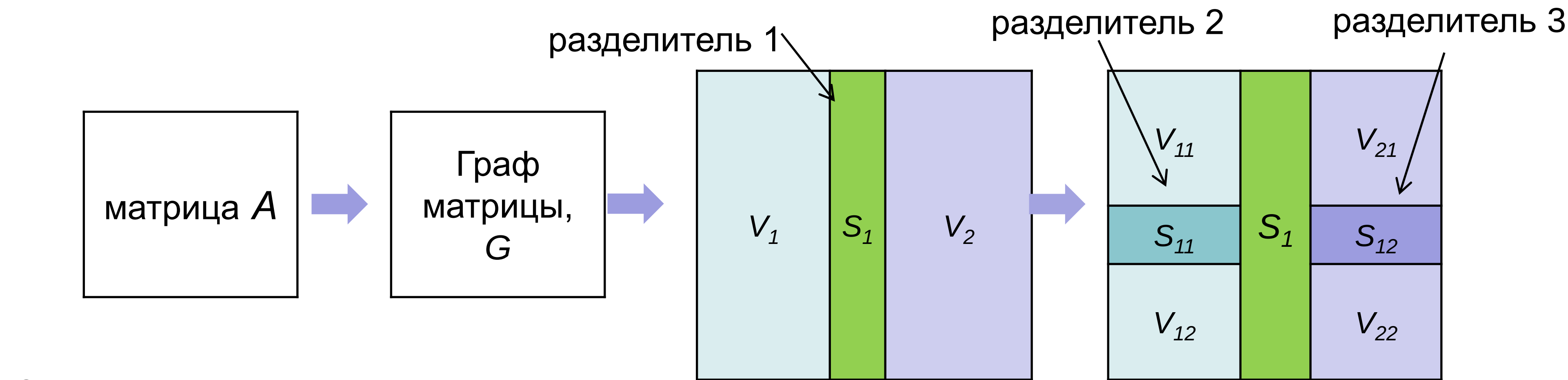


Примеры применения AMD перестановки

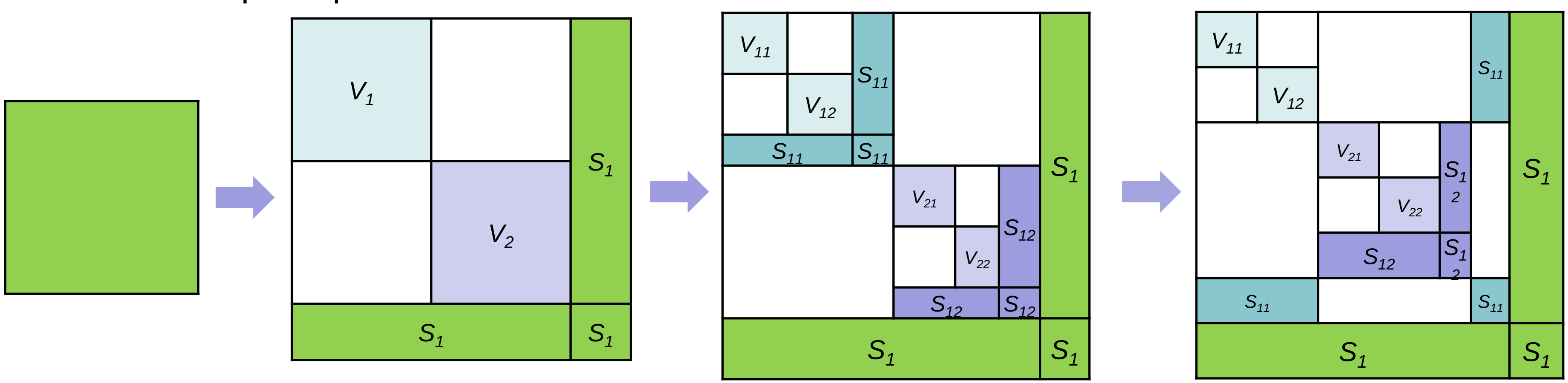


МЕТОД ВЛОЖЕННЫХ СЕЧЕНИЙ

Метод вложенных сечений (nested dissections, ND)



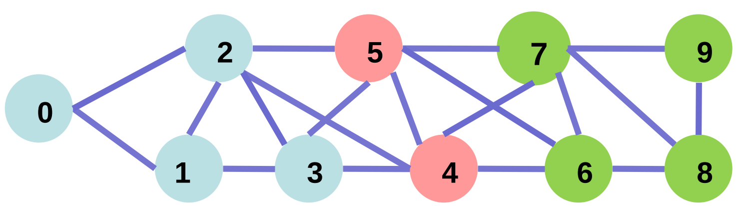
Заполнение фактора L:

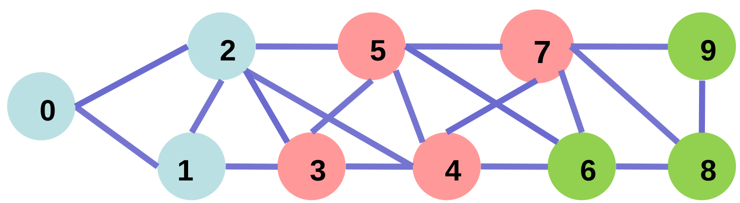


Вершинный разделитель графа

- *Вершинный разделитель связного графа* $G = (V, E)$ – подмножество его вершин $S \subset V$, после удаления которых вместе с инцидентными им ребрами граф $G(V \backslash S)$ становится несвязным.

