НАХОЖДЕНИЕ КОМПОНЕНТ СВЯЗНОСТИ ГРАФА С ПОМОЩЬЮ ВОЗМУЩЕНИЙ ЭЛЕМЕНТОВ МАТРИЦ СМЕЖНОСТИ

А.В. Пролубников

Омский государственный университет

a.v.prolubnikov@mail.ru

16.03.23

ЗАДАЧИ О СВЯЗНОСТИ НА ГРАФАХ

Связность графа

$G = \langle V(G), E(G) \rangle$ — неориентированный граф.
– G *связен*, если $\forall i, j \in V(G)$ в G существует соединяющая их *цепь.*
– *компонента связности* — максимальный по включению связный подграф в G.

Приложения:

- приложения, связанные с надёжностью инфраструктуры:
  - дорожная сеть,
  - маршрутизация в сетях (Интернете),
  - разработка БИС
  - и др.
- другие приложения, где требуется численно решать *большие* задачи о связности.

Как правило, большие графы — разреженные: $|E(G)| = O(|V(G)|)$.

Поиск в ширину (BFS)

Breadth-First Search

— естественный подход, используемый для решения задач о связности: **строим дерево достижимости вершин из выбранной вершины**

ℓ — диаметр графа $\Rightarrow$ *построение дерева достижимости за* ℓ *итераций*

Вычислительная сложность: $O(n+m)$.

- K. Zuse. Der Plankalkul. Стр. 96–105 (2.47–2.56). Konrad Zuse Internet Archive, 1972. URL: http: //zuse.zib.de/item/gHI1cNsUuQweHB6.
- E.F. Moore. *The shortest path through a maze* // Proceedings of the International Symposium on the Theory of Switching. Harvard University Press, 1959, pp. 285–292.
- C.Y. Lee. *An Algorithm for Path Connections and Its Applications* // IRE Transactions on Electronic Computers. 1961.

Алгебраический BFS

— *обход вершин как проведение вычислений вида:*

$$x^{(0)} = e_i, \quad x^{(k+1)} = Ax^{(k)}, \quad A \text{ — матрица смежности графа}$$

— GraphBLAS и другие *быстрые* реализации алгебраического BFS для **разреженных графов**.

Вычислительная сложность: $O(n) - O(mn)$.

- H.M. Bucker, C. Sohr. *Reformulating a breadth-fist search algorithm on an undirected graph in the language of linear algebra* // Intern. Conf. on Math. and Compu. in Sci. and in Industry, 2014, pp. 33–35.
- M. Besta, F. Marending, E. Solomonik, T. Hoefler. *SlimSell: a vectorizable graph representation for breadth-first seach* // 2017 IEEE Intern. Paral. and Distr. Proc. Symp., 2017, pp. 32–41.
- P. Burkhardt. *Optimal algebraic Breadth-First Search for sparse graphs* // arXiv:1906.03113v4 [cs.DS]. 30 Apr 2021.

ВОЗМУЩЕНИЯ МАТРИЦ ГРАФОВ

$A = A(G)$ — модифицированная (обратимая) матрица смежности.

$$A' = A + \varepsilon E_i,$$

$$(E_i)_{ii} = 1,\ (E_i)_{jk} = 0,\ j \neq i,\ k \neq i: \qquad a'_{ii} = a_{ii} + \varepsilon,\ \varepsilon > 0$$

$$\Rightarrow\ G \to G + \varepsilon(i, i)$$

- производим *возмущение* диагонального элемента A;
- выделяем компоненты связности, исходя из изменений элементов A^{-1}.

ВОЗМУЩЕНИЯ РАСПРОСТРАНЯЮТСЯ В ПРЕДЕЛАХ КОМПОНЕНТЫ СВЯЗНОСТИ:

$$(A^{-1})_{ij} = (-1)^{i+j} \cdot \frac{\det A_{ij}}{\det A} = (-1)^{i+j} \cdot \frac{\det A_{1,ij} \det A_2}{\det A_1 \det A_2} = (-1)^{i+j} \cdot \frac{\det A_{1,ij}}{\det A_1},$$

где A_1, A_2 — матрицы компонент связности.

