

## ПАРАЛЛЕЛЬНЫЕ АЛГОРИТМЫ И ЗАДАЧИ МНОГОКРИТЕРИАЛЬНОЙ ДИСКРЕТНОЙ ОПТИМИЗАЦИИ НА МАСШТАБНО-ИНВАРИАНТНЫХ (ПРЕДФРАКТАЛЬНЫХ) ГРАФАХ

Р.А. Кочкаров

*Финансовый университет при Правительстве РФ*  
Россия, 125993, Москва, Ленинградский просп., 49  
E-mail: [rasul\\_kochkarov@mail.ru](mailto:rasul_kochkarov@mail.ru)

А.А. Кочкаров

*Институт проблем управления РАН им. В.А. Трапезникова*  
Россия, 117997, Москва, ул. Профсоюзная, 65  
E-mail: [azret\\_kochkarov@mail.ru](mailto:azret_kochkarov@mail.ru)

**Ключевые слова:** масштабно-инвариантные (фрактальные, самоподобные) графы, параллельные алгоритмы на графах, оптимизационные задачи на графах

**Key words:** Scale-free (self-similar, fractal) graphs, parallel algorithms on graphs, graph optimization problems

Работа посвящена параллельным алгоритмам решения многокритериальных задач на предфрактальных графах. Распараллеливание алгоритмов проводится благодаря свойству структурного самоподобия предфрактального графа. Вычислительная сложность алгоритмов на порядок ниже, чем у общеизвестных алгоритмов.

**PARALLEL VERIFICATION OF ROUTING INFORMATION PROTOCOL MODEL** / R.A. Kochkarov (Finance University under the Government of the Russian Federation, Leningradsky Prospect, 49-55, Moscow, 125993, Russia.), A.A. Kochkarov (Institute of Control Sciences RAS, Profsoyuznaya str., 65, Moscow, 117997, Russia). This paper is devoted to parallel algorithms for solving the multicriteria prefractal graph problems. The algorithms are made parallel owing to the fact that the self-similarity is structural property of prefractal graphs. The computational complexity of these parallel algorithms is lower than the computational complexity of well-known algorithms.

## 1. Введение

В практической деятельности постоянно возникают задачи «наилучшего» размещения оборудования (или средств обслуживания) в сетях или графах. В частности, если граф представляет сеть дорог и вершины соответствуют отдельным районам, то можно поставить задачу оптимального размещения больниц, полицейских участков, пожарных частей и многих других крайне необходимых предприятий и служб. В таких случаях критерий оптимальности может состоять в минимизации расстояния (или времени проезда) от пункта обслуживания до самой отдаленной вершины графа, т.е. в оптимизации «наихудшего варианта». В более общей задаче требуется разместить несколько таких пунктов обслуживания (а не только один). При этом самая отдаленная вершина графа должна находиться по крайней мере от одного пункта обслуживания на минимально возможном расстоянии. К таким задачам относятся задачи размещения аварийных служб, и поэтому объективным требованием здесь является минимизация наибольшего расстояния от произвольной вершины графа до ближайшего к ней пункта обслуживания. По очевидным причинам задачи такого типа называются *минимаксными задачами размещения*. Полученные при решении этих задач места размещения пунктов обслуживания называются *центрами графа*.

В некоторых задачах размещения лучше всего было бы минимизировать сумму всех расстояний от вершин графа до центра обслуживания (если предполагать, что ищется место для размещения только одного такого пункта обслуживания). Такой критерий является наиболее подходящим, например, в задаче о размещении склада в сети дорог, где вершины

представляют потребителей, обслуживаемых этим складом, или в задаче размещения телефонных станций в телефонной сети, где вершины представляют абонентов. Задачи такого типа вообще относятся к *минисуммным задачам размещения*, хотя целевая функция является часто не просто суммой расстояний, а суммой различных функций от расстояний. Места размещения пунктов обслуживания, полученные в результате решения минисуммной задачи, называются *медианами* графа [1].

Среди работ, посвященных математическому моделированию структурно-сложных систем, почти не встречаются работы, связанные с моделированием систем с изменяющейся структурой. А именно такими системами являются все современные информационные, электроэнергетические и коммуникационные сети. В таких сетевых системах нередко возникает необходимость решения задачи размещения. В качестве моделей структур таких систем предложено использовать предфрактальные графы.

## 2. Фрактальные и предфрактальные графы

Термином *затравка* условимся называть какой-либо связный граф  $H = (W, Q)$ . Для определения *фрактального (предфрактального) графа* [142] нам потребуется операция *замены вершины затравкой (ЗВЗ)*. Суть операции ЗВЗ заключается в следующем. В данном графе  $G = (V, E)$  у намеченной для замещения вершины  $\tilde{v} \in V$  выделяется множество  $\tilde{V} = \{\tilde{v}_j\} \subseteq V$ ,  $j = 1, 2, \dots, |\tilde{V}|$ , смежных ей вершин. Далее из графа  $G$  удаляется вершина  $\tilde{v}$  и все инцидентные ей ребра. Затем каждая вершина  $\tilde{v}_j \in \tilde{V}$ ,  $j = 1, 2, \dots, |\tilde{V}|$ , соединяется ребром с одной из вершин затравки  $H = (W, Q)$ . Вершины соединяются произвольно (случайным образом) или по определенному правилу, при необходимости.

Предфрактальный граф будем обозначать через  $G_L = (V_L, E_L)$ , где  $V_L$  – множество вершин графа, а  $E_L$  – множество его ребер. Определим его рекуррентно, поэтапно, заменяя каждый раз в построенном на предыдущем этапе  $l = 1, 2, \dots, L - 1$  графе  $G_l = (V_l, E_l)$  каждую его вершину затравкой  $H = (W, Q)$ . На этапе  $l = 1$  предфрактальному графу соответствует затравка  $G_1 = H$ . Об описанном процессе говорят, что *предфрактальный граф*  $G_L = (V_L, E_L)$  *порожден затравкой*  $H = (W, Q)$ . Процесс порождения предфрактального графа  $G_L$ , по существу, есть процесс построения последовательности предфрактальных графов  $G_1, G_2, \dots, G_l, \dots, G_L$ , называемой *траекторией*. Фрактальный граф  $G = (V, E)$ , порожденный затравкой  $H = (W, Q)$ , определяется бесконечной траекторией [2].