$$V = S \cup V_1 \cup \cdots \cup V_k, \qquad S \cap V_i = \emptyset, V_i \cap V_j = \emptyset \; (i \neq j)$$





- Разделитель называется *минимальным*, если никакое его собственное подмножество не является разделителем графа.
- «Хороший» вершинный разделитель: разделитель минимален и дисбаланс минимален: $\|S\| \rightarrow min, \frac{\max(\|V_1\|,\|V_2\|)}{\min(\|V_1\|,\|V_2\|)} \rightarrow min$

Алгоритм метода вложенных сечений

Algorithm 2: Nested Dissection

`Input`: $G = (V, E)$: Graph, γ : minimum degree threshold

`Output`: π : node ordering

// compute 2-partition of G

$P = \{G_1 = (V_1, E_1), G_2 = (V_2, E_2)\} \leftarrow$ **PartitionGraph**(G)

// find node separator on G given the partitioning P

$(V_1', V_2', S) \leftarrow$ **ComputeSeparator**(G, P)

// continue recursively on G_1 and G_2

NestedDissection(G_1')

NestedDissection(G_2')

OrderSeparator(S, π) *// order S after G_1 and G_2*

Алгоритм гибридный: метод вложенных сечений + метод минимальной степени

Algorithm 2: Nested Dissection

Input: $G = (V, E)$: Graph, γ : minimum degree threshold

Output: π : node ordering

if $|V| \leq \gamma$ **then**

    MinimumDegree(G, π)

else

    // compute 2-partition of G

    $P = \{G_1 = (V_1, E_1), G_2 = (V_2, E_2)\} \leftarrow$ **PartitionGraph**(G)

    // find node separator on G given the partitioning P

    $(V_1', V_2', S) \leftarrow$ **ComputeSeparator**(G, P)

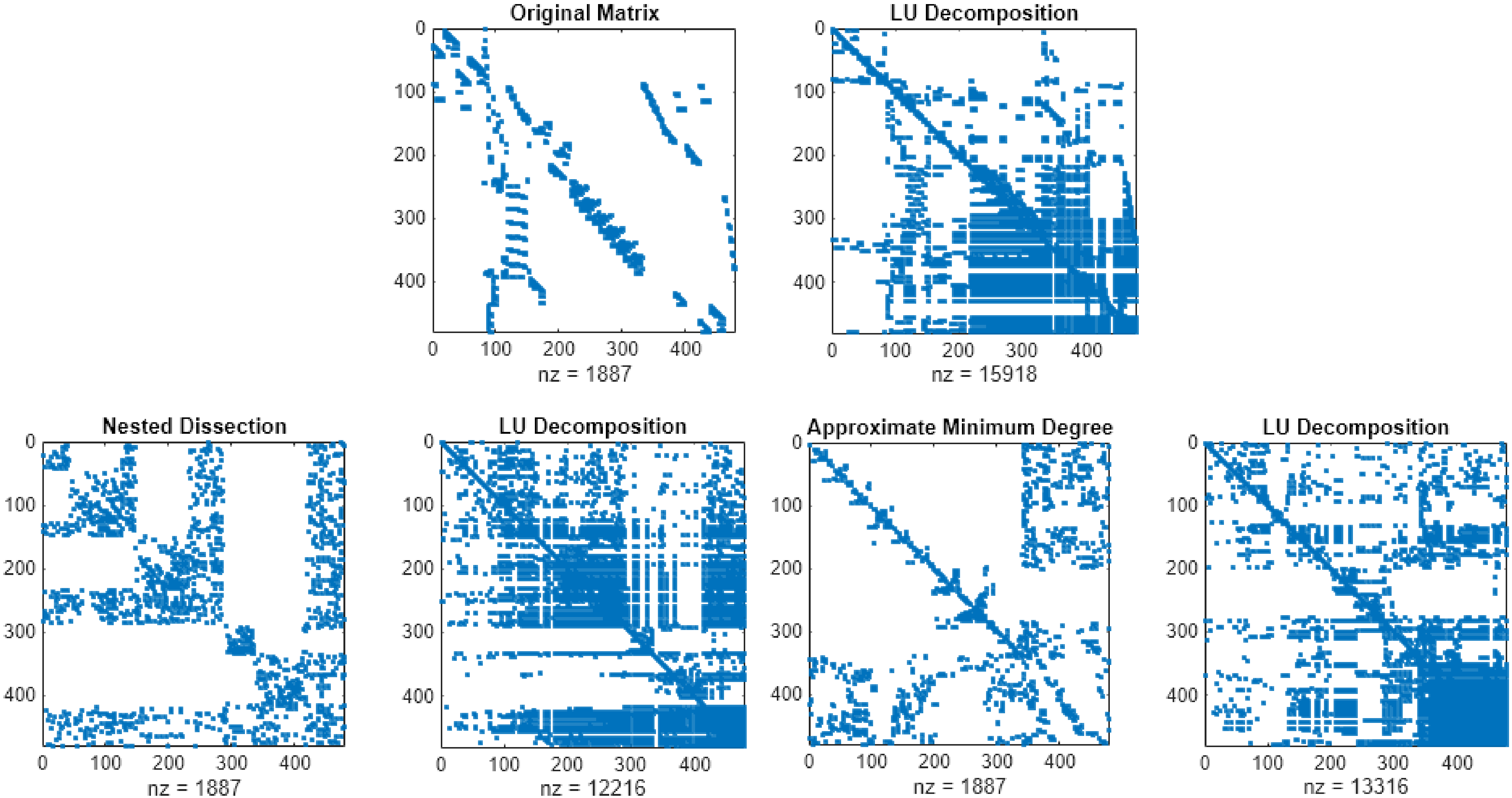
    // continue recursively on G_1 *and* G_2