Неориентированный граф: $(i,j) \in E(G) \Leftrightarrow (j,i) \in E(G)$

Модификация матрицы смежности

$A_0(G)$ — матрица смежности графа G.

$$A(G) = A_0(G) + d\mathcal{I},$$

где $\mathcal{I}$ — единичная матрица,

$$d > \max_{i \in V(G)} d_i,$$

d_i — степень вершины $i \in V(G)$.

$A(G)$ — <u>положительно определённая матрица</u> <u>со строгим диагональным преобладанием</u>.

ЗАДАЧА ПРОВЕРКИ ИЗОМОРФИЗМА ГРАФОВ

— также может быть решена с помощью возмущений матриц графов

$G = \langle V(G), E(G)\rangle,\ \ H = \langle V(H), E(H)\rangle$ — обыкновенные графы
$A(G),\ B(G)$ — модифицированные матрицы смежности.

Изоморфизм:

$G \simeq H \Leftrightarrow$

$$\Leftrightarrow\ \ \exists\varphi : V(G) \to V(H)\ \ :\ \ \Big((i,j) \in E(G) \Leftrightarrow (\varphi(i), \varphi(j)) \in E(H)\Big)$$

Теорема

$G \simeq H \Leftrightarrow$ возможно провести *согласованные возмущения*:

$$\det A^{(i)} = \det B^{(j_i)},\ \ i = \overline{1, n},$$

$A^{(0)} = A,\ B^{(0)} = B,\ A^{(i)} = A^{(i-1)} + \varepsilon_i E_i,\ B^{(i)} = B^{(i-1)} + \varepsilon_{j_i} E_{j_i}.$

Характеристический полином графа:

$$\chi_G(x) = \det(A(G) - x\mathcal{I}),\ x \in \mathbb{R}$$

Модифицированный характеристический полином:

$$\eta_G(x_1, \dots, x_n) = \det(A(G) + X),\ X = \operatorname{diag}(x_1, \dots, x_n).$$

Теорема

$$G \simeq H \text{ и } \varphi : V(G) \to V(H) \Leftrightarrow \eta_G(x_1, \dots, x_n) \equiv \eta_H(x_{\varphi(1)}, \dots, x_{\varphi(n)}).$$

Сравнение изменений элементов обратной матрицы эквивалентно проверке равенства значений полиномов η_G и η_H в точках $\varepsilon^{(i)} \in \mathbb{R}^n$: $\varepsilon^{(1)} = (\varepsilon_1, 0, 0, \dots, 0)$, $\varepsilon^{(2)} = (\varepsilon_1, \varepsilon_2, 0, \dots, 0)$, $\varepsilon^{(3)} = (\varepsilon_1, \varepsilon_2, \varepsilon_3, \dots, 0), \dots$:

$$\eta_G(\varepsilon^{(i)}) = \eta_H(\varepsilon_\varphi^{(i-1)} + \varepsilon_i e_j), \quad i = \overline{1, n}.$$

МОДИФИЦИРОВАННЫЙ ХАРАКТЕРИСТ. ПОЛИНОМ

Модифицированные характеристические полиномы
для графов на $n = 1, 2, 3$ вершинах:

- $n = 1$:

  $\eta_1 = x_1$;

- $n = 2$:

  $\eta_1 = x_1 x_2$,
  $\eta_2 = x_1 x_2 - 1$;

- $n = 3$:

  $\eta_1 = x_1 x_2 x_3$,
  $\eta_2 = x_1 x_2 x_3 - x_1$,
  $\eta_3 = x_1 x_2 x_3 - x_1 - x_3$,
  $\eta_4 = x_1 x_2 x_3 - x_1 - x_2 - x_3 + 2$.