Предфрактальный граф  $G_L = (V_L, E_L)$  условимся называть  $(n, q, L)$ -*графом*, если он порожден  $n$ -вершинной  $q$ -реберной связной затравкой  $H = (W, Q)$ , и  $(n, L)$ -*графом*, если затравка  $H = (W, Q)$  – регулярный граф.

Использование операции ЗВЗ в процессе порождения предфрактального графа  $G_L$ , для элементов  $G_l = (V_l, E_l)$ ,  $l \in \{1, 2, \dots, L - 1\}$ , его траектории позволяет ввести отображение  $\varphi : V_l \rightarrow V_{l+1}$  или  $\varphi(V_l) = V_{l+1}$ , а в общем виде

$$\varphi^t(V_l) = V_{l+t}, \quad t = 1, 2, \dots, L - l. \quad (1)$$

В выражении (1) множество  $V_{l+t}$  – *образ* множества  $V_l$ , а множество  $V_l$  – *прообраз* множества  $V_{l+t}$ .

Для предфрактального графа  $G_L$ , ребра, появившиеся на  $l$ -ом,  $l \in \{1, 2, \dots, L\}$ , этапе порождения, будем называть *ребрами ранга  $l$* . Новыми ребрами предфрактального графа  $G_L$  назовем ребра ранга  $L$ , а все остальные ребра назовем – *старыми*.

Если из предфрактального графа  $G_L$ , порожденного  $n$ -вершинной затравкой  $H$ , последовательно удалить все старые ребра (ребра ранга  $l$ ,  $l = 1, 2, \dots, L - 1$ ), то исходный граф распадется на множество связанных компонент  $\{B_L^{(1)}\}$ , каждая из которых изоморфна [48] затравке  $H$ . Множество компонент  $\{B_L^{(1)}\}$  будем называть *блоками первого ранга*. Аналогично, при удалении из предфрактального графа  $G_L$  всех старых ребер рангов  $l = 1, 2, \dots, L - 2$ , получим множество *блоков  $\{B_L^{(2)}\}$  второго ранга*. Обобщая, скажем, что при удалении из предфрактального графа  $G_L$  всех ребер рангов  $l = 1, 2, \dots, L - r$ , получим множество  $\{B_{L,i}^{(r)}\}$ ,  $r \in \{1, 2, \dots, L - 1\}$ , *блоком  $r$ -го ранга*, где  $i = 1, 2, \dots, n^{L-r}$  - порядковый номер блока. Блоки  $B_L^{(1)} \subseteq G_L$  первого ранга также будем называть *подграф-затравками  $H$*  предфрактального графа  $G_L$ . Очевидно, что всякий блок  $B_L^{(r)} = (U_L^{(r)}, M_L^{(r)})$ ,  $r \in \{1, 2, \dots, L - 1\}$ , является предфрактальным графом  $B_r = (U_r, M_r)$ , порожденным затравкой  $H$ .

Уточним для отображения  $\varphi$  в формуле (2.1) ряд подробностей. Для любой вершины  $v_j \in V_l$ ,  $j \in \{1, 2, \dots, n^l\}$ , предфрактального графа  $G_l = (V_l, E_l)$ ,  $l \in \{1, 2, \dots, L - 1\}$ , из траектории графа  $G_L$ , справедливо

$$\varphi^t(v_j) = U_{l+t,j}^{(t)}, \quad (2)$$

$$\varphi^t(v_j) = B_{l+t,j}^{(t)}, \text{ где } B_{l+t,j}^{(t)} = (U_{l+t,j}^{(t)}, M_{l+t,j}^{(t)}) \subseteq G_{l+t}, \quad t = 1, 2, \dots, L - l.$$

Аналогично,

$$\varphi^t(U_{l,i}^{(r)}) = U_{l+t,i}^{(r+t)}, \quad (3)$$

$$\varphi^t(B_{l,i}^{(r)}) = B_{l+t,i}^{(r+t)}, \quad r \in \{1, 2, \dots, L - t\}, \quad i \in \{1, 2, \dots, n^{L-r}\}.$$

Два блока предфрактального графа назовем *смежными*, если существует ребро, вершины которого принадлежат различным блокам. Не требует доказательства тот факт, что блоки предфрактального графа смежны тогда и только тогда, когда смежны их прообразы из (2).

**УТВЕРЖДЕНИЕ 1.** *Всякий предфрактальный граф  $G_L$  можно представить в виде множества подграф-затравок  $\{B_L^{(1)}\}$ , соединенных старыми ребрами разных рангов. А именно, старые ребра  $(L - 1)$ -го ранга объединяют множество подграф-затравок в множество блоков  $\{B_L^{(2)}\}$  второго ранга, их в свою очередь, старые ребра  $(L - 2)$ -го ранга объединяют в множество блоков  $\{B_L^{(3)}\}$  третьего ранга и т.д. Окончательно, старые ребра первого ранга объединяют множество  $\{B_L^{(L-1)}\}$  блоков  $(L - 1)$ -го ранга в связный предфрактальный граф  $G_L$ . ◀*

Термином *подграф-затравка*  $z_s^{(l)}$  будем называть блок  $B_{l,s}^{(1)}$ ,  $s = \overline{1, n^{l-1}}$ , первого ранга предфрактального графа  $G_l$ ,  $l = \overline{1, L}$  из траектории. Последовательное выделение подграф-затравок  $z_s^{(l)}$  на графах  $G_1, G_2, \dots, G_L$  из траектории предфрактального графа  $G_L$  разбивает множество ребер  $E_L$  на непересекающиеся подмножества подграф-затравок  $Z(G_L) = \{z_s^{(l)}\}$ , где  $l = \overline{1, L}$ , – ранг подграф-затравки, а  $s = \overline{1, n^{l-1}}$  – ее порядковый номер. Такое разбиение на подмножества позволит нам сохранить информацию смежности старых ребер на момент их появления в предфрактальном графе. В траектории переход от графа  $G_{l-1}$  к  $G_l$  осуществляется  $|V_{l-1}| = n^{l-1}$  операциями ЗВЗ, поэтому общее число использованных затравок в порождении предфрактального графа  $G_L$  равно  $1 + n + n^2 + \dots + n^{L-1} = \frac{n^L - 1}{n - 1}$ . Тогда мощность множества  $Z(G_L)$  всех подграф-затравок из траектории графа  $G_L$  также равно  $|Z(G_L)| = \frac{n^L - 1}{n - 1}$ .