    NestedDissection(G_1')

    NestedDissection(G_2')

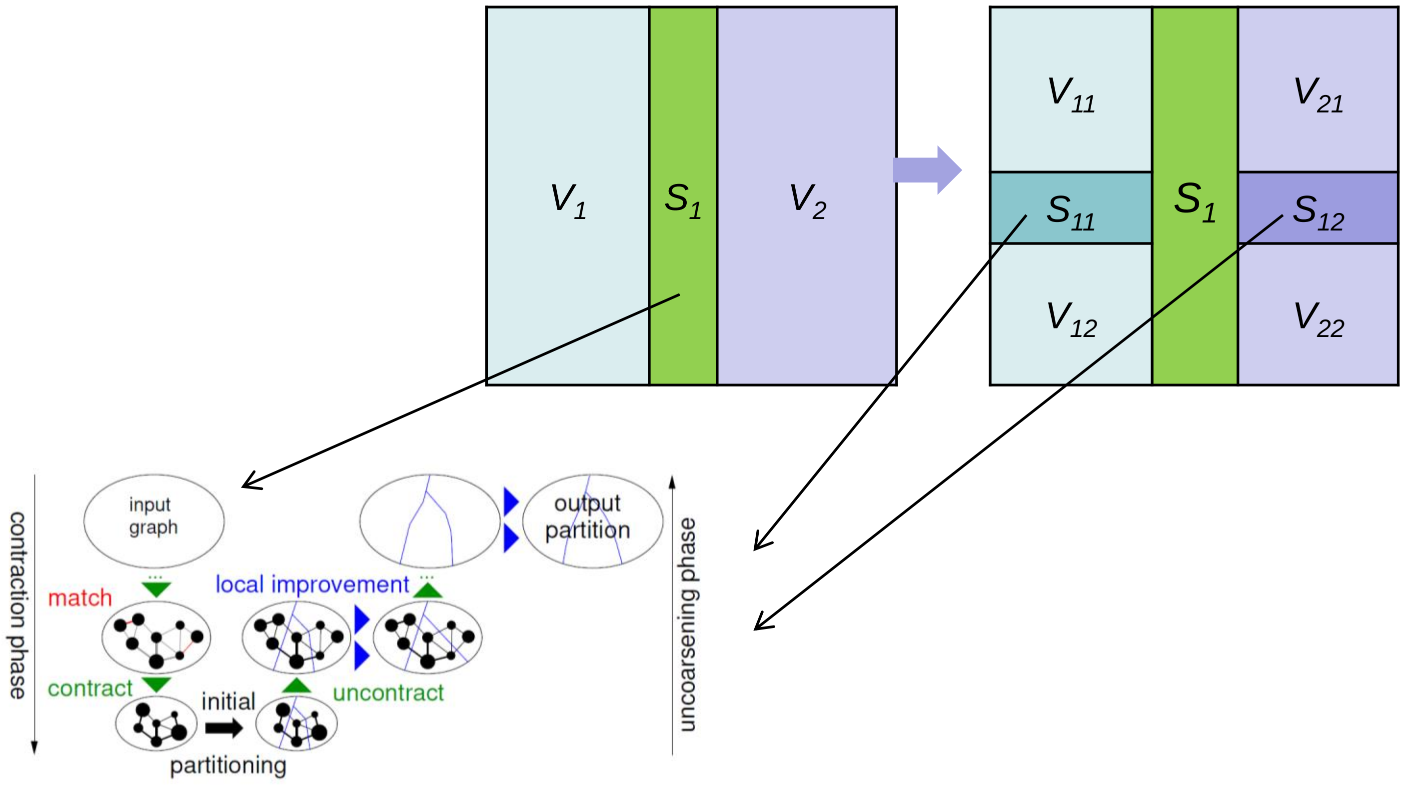
    OrderSeparator(S, π) *// order S after* G_1 *and* G_2

Примеры применения ND перестановки



Вершинный разделитель для ND

- Разделитель можно найти разными способами (спектральное разделение, геометрический метод и т.д.)
- Самый распространенный метод – **многоуровневый метод вложенных сечений**, аналог многоуровневого разделения графа.