ПОДХОД К РЕШЕНИЮ ЗАДАЧИ ИГ

- R.T. Faizullin, A.V. Prolubnikov *An algorithm of the spectral splitting for the double permutation cipher* // Pattern Recognition and Image Analysis. 2002. Vol. 12, No. 4. P. 365–375.
- А.В. Пролубников. *Сведение задачи проверки изоморфизма графов к задаче проверки равенства полиномов от n переменных* // Труды института математики и механики УрО РАН. 2016. Т. 22, №1. С. 235–240.
- А.В. Пролубников. *Точность и сложность вычислений, необходимые для проверки изоморфизма графов сравнением полиномов* // Вычислительные технологии. 2016. Т. 21, № 6. С. 71–88.
- A.V. Prolubnikov. Reduction of the graph isomorphism problem to equality checking of n-variables polynomials and the algorithms that use the reduction // arXiv.org, 2016. http://arxiv.org/pdf/1512.03139.pdf

ОТКЛИК НА ВОЗМУЩЕНИЕ

Производим возмущение матрицы A графа и анализируем отклик на него в элементах A^{-1}

Нахождение элементов (строк/столбцов) A^{-1}:

Решаем СЛАУ $Ax = e_i$. x — i-й столбец A^{-1}

Проверка принадлежности вершин одной комп.св.:

1). Решаем СЛАУ

$$Ax = e_i. \tag{1}$$

2). Производим возмущение: $A' = A + \varepsilon E_i$.

3). Решаем СЛАУ

$$A'x' = e_i. \tag{2}$$

4). <u>Сравниваем:</u> $x_j \overset{?}{\neq} x'_j$ — **да** $\Rightarrow$ в одной комп.св., **нет** $\Rightarrow$ в разных.

Компоненты решения СЛАУ

$$x_j = x_j(0) = \frac{A_{ij}}{\det A(0)} = (-1)^{i+j} \cdot \frac{\eta_{G_{ij}}(0,\ldots,0)}{\eta_G(0,\ldots,0)},$$

$$x'_j = x_j(\varepsilon) = \frac{A_{ij}}{\det A(\varepsilon)} = (-1)^{i+j} \cdot \frac{\eta_{G_{ij}}(0,\ldots,0)}{\eta_G(\varepsilon e_i)}.$$

Имеем: $\eta_G(0,\ldots,0) \neq \eta_G(\varepsilon e_i),\ \varepsilon > 0.$

Проверка неравенства компонент:

$$x_j \neq x'_j \ \Leftrightarrow\ x_j(0) \neq x_j(\varepsilon) \Leftrightarrow \eta_{G_{ij}}(0,\ldots,0) \neq 0$$

$\eta_{G_{ij}}(0,\ldots,0) \neq 0\ \Leftrightarrow i,j$ принадлежат одной компоненте связности

АЛГОРИТМ 1

Нахождение компонент связности (G)

$V \leftarrow V(G)$;
$K \leftarrow 1$;
$V_K \leftarrow \varnothing$;
while $V \neq \varnothing$:
выбрать $i \in V$;
$V_K \leftarrow V_K \cup \{i\}$; $V \leftarrow V \setminus \{i\}$;
найти решение СЛАУ (1) — вектор x;
найти решение СЛАУ (2) — вектор x';
for $\forall j \in V$:
if $x'_j \neq x_j$
$V_K \leftarrow V_K \cup \{j\}$; $V \leftarrow V \setminus \{j\}$;
$K \leftarrow K + 1$;

Выход: V_k, $k = \overline{1, K}$ — компоненты связности G.

ОБОСНОВАНИЕ АЛГОРИТМА 1

УТВЕРЖДЕНИЕ 1.

$\det A_{ij} = 0$ тогда и только тогда, когда $i, j \in V(G)$ принадлежат различным компонентам связности графа G.

Если $i, j \in V(G)$ принадлежат одной компоненте связности, то

$$\left|x_j - x_j'\right| = \frac{\varepsilon \left|\det A_{ij}\right| \det A_{ii}}{\det A(\det A + \varepsilon \det A_{ii})} \geq \Delta > 0.$$

УТВЕРЖДЕНИЕ 2.

Если $i, j \in V(G)$ принадлежат одной компоненте связности, то

$$\Delta > \frac{\varepsilon}{d^{n+1}} = \frac{10}{d^n},$$

при $\varepsilon = 10d$.

Наихудший случай: G — простая цепь, $\ell(i,j) = n$.

Точность и сложность вычислений

Для итерационных методов (простая итерация, метод Гаусса-Зейделя):

$$\left|x_j^{(k+1)} - x_j^{(k)}\right| \le \left\|x^{(k+1)} - x^{(k)}\right\| < \frac{\Delta_0}{\mu^k},$$

где Δ_0 — точность начального приближения, μ: $d = \mu d_{\max}$.