Будем говорить, что *предфрактальный граф*  $G_L = (V_L, E_L)$  – *взвешен*, если каждому его ребру  $e^{(l)} \in E_L$  приписано действительное число  $w(e^{(l)}) \in (\theta^{l-1}a, \theta^{l-1}b)$ , где  $l = \overline{1, L}$  – ранг ребра,  $a > 0$ , и  $\theta < \frac{a}{b}$ . Обобщением описанного процесса порождения предфрактального графа  $G_L$  является такой случай, когда вместо единственной затравки  $H$  используется множество затравок  $\mathbf{H} = \{H_t\} = \{H_1, H_2, \dots, H_t, \dots, H_T\}$ ,  $T \geq 2$ . Суть этого обобщения состоит в том, что при переходе от графа  $G_{l-1}$  к графу  $G_l$  каждая вершина замещается некоторой затравкой  $H_t \in \mathbf{H}$ , которая выбирается случайно или согласно определенному правилу, отражающему специфику моделируемого процесса или структуры.

### 3. Многокритериальная задача размещения на предфрактальных графах

Сформулируем многокритериальную постановку задачи выделения  $p$ -центра предфрактального графа [2,3]. Рассмотрим взвешенный предфрактальный граф  $G_L = (V_L, E_L)$ , порожденный затравкой  $H = (W, Q)$ , у которой  $|W| = n$ ,  $|Q| = q$ .

Пусть  $x$  – подмножество (содержащее  $p$  вершин) множества вершин предфрактального графа  $G_L = (V_L, E_L)$ . Через  $d(x, v_i)$  будем обозначать наикратчайшее из расстояний между вершинами множества  $x$  и вершиной  $v_i$ , т.е.  $d(x, v_i) = \min_{v_j \in x} [d(v_j, v_i)]$ .

Число разделения  $s(x)$  для множества вершин  $x$  определяется следующим образом –  $s(x) = \max_{v_j \in V_L} [d(x, v_j)]$ . Множество  $x^*$  для которого  $s(x^*) = \min_{x \subseteq V_L} [s(x)]$  называется

*$p$ -центром предфрактального графа*  $G_L = (V_L, E_L)$ .

Всевозможные  $p$ -центры  $\{x\}$  предфрактального графа  $G_L$  образуют множество допустимых решений  $X = X(G_L) = \{x\}$  (МДР).

На множестве  $x$  определим векторно-целевую функцию (ВЦФ):

$$\mathbf{F}(x) = (F_1(x), F_2(x), F_3(x), F_4(x)), \quad (4)$$

$$F_1(x) = s(x) \rightarrow \min, \quad (5)$$

где  $s(x)$  – число разделения множества  $x$ ;

$$F_2(x) = \sum_{t=1}^p \rho_t \rightarrow \min, \quad (6)$$

где  $\rho_t$  – радиус отдельного центра;

$$F_3(x) = \hbar \rightarrow \min, \quad (7)$$

где  $\hbar$  – количество типов центров;

$$F_4(x) = p \rightarrow \min, \quad (8)$$

где  $p$  – количество вершин, составляющих  $p$ -центр.

Все критерии (5)-(8) ВЦФ (4) имеют конкретную содержательную интерпретацию. Веса, приписанные ребрам предфрактального графа  $G_L$ , могут отражать конкретные ограничения (время, расстояние), налагаемые на систему служб (аварийные, пожарные депо, милиейские участки, больницы), так и общие затраты, выражаемые в условных единицах.

На практике число разделения  $s(x)$  может означать, например, расстояние от самого далекого потребителя (дома, квартала, организации) до системы служб.

#### 4. Алгоритм поиска $p$ -центра предфрактального графа

Рассмотрим взвешенный предфрактальный граф  $G_L = (V_L, E_L)$ , смежность старых ребер которого сохраняется, порождающая затравка  $H = (W, Q)$ ,  $|W| = n$ ,  $|Q| = q$ .

Затравка  $H$  является сильно связным графом, то есть каждая вершина достижима из всякой другой вершины. Каждая вершина предфрактального графа  $G_L$  также достижима из всякой другой его вершины. Затравка  $H$  – неориентированный граф, тогда порожденный предфрактальный граф  $G_L$  тоже неориентированный.

Алгоритм  $\alpha_1$  позволяет построить  $p$ -центр предфрактального графа для  $p = n^{L-1}$  вершин. Алгоритм  $\alpha_1$  основан на алгоритме выделения центра на простом графе  $G = (V, E)$ . Центр графа легко может быть получен из матрицы взвешенных расстояний. Метод взвешенных расстояний будет использоваться в качестве процедуры МВР [4,5,6].

Основная идея алгоритма заключается в том, что каждая подграф-затравка  $z_s^{(L)}$ ,  $s = \overline{1, n^{L-1}}$  рассматривается как отдельно взятый граф. Последовательно на каждой из  $n^{L-1}$  подграф-затравке находятся центры  $x_s^{(L)}$ . Поиск центра на отдельно взятой подграф-затравке осуществляется с помощью Метода взвешенных расстояний. Осуществив поиск центров на подграф-затравках  $L$ -го ранга, получим  $p$ -центр  $x_p^*$  предфрактального графа  $G_L$ . Поскольку все локальные центры находятся на подграф-затравках  $L$ -го ранга, полученный  $p$ -центр предфрактального графа по-другому будем называть  $p$ -центром  $L$ -го ранга предфрактального графа  $G_L$  [7].

АЛГОРИТМ  $\alpha_1$ .

ВХОД: взвешенный предфрактальный граф  $G_L = (V_L, E_L)$ .

ВЫХОД:  $p$ -центр предфрактального графа  $G_L$ .

ШАГ 1. Последовательно для каждой подграф-затравки  $z_s^{(L)}$ ,  $s = \overline{1, n^{L-1}}$  найти центр, используя процедуру МВР.

ШАГ 2. На выходе шага 1 получаем  $n^{L-1}$  центров, для затравок  $z_s^{(L)}$ . Объединяя эти центры, получим  $p$ -центр предфрактального графа  $G_L$ .

ПРОЦЕДУРА МВР.

ВХОД: взвешенный граф  $G = (V, E)$ .

ВЫХОД:  $p$ -центр графа  $G$ .

ТЕОРЕМА 1. Алгоритм  $\alpha_1$  выделяет  $p$ -центр предфрактального графа  $G_L$ , смежность старых ребер которого сохраняется.