Вершинный разделитель для ND

- Отличия от реберного разделения:
  1. Построение начального вершинного разделителя:
     - Выделить уровень дерева поиска в ширину
     - «Вершинная оболочка» реберного разделителя
  2. Улучшение разделения: модификация метода Федуччиа-Мэтьюса.
     1. Сравнение качества разделений:

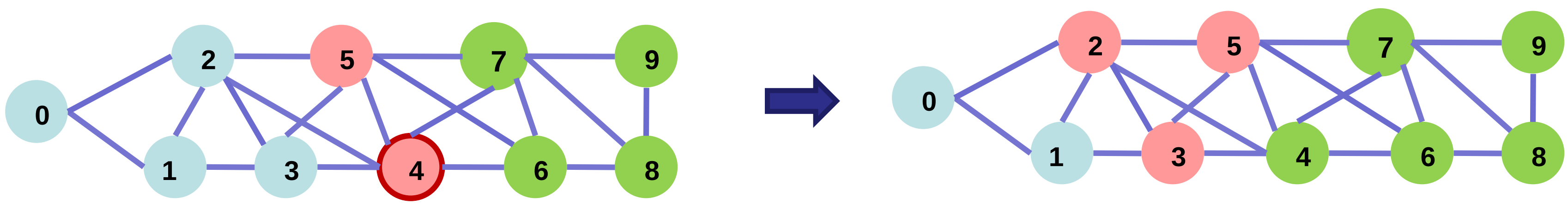
$$Eval(\boldsymbol{P}) = \|S\| * \left(\alpha + \beta * \frac{\max(\|V_1\|, \|V_2\|)}{\min(\|V_1\|, \|V_2\|)}\right)$$

     2. Приоритет перемещения вершины:

$$Gain(s \to V_i) = w(s) - \sum_{u \in Adj(s) \cap V_j} w(u)\,, j \neq i$$

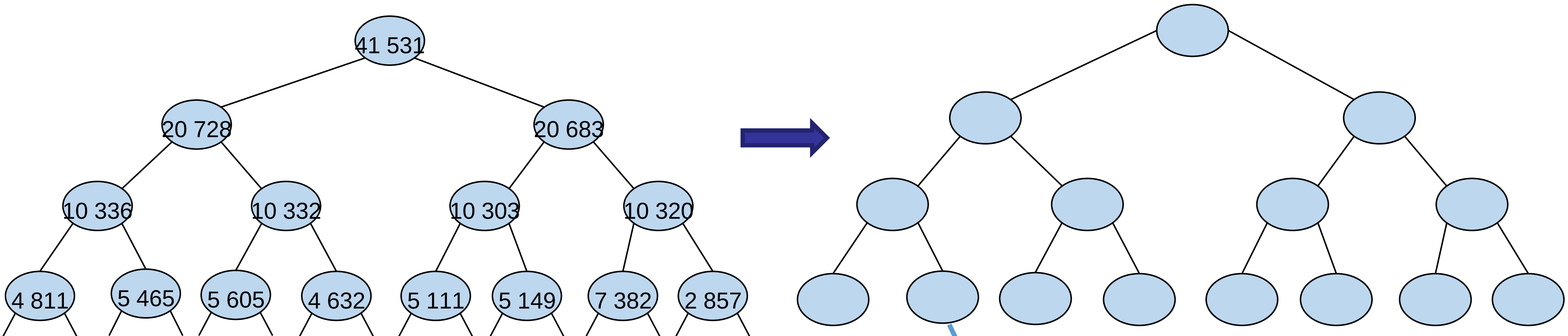
Улучшение разделения для ND

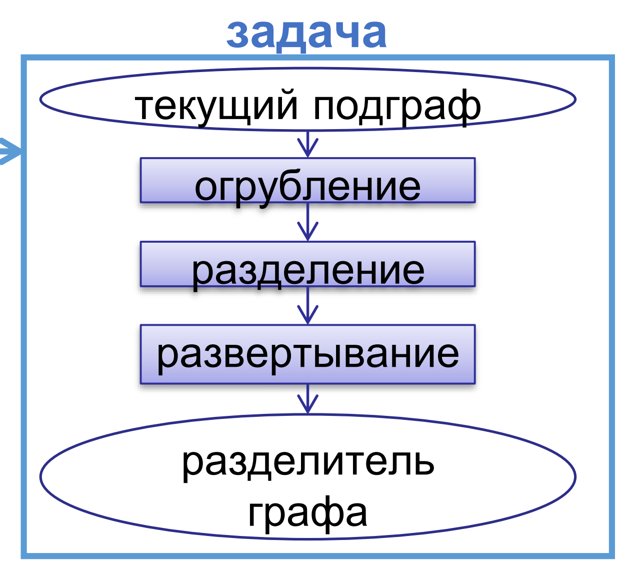
- Сохранение корректности разделителя:
  Пусть $\boldsymbol{P} = \{S, V_1, V_2\}$ и одна из частей графа меньше другой: $\|V_1\| < \|V_2\|$. Если $s \to V_1$, то $S^* = (S \cup Adj(s)) \backslash (s \cup V_1) = S \backslash s \ \cup \ Adj(s) \backslash V_1$,
  $$V_1^* = V_1 \cup s, V_2^* = V_2 \backslash Adj(s)$$