Для фиксирования неравенства точных значений $x_j \neq x_j'$ надо провести такое количество итераций численного метода N, что

$$\frac{\delta_0}{\mu^N} < \frac{\Delta}{4}.$$

$$\Rightarrow \quad N > \log_\mu\left(\frac{4\delta_0}{\Delta}\right) = (n+1)\log_\mu d + \log_\mu(8\delta_0) \approx (n+1)\log_\mu d,$$

где $\log_\mu d = \log_\mu(\mu d_{\max}) = 1 + \log_\mu d_{\max} = 2$ при $\mu = d_{\max}$.

$N = O(n)$. Сложность одной итерации — $O(m)$.

$\Rightarrow$ общая сложность **Алгоритма 1** — $O(nm)$.

Вычислительная сложность BFS

BFS как обход графа:

вычислительная сложность — $O(m+n)$:

за время работы

- обходится не более m рёбер,
- посещается n вершин.

Можем обрабатывать только по одному уровню дерева, иначе возможно зацикливание.

$\Rightarrow$ **не существует алгоритма, реализующего BFS, со сложностью менее, чем $O(m+n)$.**

Реализации алгебраического BFS при теоретической сложности $O(mn)$ могут давать ускорение относительно теоретически лучшего BFS, реализуемого через обход,
за счёт распараллеливания вычислений на одном уровне.

Использование реализаций численных методов решения СЛАУ для решения задач о связности:

- распараллеливание BFS затруднено.

Тогда как

- есть эффективные параллельные реализации численных методов решения СЛАУ

  в том числе для разреженных графов.

ИТЕРАЦИОННЫЕ ЧИСЛЕННЫЕ МЕТОДЫ И ОБХОДЫ ГРАФА

$x^{(0)} = e_i = (0, \dots, 1, \dots, 0).$

1. Метод простой итерации:

итерации — модификация алгебраического BFS:

$$x^{(k+1)} = Ax^{(k)} \ \rightarrow \ x^{(k+1)} = b - D^{-1}Ax^{(k)}.$$

$$x_j^{(k+1)} = \frac{1}{a_{jj}}\left(b_j - \sum_{l \neq j} a_{jl}x_j^{(k)}\right), \quad b_j \in \{0, 1\}.$$

2. Метод Гаусса-Зейделя:

$$(L + D)x^{(k+1)} = -Ux^{(k)} + b.$$

$$x_j^{(k+1)} = \frac{1}{a_{jj}}\left(b_j - \sum_{l=1}^{j-1} a_{jl}x_l^{(k+1)} - \sum_{l=j+1}^{n} a_{jl}x_l^{(k)}\right), \quad b_j \in \{0, 1\}.$$

итерации — **НЕ** модификация алгебраического BFS!

Если достигать сходимости итрационных численных методов, то не принципиально, какой метод мы используем при реализации **Алгоритма 1.**

Но если количество итераций относительно невелико, то разные методы задают разные последовательности приближений, соответствующие различным вариантам обхода графа.

Обход, который, например, даёт метод Гаусса-Зейделя отличается от обхода, который реализуется при BFS.

Обход: переход по ребру из уже достигнутой вершины в не достигнутую ранее вершину j эквивалентен тому, что значение x_j перестаёт быть равным нулю:

$$x_j^{(k)} = 0, \text{ но } x_j^{(k+1)} \neq 0 \text{ становится ненулевым.}$$

АЛГОРИТМ 2 (GSS)

Нахождение компоненты связности 2 $(G; i \in V(G)) : C;$

$x^{(0)} = e_i;\ C \leftarrow \{i\};$

$x^{(1)} = 0 \in \mathbb{R}^n;$

$k \leftarrow 1;$

while $\exists j \in V(G) : (x_j^{(k)} = 0$ и $x_j^{(k+1)} \neq 0)$

  for $\forall j \in V(G)$:

$$x_j^{(k+1)} = a_{jj} \cdot \left(b_j - \sum_{l=1}^{j-1} a_{jl} x_l^{(k+1)} - \sum_{l=j+1}^{n} a_{jl} x_l^{(k)} \right), \quad b_j \in \{0, 1\}.$$