ДОКАЗАТЕЛЬСТВО. На последнем шаге  $L$  построения предфрактального графа  $G_L$  все вершины замещаются затравкой  $H = (W, Q)$ , тогда каждая вершина  $L$ -ого ранга принадлежит какой-либо подграф-затравке  $z_s^{(L)}$ . Найдя центры на подграф-затравках  $z_s^{(L)}$ ,  $s = \overline{1, n^{L-1}}$ , все вершины предфрактального графа оказываются в области «обслуживания».

На подграф-затравке  $L$ -ого ранга  $z_1^{(L)}$  выделяется центр  $x_1^{(L)}$ , который будет обслуживать  $n$  вершин. Далее выделяется центр  $x_2^{(L)}$  на второй подграф-затравке  $L$ -ого ранга  $z_2^{(L)}$ , который обслуживает еще  $n$  вершин. После выделения центров  $x_s^{(L)}$  на всех подграф-затравках  $z_s^{(L)}$ , всего будет обслуживаться  $n^{L-1} \cdot n = n^L$  вершин. Тогда обслуживается все множество вершин  $G_L$ , объединив  $n^{L-1}$  центров  $x_s^{(L)}$  получим  $p$ -центр  $x_p^*$  предфрактального графа.

Предположим, что выделенный  $p$ -центр предфрактального графа, где число вершин  $p = n^{L-1}$ , не оптимальный. Попробуем улучшить значение числа деления центра на какой-либо подграф-затравке  $L$ -ого ранга  $z_s^{(L)}$ . Переместив центр из выделенной вершины в любую другую вершину подграф-затравки  $z_{s^*}^{(L)}$ , число деления не может уменьшиться на самой подграф-затравке, так как вершина, выделенная алгоритмом, является центром подграф-затравки. Также не уменьшатся значения чисел деления на любых других подграф-затравках, так как на каждой из них найден свой центр.

Переместив центр подграф-затравки  $z_{s^*}^{(L)}$  в вершину другой подграф-затравки  $z_{s^{**}}^{(L)}$ , значение числа деления для подграф-затравки  $z_{s^*}^{(L)}$  увеличивается, так как смежность старых ребер сохраняется и для того, чтобы дойти до вершин  $z_{s^*}^{(L)}$  следует затратить дополнительный путь от  $z_{s^{**}}^{(L)}$  до  $z_{s^*}^{(L)}$ .

Перемещать несколько центров разных подграф-затравок (от двух до  $n^{L-1}$ ) также не имеет смысла, так как условие сохранения смежности старых ребер гарантирует не уменьшение чисел деления, как отдельных подграф-затравок, так и всего предфрактального графа.

фа. Таким образом, для числа  $p = n^{L-1}$  алгоритм  $\alpha_1$  выделяет  $p$ -центр  $x_p^*$  предфрактального графа  $G_L$ . ◀

СЛЕДСТВИЕ 1.1. Для всякого взвешенного центрированного предфрактального графа  $G_L$ , порожденного затравкой-звездой: вершины  $p$ -центра,  $p = n^{L-1}$ , предфрактального графа размещаются в центрах подграф-затравок  $L$ -го ранга. ◀

ТЕОРЕМА 2. Вычислительная сложность алгоритма  $\alpha_1$ , поиска  $p$ -центра предфрактального графа  $G_L$ , смежность старых ребер которого сохраняется, равна  $O(3n^2 \cdot N)$ .

ДОКАЗАТЕЛЬСТВО. Алгоритм  $\alpha_1$  представляет собой, по существу многократное выполнение шага 1, то есть поиск центра на каждой подграф-затравке  $L$ -го ранга. Процедура МВР требует затраты  $n^3 + n^2 + n < 3n^3$  операций. Для поиска центров на всех подграф-затравках  $L$ -го ранга  $z_s^{(L)}$  потребуется в сумме затратить  $3n^3 \cdot n^{L-1} = 3n^2 \cdot N$  операций. Тогда вычислительная сложность алгоритма  $\alpha_1$  равна  $O(3n^2 \cdot N)$ . ◀

ТЕОРЕМА 3. Алгоритм  $\alpha_1$  выделяет  $p$ -центр  $x_5$  на предфрактальном графе  $G_L = (V_L, E_L)$ , оптимальный по первому  $F_1(x_5)$  и третьему  $F_3(x_5)$  критериям, и оцениваемое по второму  $n^{L-1} \cdot \theta^{L-1} a \leq F_2(x_5) \leq n^L \cdot \theta^{L-1} b$  и четвертому  $F_4(x_5) = n^{L-1}$  критериям.

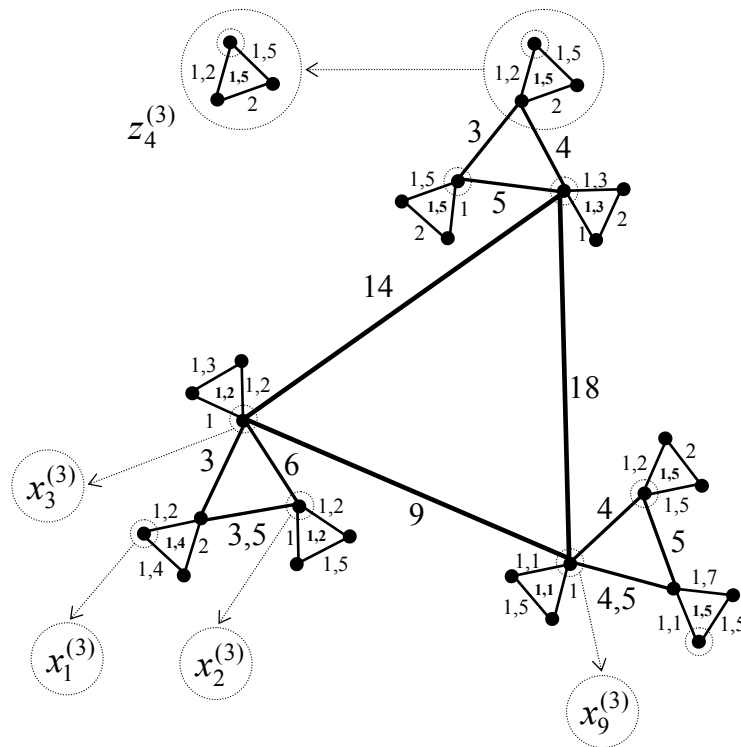
ДОКАЗАТЕЛЬСТВО. Первый критерий  $F_1(x_5)$  минимизирует число разделения  $p$ -центра. Доказательством оптимальности по первому критерию служит теорема 3.14, поскольку на выходе алгоритма  $\alpha_1$  получаем  $p$ -центр предфрактального графа. Посчитаем в каком диапазоне лежит число разделения  $p$ -центра предфрактального графа. Рассмотрим наихудший случай, когда нужно пройти  $(n-1)$  ребер и наилучший случай – одно ребро. Тогда число разделения  $s(x_5)$  оценивается как:  $\theta^{L-1} a \leq s(x_5) \leq (n-1)\theta^{L-1} b$ .