Параллельный метод вложенных сечений для систем с общей памятью

Дерево подграфов | Дерево задач

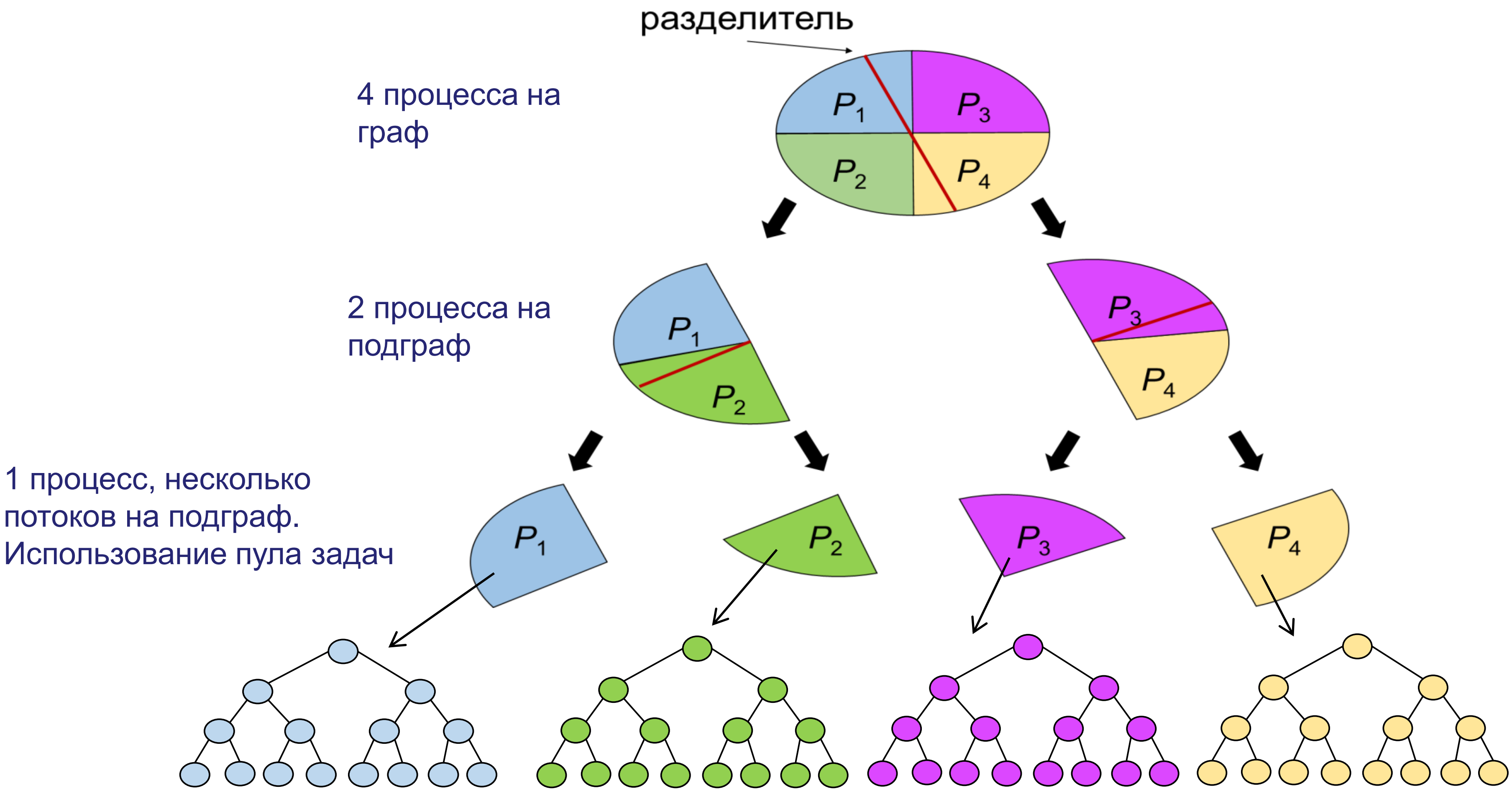




- 1 задача = найти разделитель в подграфе многоуровневым методом вложенных сечений
- Обход дерева выполняется сверху вниз, дерево строится в процессе переупорядочения
- Проблема динамической балансировки нагрузки