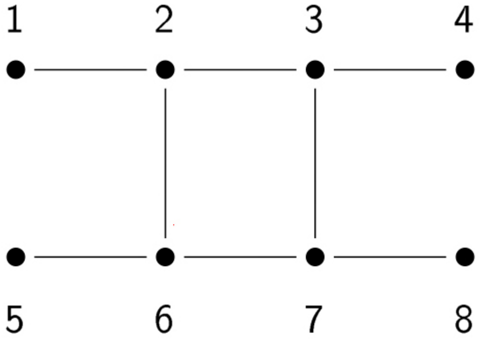
  $k \leftarrow k + 1;$

$C \leftarrow C \cup \{j : x_j^{(k)} \neq 0\}.$

Выход: C — вершины компоненты связности G, содержащей i.

ИТЕРАЦИЯ BFS (SIS) ДЛЯ ГРАФА:

Граф G:



$i = 1$

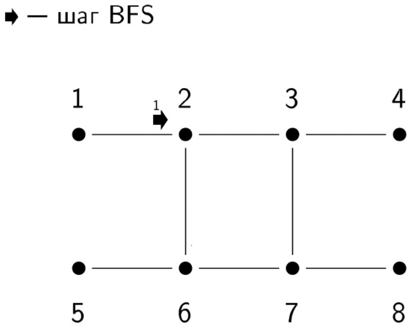
Итерация BFS (SIS) для графа:

- $x_1^{(k+1)} = \ldots - x_2^{(k)};$
- $x_2^{(k+1)} = \ldots - x_1^{(k)} - x_3^{(k)} - x_6^{(k)};$
- $x_3^{(k+1)} = \ldots - x_2^{(k)} - x_4^{(k)} - x_7^{(k)};$
- $x_4^{(k+1)} = \ldots - x_3^{(k)};$
- $x_5^{(k+1)} = \ldots - x_6^{(k)};$
- $x_6^{(k+1)} = \ldots - x_2^{(k)} - x_5^{(k)} - x_7^{(k)};$
- $x_7^{(k+1)} = \ldots - x_3^{(k)} - x_6^{(k)} - x_8^{(k)};$
- $x_8^{(k+1)} = \ldots - x_7^{(k)}.$

BFS. Итерация 1.

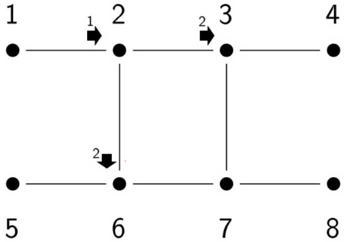
$x^{(0)} = e_i,\ x^{(k+1)} = Ax^{(k)}$.

$x_j^{(k-1)} \neq 0 \Leftrightarrow$ вершина j достигнута до k-й итерации.



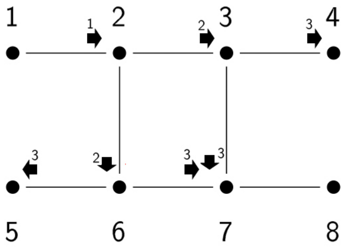
BFS. Итерация 2.

➡ — шаг BFS



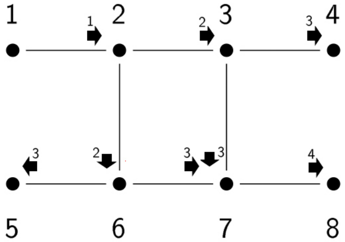
BFS. ИТЕРАЦИЯ 3.

➡ — шаг BFS



BFS. Итерация 4.