Второй критерий минимизирует сумму радиусов  $p$ -центра. Веса ребер на подграф-затравках  $L$ -го ранга лежат в промежутке  $[\theta^{L-1} a; \theta^{L-1} b]$ , где  $a > 0$ , и  $\theta < \frac{a}{b}$ . Число вершин на затравке равно  $n$ , тогда радиус подграф-затравки  $z_s^{(L)}$  в худшем случае равен  $(n-1) \cdot \theta^{L-1} b$ , в лучшем  $\theta^{L-1} a$ . Всего на предфрактальном графе  $p = n^{L-1}$  центров, тогда суммарный вес радиусов оценивается как:  $n^{L-1} \cdot \theta^{L-1} a \leq F_2(x_5) \leq n^L \cdot \theta^{L-1} b$ .

Третий критерий минимизирует количество типов центров. Так как  $p$ -центр состоит только из центров ранга  $L$ , то критерий принимает свое минимально возможное значение равное одному:  $F_3(x_5) = \hbar = 1$ .

Четвертый критерий  $F_4(x) = p$  минимизирует количество центров предфрактального графа. Алгоритм  $\alpha_1$  выделяет  $p$ -центр предфрактального графа, где  $p = n^{L-1}$ . Тогда  $F_4(x_5) = n^{L-1}$ . ◀





$$G_3 = (V_3, E_3)$$

Рис. 1. Поиск  $p$ -центра предфрактального графа  $G_3$ ,  $p = n^{L-1}$ .

На рис. 1. представлен пример поиска  $p$ -центра предфрактального графа  $G_3$ , где  $p = n^{L-1}$ . Алгоритм  $\alpha_1$  работает только на подграф-затравках 3-го ранга. Внутри подграф-затравок 3-го ранга мелким шрифтом указаны числа разделения  $s(x_{s_3}^{(3)})$  центров  $x_{s_3}^{(3)}$ ,  $s_3 = 1, 2, \dots, 9$ .

Локальные центры  $x_{s_3}^{(3)}$ , расположенные на подграф-затравках 3-го ранга, обведены пунктирными линиями. Множество всех вершин  $\{x_{s_3}^{(3)}\}$  составляет  $p$ -центр  $x_p^*$  предфрактального графа  $G_3$ . Число разделения множества  $x_p^* : s_p(x_p^*) = \max_{s_3=1,2,\dots,9} [s(x_{s_3}^{(3)})] = 1,5$ .

### Параллельный алгоритм $\alpha_1^*$ поиска $p$ -центра предфрактального графа

Параллельные алгоритмы, рассматриваемые в данной работе, построены для PRAM (Parallel Random Access Machine) – модели параллельной вычислительной системы [8].

Далее предлагается параллельный алгоритм выделения  $p$ -центра предфрактального графа. Алгоритм  $\alpha_1^*$  основан на алгоритме выделения  $p$ -центра на простом графе. Центр графа легко может быть получен из матрицы взвешенных расстояний [9,10].

Количество процессоров равно количеству всех затравок  $L$ -ого ранга, то есть  $k = n^{L-1}$ . Тогда каждый из процессоров назначим одной из затравок  $z_s^{(L)}$ ,  $s = \overline{1, n^{L-1}}$ .

Каждая подграф-затравка  $z_s^{(L)}$ ,  $s = \overline{1, n^{L-1}}$  рассматривается как отдельно взятый граф, все  $k$  процессоров параллельно независимо друг от друга находят  $p_s$ -центры. Поиск  $p_s$ -центра на отдельно взятой подграф-затравке осуществляется с помощью процедуры МВР. Осуществив поиск  $p_s$ -центров на подграф-затравках, получим  $p$ -центр всего предфрактального графа  $G_L$ . После чего алгоритм  $\alpha_1^*$  заканчивает свою работу.

#### АЛГОРИТМ $\alpha_1^*$ .

ВХОД: взвешенный предфрактальный граф  $G_L = (V_L, E_L)$ .

ВЫХОД:  $p$ -центр предфрактального графа  $G_L$ .

Шаг 1. Параллельно и независимо друг от друга  $k$  процессоров для каждой затравки  $z_s^{(L)}$ ,  $s = \overline{1, n^{L-1}}$  находят  $p_s$ -центры, используя процедуру МВР.

Шаг 2. На выходе шага 1 получаем  $n^{L-1}$   $p_s$ -центров, для затравок  $z_s^{(L)}$ . Объединяя  $p_s$ -центры, получим  $p$ -центр предфрактального графа  $G_L$ .

ПРОЦЕДУРА МВР.

ВХОД: взвешенный граф  $G = (V, E)$ .

ВЫХОД:  $p$ -центр графа  $G$ .

ТЕОРЕМА 4. *Вычислительная сложность параллельного алгоритма  $\alpha_1^*$ , поиска  $p$ -центра предфрактального графа  $G_L$ , равна  $O(3n^2 \cdot N)$ .*

ДОКАЗАТЕЛЬСТВО. Алгоритм  $\alpha_1^*$  осуществляет параллельный поиск центра на каждой подграф-затравке  $L$ -ого ранга. Процедура МВР требует затраты  $n^3 + n^2 + n < 3n^3$  операций. Для поиска центров на всех подграф-затравках  $L$ -ого ранга  $z_s^{(L)}$  потребуется в сумме затратить  $3n^3 \cdot n^{L-1} = 3n^2 \cdot N$  операций. Тогда вычислительная сложность алгоритма  $\alpha_1^*$  равна  $O(3n^2 \cdot N)$ . ◀

ТЕОРЕМА 5. *Временная сложность параллельного алгоритма  $\alpha_1^*$ , поиска  $p$ -центра предфрактального графа  $G_L$ , равна  $O(3n^2 \cdot N)$ .*

ДОКАЗАТЕЛЬСТВО. Алгоритм  $\alpha_1^*$  осуществляет параллельный поиск центра на каждой подграф-затравке  $L$ -ого ранга. Максимальное время исполнения процедуры МВР одним процессором равно  $n^3 + n^2 + n < 3n^3$ . Поскольку все задействованные процессоры работают

одновременно и независимо друг от друга, временная сложность алгоритма  $\alpha_1^*$  равна  $O(3n^2)$ . ◀

СЛЕДСТВИЕ 5.1. *Время исполнения параллельного алгоритма  $\alpha_1^*$  меньше времени исполнения последовательного алгоритма  $\alpha_1$  в  $N$  раз.* ◀

### Параллельный алгоритм $\alpha_2^*$ поиска внутреннего центра предфрактального графа

Рассмотрим взвешенный предфрактальный граф  $G_L$ , порожденный ориентированной затравкой  $H$ , смежность старых ребер которого сохраняется. Затравка  $H$  является сильно связным графом, то есть каждая вершина достижима из всякой другой вершины. Это означает также, что каждая вершина всего предфрактального графа достижима из всякой другой его вершины.