Гибридный алгоритм переупорядочения для систем с распределенной памятью

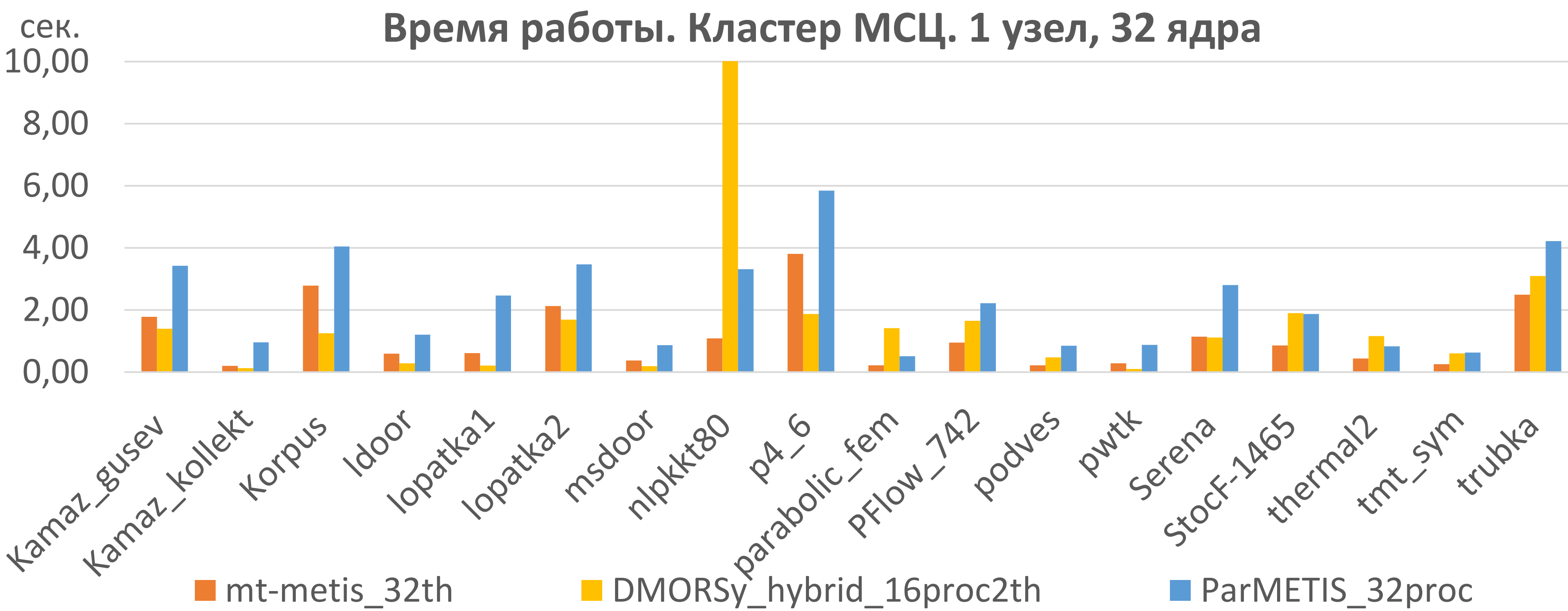
- Гибридный MPI + OpenMP алгоритм.



Библиотеки для переупорядочения

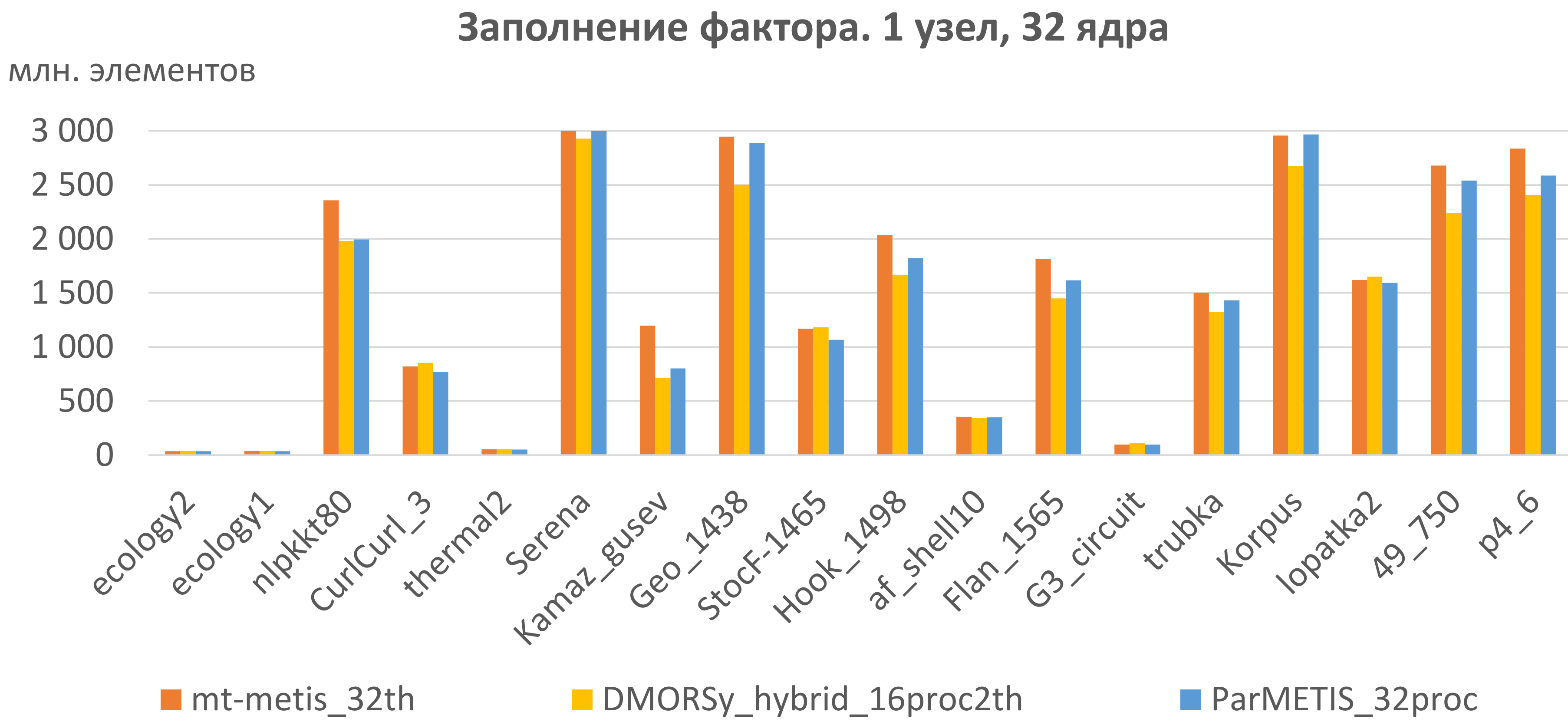
- В составе решателей СЛАУ (MUMPS, SuperLU, SuiteSparse, Intel OneAPI MKL, …)
- Специализированные библиотеки
  - METIS / ParMETIS / mt-metis – версии последовательная, параллельная для общей памяти, параллельная для распределенной памяти (MPI)
  - Scotch / PT-Scotch – версии последовательная, параллельная для распределенной памяти (MPI)
  - DMORSy (ННГУ) – версии последовательная, параллельная для общей памяти, параллельная для распределенной памяти (MPI + OpenMP)