➡ — шаг BFS



ИТЕРАЦИЯ BFS (SIS) ДЛЯ ГРАФА:

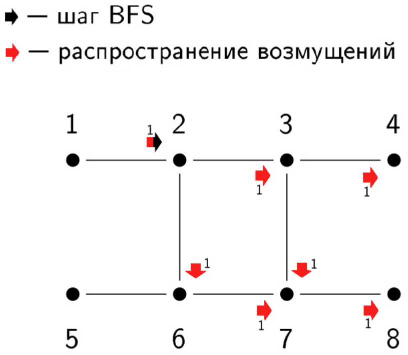
- $x_1^{(k+1)} = \ldots - x_2^{(k)}$;
- $x_2^{(k+1)} = \ldots - x_1^{(k)} - x_3^{(k)} - x_6^{(k)}$;
- $x_3^{(k+1)} = \ldots - x_2^{(k)} - x_4^{(k)} - x_7^{(k)}$;
- $x_4^{(k+1)} = \ldots - x_3^{(k)}$;
- $x_5^{(k+1)} = \ldots - x_6^{(k)}$;
- $x_6^{(k+1)} = \ldots - x_2^{(k)} - x_5^{(k)} - x_7^{(k)}$;
- $x_7^{(k+1)} = \ldots - x_3^{(k)} - x_6^{(k)} - x_8^{(k)}$;
- $x_8^{(k+1)} = \ldots - x_7^{(k)}$.

Итерация GSS для графа:

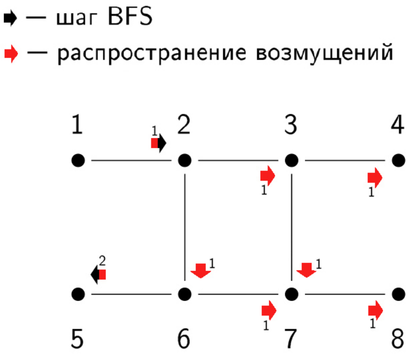
- $x_1^{(k+1)} = \ldots - x_2^{(k)};$
- $x_2^{(k+1)} = \ldots - x_1^{(k+1)} - x_3^{(k)} - x_6^{(k)};$
- $x_3^{(k+1)} = \ldots - x_2^{(k+1)} - x_4^{(k)} - x_7^{(k)};$
- $x_4^{(k+1)} = \ldots - x_3^{(k+1)};$
- $x_5^{(k+1)} = \ldots - x_6^{(k)};$
- $x_6^{(k+1)} = \ldots - x_2^{(k+1)} - x_5^{(k+1)} - x_7^{(k)};$
- $x_7^{(k+1)} = \ldots - x_3^{(k+1)} - x_6^{(k+1)} - x_8^{(k)};$
- $x_8^{(k+1)} = \ldots - x_7^{(k+1)}.$

ПЕРВАЯ ИТЕРАЦИЯ GSS

Возмущение диагонального элемента a_{11} *распространяется через обходы по всем цепям* **с правильным порядком**, *исходящим из достигнутых на итерации вершин:*

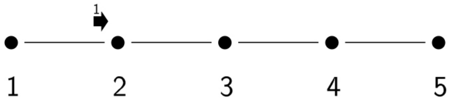


ВТОРАЯ ИТЕРАЦИЯ GSS

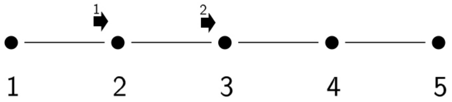


Если в вершину доставлено возмущение,
то на следующей итерации из него стартует ещё один BFS-обход.

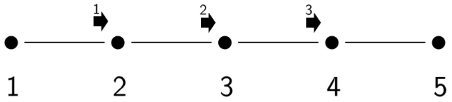
BFS. ИТЕРАЦИЯ 1.



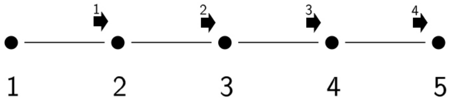
BFS. Итерация 2.



BFS. ИТЕРАЦИЯ 3.



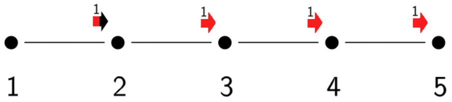
BFS. Итерация 4.



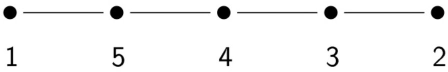
Итерация GSS для графа:

- $x_1^{(k+1)} = \ldots - x_2^{(k)}$;
- $x_2^{(k+1)} = \ldots - x_1^{(k+1)} - x_3^{(k)}$;
- $x_3^{(k+1)} = \ldots - x_2^{(k+1)} - x_4^{(k)}$;
- $x_4^{(k+1)} = \ldots - x_3^{(k+1)} - x_5^{(k)}$;
- $x_5^{(k+1)} = \ldots - x_4^{(k+1)}$.