Алгоритм  $\alpha_2^*$  использует  $k$  процессоров  $Pr_1, Pr_2, \dots, Pr_k$ , где  $k = n^{L-1}$ . Опишем принцип работы алгоритма  $\alpha_2^*$ . Алгоритм начинает свою работу с подграф-затравок  $L$ -го ранга  $z_{s_L}^{(L)}$ ,  $s_L = \overline{1, n^{L-1}}$ . На последнем шаге порождения предфрактального графа  $G_L$  каждая вершина графа  $G_{L-1}$  была замещена затравкой  $H$ . Поскольку при порождении предфрактального графа действует правило сохранения смежности старых ребер, к каждой вершине  $G_{L-1}$  "привязываются" затравки одной из своих вершин. Назначим каждой подграф-затравке  $z_{s_L}^{(L)}$  по одному процессору из  $Pr_{s_L}$ ,  $s_L = 1, 2, \dots, n^{L-1}$ .

Рассмотрим подграф-затравку  $z_1^{(L)}$ , так как затравка  $H$  является сильно связной, то для всякой ее вершины можно найти путь к любой другой. Процессор  $Pr_1$  находит кратчайшие пути от  $(n-1)$  вершин  $v_{j_1}^{(L)}$  подграф-затравки  $z_1^{(L)}$  до "общей" вершины  $x_1^{(L)}$ . Среди кратчайших путей найдем максимальный и определим число  $s_t(x_1^{(L)}) = \max_{j_1=1, n-1} [d(v_{j_1}^{(L)}, x_1^{(L)})]$ , которое назовем *числом внутреннего разделения* вершины  $x_1^{(L)}$  подграф-затравки  $z_1^{(L)}$ .

Таким образом, на первом шаге  $k$  процессоров  $Pr_1, Pr_2, \dots, Pr_k$  параллельно и независимо друг от друга, находят числа внутреннего разделения  $s_t(x_{s_L}^{(L)})$ , каждый на своей подграф-затравке  $L$ -го ранга  $z_{s_L}^{(L)}$ :

$$s_t(x_{s_L}^{(L)}) = \max_{j_L=1, n-1} [d(v_{j_L}^{(L)}, x_{s_L}^{(L)})], \text{ где } d(x_{s_L}^{(L)}, x_{s_L}^{(L)}) = 0.$$

Поиск кратчайших путей осуществляется с помощью известного алгоритма Дейкстры. Алгоритм Дейкстры будет использоваться в качестве процедуры, вызываемой по мере необходимости.

Рассмотрим далее подграф-затравки  $(L-1)$ -го ранга  $z_{s_{L-1}}^{(L)}$ ,  $s_{L-1} = \overline{1, n^{L-2}}$ . Каждая из них в процессе порождения предфрактального графа  $G_{L-1}$  была привязана к вершинам предыдущего в траектории графа  $G_{L-2}$ , так как действует правило сохранения смежности старых ребер. Тогда каждая подграф-затравка  $(L-1)$ -го ранга  $z_{s_{L-1}}^{(L)}$  также имеет одну общую

вершину с соответствующими затравками  $(L-2)$ -го ранга  $z_{s_{L-2}}^{(L)}$ ,  $s_{L-2} = \overline{1, n^{L-3}}$ . Назначим каждой подграф-затравке  $z_{s_{L-1}}^{(L)}$  по одному процессору из  $\text{Pr}_{s_{L-1}}$ ,  $s_{L-1} = \overline{1, n^{L-2}}$ .

На втором шаге  $n^{L-2}$  процессоров  $\text{Pr}_{s_{L-1}}$  параллельно находят числа внутреннего разделения каждый для своей подграф-затравки  $(L-1)$ -го ранга  $z_{s_{L-1}}^{(L)}$ ,  $s_{L-1} = \overline{1, n^{L-2}}$ :  $s_t(x_{s_{L-1}}^{(L-1)}) = \max_{j_{L-1}=1, n-1} [d(v_{j_{L-1}}^{(L-1)}, x_{s_{L-1}}^{(L-1)}) + s_t(x_{s_L}^{(L)})]$ , где  $d(x_{s_{L-1}}^{(L-1)}, x_{s_{L-1}}^{(L-1)}) = 0$ .

То есть, осуществляется поиск кратчайших путей от  $(n-1)$  вершин  $v_{j_{L-1}}^{(L-1)}$  подграф-затравки  $z_{s_{L-1}}^{(L)}$  до общей вершины  $x_{s_{L-1}}^{(L-1)}$ . Далее к длине кратчайшего пути  $d(v_{j_{L-1}}^{(L-1)}, x_{s_{L-1}}^{(L-1)})$  добавляется соответствующее число разделения  $s_t(x_{s_L}^{(L)})$ , найденное для подграф-затравок предыдущего ранга, и среди получившихся сумм выбирается максимальное. Заметим, что  $s_t(x_{s_L}^{(L)})$  - число внутреннего разделения той вершины, от которой начинается путь  $d(v_{j_{L-1}}^{(L-1)}, x_{s_{L-1}}^{(L-1)})$ .

Можно считать, что добавляем  $s_t(x_{s_L}^{(L)}) = s_t(v_{j_{L-1}}^{(L-1)})$ . Сумма  $d(x_{s_{L-1}}^{(L-1)}, x_{s_{L-1}}^{(L-1)}) + s_t(x_{s_L}^{(L)}) = s_t(x_{s_L}^{(L)})$ , так как  $d(x_{s_{L-1}}^{(L-1)}, x_{s_{L-1}}^{(L-1)}) = 0$ .