Сравнение переупорядочивателей



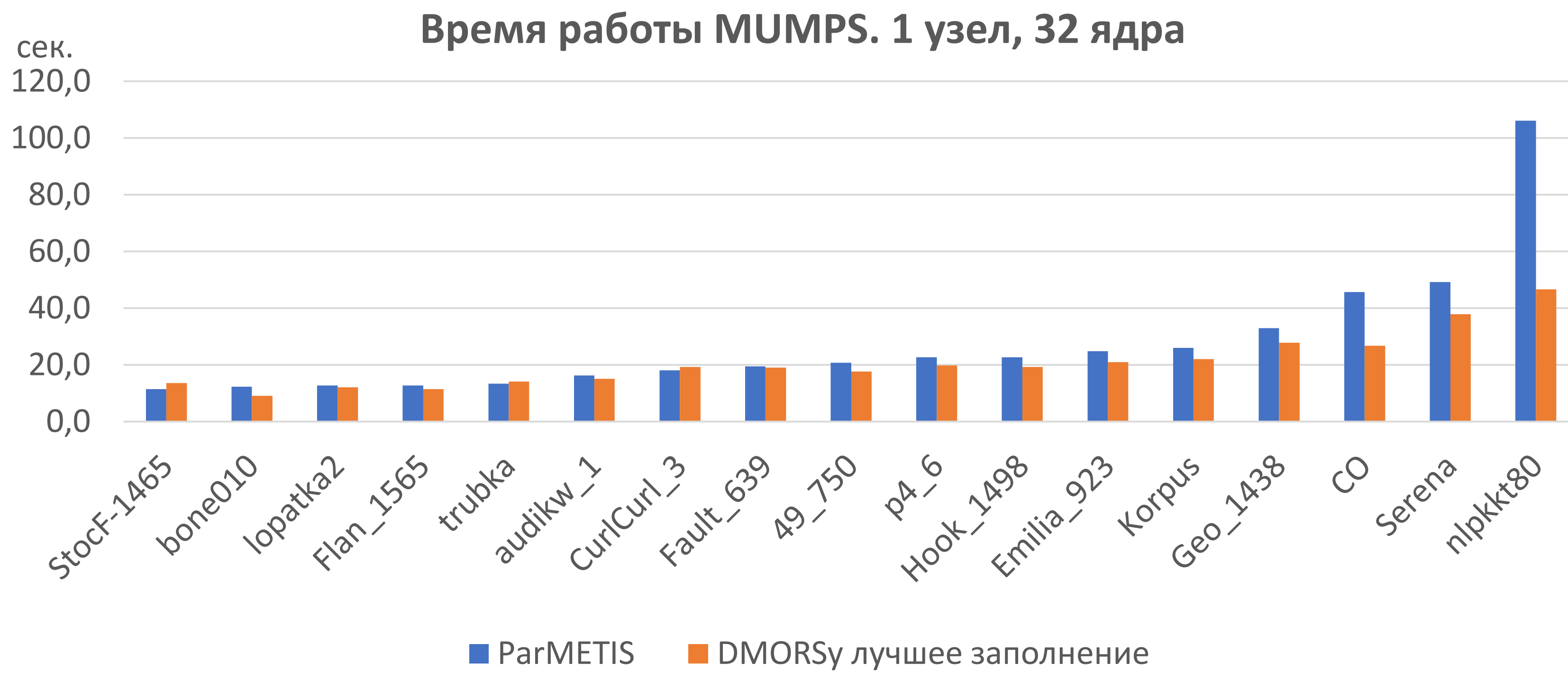
DMORSy работает быстрее, чем ParMETIS, на матрицах порядка менее 1 млн. строк и матрицах с регулярной структурой.