ИТЕРАЦИЯ GSS ДЛЯ ГРАФА:

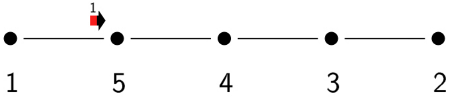


ИТЕРАЦИЯ GSS ДЛЯ ГРАФА:

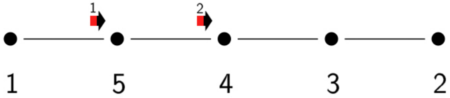


- $x_1^{(k+1)} = \ldots - x_5^{(k)}$;
- $x_2^{(k+1)} = \ldots - x_3^{(k)}$;
- $x_3^{(k+1)} = \ldots - x_2^{(k+1)} - x_4^{(k)}$;
- $x_4^{(k+1)} = \ldots - x_3^{(k+1)} - x_5^{(k)}$;
- $x_5^{(k+1)} = \ldots - x_1^{(k+1)} - x_4^{(k+1)}$.

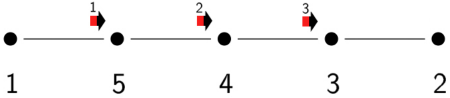
GSS. ИТЕРАЦИЯ 1.



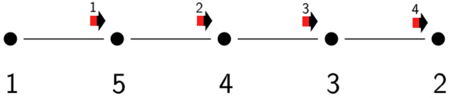
GSS. ИТЕРАЦИЯ 2.



GSS. Итерация 3.



GSS. Итерация 4.



ТЕСТОВЫЕ ЗАДАЧИ

РАЗРЕЖЕННЫЕ ГРАФЫ

- $n = 90\ 000$, $m = 89\ 100$, $K = 900$.

Компоненты связности — цепи на 100 вершинах.

- Решение СЛАУ — *простая итерация:*
  $N = 110$, $d = 100$

  — 20 **мин.**
- Решение СЛАУ — *метод Гаусса-Зейделя:*
  $N = 80$, $d = 1000$

  — 9 **мин.**
- GSS:
  $d = 1$

  — 3 **мин.**

— ПОСЛЕДОВАТЕЛЬНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМОВ

ТЕСТОВЫЕ ЗАДАЧИ

РАЗРЕЖЕННЫЕ ГРАФЫ

Граф *задачи* о связности транспортной сети:

— наличие ребёр соответствует вхождению переменных в одно уранение или неравенство:

- Решение СЛАУ — **простая итерация:**
- $n = 367\ 840$, $m = 53\ 404\ 685$, $m = 150n$, $K = 224$.

32 комп. связн. по 11 429 вершин, 192 — по 11 (цепи) — **97 мин.**

не алгебраический BFS ≈48 **часов**

— ПОСЛЕДОВАТЕЛЬНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМОВ

Вычислительная сложность алгоритма 2 (GSS)

Сложность одной итерации — $O(m)$.
ℓ — диаметр графа. Количество итераций в худшем случае — ℓ.
$\Rightarrow$ общая сложность — $O(\ell m)$.

Количество итераций определяется нумерацией вершин.

Утверждение 3.

Для любой индивидуальной задачи нахождения компонент связности вычислительная сложность GSS не превышает вычислительной сложности алгебраического BFS.

Вычислительная сложность GSS меньше сложности BFS, если в ходе итераций GSS достигаются вершины, лежащие на цепи с длиной равной диаметру,

из которых исходят цепи с правильным порядком.

Выводы

- Предложен подход к решению задач о связности графа, использующий возмущения элементов матрицы смежности.
- Подход позволяет решать задачи, используя эффективные реализации решения СЛАУ.
- Итерационные методы решения СЛАУ рассмотрены как обходы графа.
- Предложен алгоритм обхода графа для нахождения компонент связности графа, <u>не эквивалентный</u> подходу, реализуемому BFS, для любой индивидуальной задачи гарантированно имеющий сложность не большую, чем сложность BFS.