Указанным способом находим числа внутреннего разделения  $s_t(x_{s_l}^{(l)})$  для общих вершин  $x_{s_l}^{(l)}$  подграф-затравок  $z_{s_l}^{(l)}$ ,  $s_l = \overline{1, n^{l-1}}$  до 2-го ранга включительно, то есть для всех  $l = L, L-1, \dots, 2$ . На каждом шаге  $l = L, L-1, \dots, 2$  подграф-затравкам  $z_{s_l}^{(l)}$  назначаются процессоры  $\text{Pr}_{s_l}$ ,  $s_l = \overline{1, n^{l-1}}$ .

На последнем шаге  $l = 2$  найдены числа разделения  $s_t(x_{s_2}^{(2)})$ ,  $s_2 = \overline{1, n}$  для общих вершин  $x_{s_2}^{(2)}$  подграф-затравок 2-го ранга  $z_{s_2}^{(2)}$  и одной подграф-затравки первого ранга  $z_{s_1}^{(1)} = z_1^{(1)}$ . Подграф-затравка  $z_1^{(1)}$ , по сути, соответствует графу  $G_1$  из траектории  $G_1, G_2, \dots, G_L$ . Тогда для каждой вершины подграф-затравки  $z_1^{(1)}$  найдено число  $s_t(x_{s_2}^{(2)})$ ,  $s_2 = \overline{1, n}$ .

Далее рассматриваем подграф-затравку  $z_1^{(1)}$ , которой назначим процессор  $\text{Pr}_1$ . Процессор  $\text{Pr}_1$  находит для каждой ее вершины  $x_{s_1}^{(1)}$  число внутреннего разделения:  $s_t(x_{s_1}^{(1)}) = \max_{j_1=1, n-1} [d(v_{j_1}^{(1)}, x_{s_1}^{(1)}) + s_t(x_{s_2}^{(2)})]$ . Вершина  $x_o^*$ , для которой число внутреннего разделения  $s_t(x_t^*)$  минимальное является внутренним центром предфрактального графа  $G_L$ :  $s_t(x_t^*) = \min_{s_1=1, 2, \dots, n} [s_t(x_{s_1}^{(1)})]$ .

Представим далее алгоритм  $\alpha_2^*$ , где для поиска кратчайшего пути между двумя любыми вершинами графа используется ПРОЦЕДУРА ДЕЙКСТРЫ.

АЛГОРИТМ  $\alpha_2^*$ .

ВХОД: взвешенный предфрактальный граф  $G_L = (V_L, E_L)$ .

ВЫХОД:  $x_t^*$  - внутренний центр предфрактального графа  $G_L$ .

ШАГ 1. (1) Назначим каждый из  $k = n^{L-1}$  процессоров  $Pr_1, Pr_2, \dots, Pr_k$  подграф-затравкам  $z_{s_L}^{(L)}$ ,  $s_L = \overline{1, n^{L-1}}$ . Каждый процессор будет обрабатывать только назначенную ему подграф-затравку.

(2) Одновременно  $k$  процессоров  $Pr_1, Pr_2, \dots, Pr_k$  параллельно и независимо друг от друга находят числа внутреннего разделения  $s_t(x_{s_L}^{(L)})$ , каждый на назначенной ему подграф-затравке  $L$ -го ранга  $z_{s_L}^{(L)}$ :  
 $s_t(x_{s_L}^{(L)}) = \max_{j_L=1, n-1} [d(v_{j_L}^{(L)}, x_{s_L}^{(L)})]$ , где  $d(x_{s_L}^{(L)}, x_{s_L}^{(L)}) = 0$ . Поиск кратчайших путей между вершинами осуществляется с помощью процедуры Дейкстры.

Для всех  $l = L-1, L-2, \dots, 2$  выполнить:

ШАГ  $L-l+1$ . (1) Назначим каждый из  $n^{l-1}$  процессоров  $Pr_{s_l}$  подграф-затравкам  $z_{s_l}^{(l)}$ ,  $s_l = \overline{1, n^{l-1}}$ . Каждый процессор будет обрабатывать только назначенную ему подграф-затравку.

(2) Одновременно  $n^{l-1}$  процессоров  $Pr_{s_l}$  параллельно и независимо друг от друга находят числа внутреннего разделения  $s_t(x_{s_l}^{(l)})$ , каждый на назначенной ему подграф-затравке  $l$ -го ранга  $z_{s_l}^{(l)}$ :  
 $s_t(x_{s_l}^{(l)}) = \max_{j_l} [d(v_{j_l}^{(l)}, x_{s_l}^{(l)}) + s_t(x_{j_l}^{(l+1)})]$ , где  $d(x_{s_l}^{(l)}, x_{s_l}^{(l)}) = 0$ . Поиск кратчайших путей между вершинами осуществляется с помощью процедуры Дейкстры.

ШАГ  $L$ . (1) Назначим каждый из  $n$  процессоров  $Pr_{s_1}$  подграф-затравку первого ранга  $z_1^{(1)}$ ,  $s_1 = \overline{1, n}$ . Каждый процессор будет работать с подграф-затравкой  $z_1^{(1)}$  как с отдельным графом.

(2) Одновременно  $n$  процессоров  $Pr_{s_1}$  параллельно и независимо друг от друга находят числа внутреннего разделения  $s_t(x_{s_1}^{(1)})$ , каждый на назначенной ему подграф-затравке первого ранга  $z_1^{(1)}$ :  
 $s_t(x_{s_1}^{(1)}) = \max_{j_1=1, n-1} [d(v_{j_1}^{(1)}, x_{s_1}^{(1)}) + s_t(x_{j_1}^{(2)})]$ , где  $d(x_{s_1}^{(1)}, x_{s_1}^{(1)}) = 0$ . Поиск кратчайших путей между вершинами осуществляется с помощью процедуры Дейкстры.

ШАГ  $L+1$ . Используя процессор  $Pr_1$  из всех вершин  $x_{s_1}^{(1)}$ ,  $s_1 = 1, 2, \dots, n$  в качестве внутреннего центра предфрактального графа  $G_L$  выбрать вершину  $x_t^*$  с

наименьшим числом разделения:  $s(x_t^*) = \min_{s_1=1,2,\dots,n} [s_t(x_{s_1}^{(1)})]$ .

ПРОЦЕДУРА ДЕЙКСТРЫ.

ВХОД: взвешенный граф  $G = (V, E)$ .

ВЫХОД: кратчайшее расстояние  $d(v_i, v_j)$ .