Сравнение переупорядочивателей



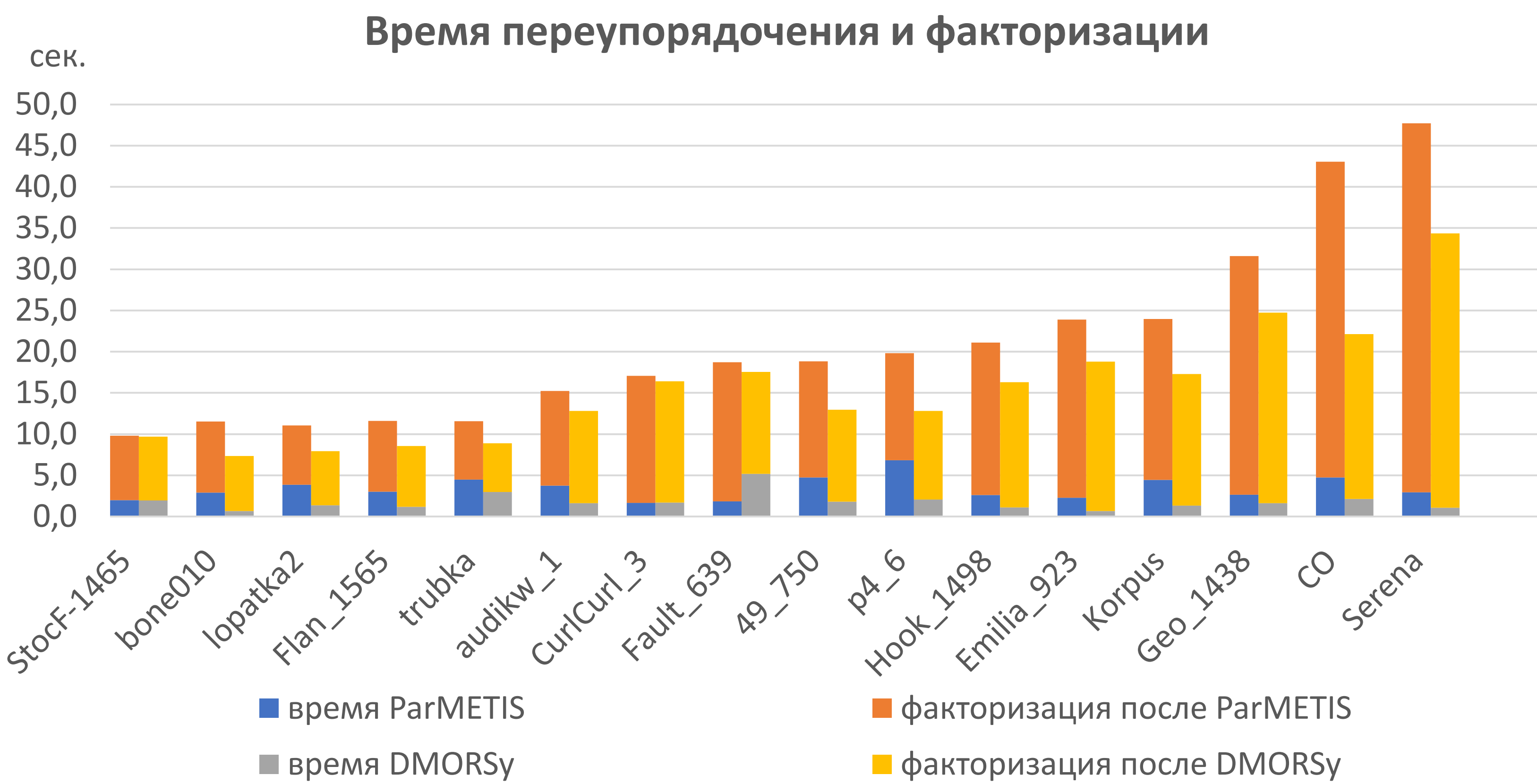
По заполнению фактора на половине тестовых матриц DMORSy опережает ParMETIS на ~8%, на других матрицах DMORSy уступает ParMETIS на ~12%.

Применение перестановок для решения СЛАУ



- Решение СЛАУ с перестановками из DMORSy позволяет **сократить время работы** на большинстве тестовых матриц (26 из 37).
- Среднее **опережение** по общему времени работы – 27% (26 матриц), среднее отставание на оставшихся матрицах – 26% (11 матриц).

Применение перестановок для решения СЛАУ



- На матрицах порядка 10^7 время численной факторизации в 4–30 раз больше, чем время переупорядочения

Литература

1. Davis T. A. Direct methods for sparse linear systems. – SIAM, 2006.
2. Heggernes P. et al. The computational complexity of the minimum degree algorithm. – Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States), 2002. – №. UCRL-JC-148375.
3. Пирова А.Ю. Гибридный MPI + OpenMP алгоритм переупорядочения симметричных разреженных матриц и его применение к решению СЛАУ // Проблемы информатики. – 2022.
   – № 1(54). – С. 28–41.
4. Chevalier C., Pellegrini F. PT-SCOTCH: a tool for efficient parallel graph ordering // Parallel Computing. – 2008. – Vol. 34, No. 6–8. – P. 318–331.
5. Karypis G., Kumar V. A fast and high-quality multilevel scheme for partitioning irregular graphs // SIAM J. on Scientific Computing. – 1999. – Vol. 20, No. 1. – P. 359–392.
6. LaSalle D., Karypis G. Efficient nested dissection for multicore architectures // Euro-Par 2015. – Springer Berlin Heidelberg, 2015. – С. 467-478.

Контакты

Нижегородский государственный университет

http://www.unn.ru

Институт информационных технологий, математики и механики

http://www.itmm.unn.ru

Пирова А.Ю.

anna.pirova@itmm.unn.ru