ТЕОРЕМА 6. *Вычислительная сложность алгоритма  $\alpha_2^*$  равна  $O(4n^2 \cdot N)$ .*

ДОКАЗАТЕЛЬСТВО. На первом шаге алгоритм работает на подграф-затравках  $L$ -го ранга  $z_{s_L}^{(L)}$ , где находит числа внутреннего разделения вершин  $x_{s_L}^{(L)}, s_L = 1, 2, \dots, n^{L-1}$ . Поиск числа внутреннего разделения на отдельно взятой подграф-затравке состоит из: 1) поиска кратчайших путей  $d(v_{j_L}^{(L)}, x_{s_L}^{(L)})$  и 2) выбора максимального из кратчайших путей. Поиск кратчайших путей осуществляется с помощью процедуры Дейкстры, вычислительная сложность которого равна  $n^2$  [2], а выбор максимального элемента или в худшем случае сортировка элементов по возрастанию требует выполнения также  $n^2$  операций. Тогда поиск числа внутреннего разделения на одной подграф-затравке требует в сумме  $2n^2$  операций. Число всех подграф-затравок  $L$ -го ранга равно  $n^{L-1}$ , для выполнения шага 1 или поиска всех  $s_t(x_{s_L}^{(L)}), s_L = 1, 2, \dots, n^{L-1}$  требуется  $2n^2 \cdot n^{L-1} = 2n^{L+1}$  операций.

На следующем шаге осуществляется поиск чисел внутреннего разделения для вершин  $x_{s_{L-1}}^{(L-1)}, s_{L-1} = 1, 2, \dots, n^{L-2}$ , что требует  $2n^2 \cdot n^{L-2} = 2n^L$  операций. Продолжая поиск внутренних центров вершин до второго ранга включительно получаем:  $2n^{L+1} + 2n^L + 2n^{L-1} + \dots + 2n^3$ . На шаге  $L$  осуществляется поиск кратчайших путей попарно между всеми вершинами  $x_{s_1}^{(1)}, s_1 = 1, 2, \dots, n$ , что требует  $n^2 \cdot n = n^3$  операций плюс поиск максимального элемента на шаге  $L+1$ .

В сумме получаем:

$$\begin{aligned} & 2n^{L+1} + 2n^L + 2n^{L-1} + \dots + 2n^3 + n^3 + n^2 \leq \\ & \leq 2n^{L+1} + 2n^L + 2n^{L-1} + \dots + 2n^3 + 2n^2 + n^3 = 2 \cdot \frac{n^{L+1} \cdot n - n^2}{n-1} + n^3 \leq \\ & \leq 2n^{L+2} - n^2 + n^3 \leq 2n^{L+2} + n^3 \leq 2n^{L+2} + 2n^{L+2} = 4n^2 \cdot n^L = 4n^2 \cdot N. \end{aligned}$$

Таким образом, вычислительная сложность алгоритма  $\alpha_2^*$  равна  $O(4n^2 \cdot N)$ . ◀

Отметим, что в качестве процедуры ДЕЙКСТРЫ можно использовать другие известные последовательные алгоритмы поиска кратчайших путей, тогда будет меняться и вычислительная сложность алгоритма  $\alpha_2^*$ .

ТЕОРЕМА 7. *Временная сложность алгоритма  $\alpha_2^*$  равна  $O(4n^3 \cdot L)$ .*

ДОКАЗАТЕЛЬСТВО. На первом шаге алгоритм  $\alpha_2^*$  находит числа внутреннего разделения на подграф-затравках  $L$ -го ранга  $z_{s_L}^{(L)}, s_L = 1, n^{L-1}$ . На каждой подграф-затравке работает назначенный ей процессор  $Pr_{s_L}$ . Поиск числа внутреннего разделения на отдельно взятой подграф-затравке состоит из: 1) поиска кратчайших путей до вершины  $x_{s_1}^{(L)}$  от других вершин подграф-затравки и 2) выбора максимального из кратчайших путей. Поиск кратчай-

ших путей осуществляется с помощью процедуры Дейкстры, вычислительная сложность которого равна  $n^2$ , а выбор максимального элемента требует выполнения также  $(n - 1)$  операций. Тогда поиск числа внутреннего разделения на одной подграф-затравке требует в сумме не более  $2n^2$  операций. Поскольку процессоры работают параллельно, первый шаг займет  $2n^2$  времени.

На втором шаге каждой подграф-затравке  $(L - 1)$ -го ранга  $z_{s_{L-1}}^{(L-1)}$ ,  $s_L = \overline{1, n^{L-2}}$ , также назначается процессор  $Pr_{s_{L-1}}$ . Поиск числа внутреннего разделения на одной подграф-затравке требует  $2n^2$  операций. Поскольку процессоры работают параллельно, второй шаг также займет  $2n^2$  времени.

Таким образом, осуществляя поиск чисел внешнего разделения на подграф-затравках разных рангов, на каждом шаге до  $(L - 1)$ -го шага включительно требуется  $2n^2$  времени. Сложив время, потраченное на всех  $(L - 1)$  шагах, получаем  $2n^2 \cdot L$ .

На шаге  $L$  осуществляется поиск кратчайших путей попарно между всеми вершинами  $x_{s_1}^{(1)}$ ,  $s_1 = 1, 2, \dots, n$ , что требует  $n^2$  времени, а шаг  $(L + 1)$  требует затраты  $(n - 1)$  времени на поиск максимального элемента.

Складывая время необходимое для поиска внутреннего центра предфрактального графа, получаем:  $2n^2 \cdot L + n^2 + (n - 1) \leq 2n^2 \cdot (L + 1)$ . Таким образом, время исполнения алгоритма  $\alpha_2^*$  равно  $O(2n^2 \cdot (L + 1))$ . ◀

## 5. Заключение

В данной работе представлена многокритериальная постановка задачи размещения  $p$ -центра на предфрактальном графе. Подробно даны определения предфрактального и фрактального графов. Для решения задачи размещения разработан алгоритм  $\beta_1$  поиска  $p$ -центра на предфрактальном графе. Посчитана вычислительная сложность алгоритма  $\beta_1$ .

Даны оценки критериев (5)-(8) многокритериальной задачи (4). Критерии  $F_1(x_5)$  и  $F_3(x_5)$  являются оптимальными, по другим критериям вычислены ограничения.

Вычислительная сложность, предложенных алгоритмов, меньше чем у известных алгоритмов.

## Литература

1. Кристофидес Н. Теория графов. Алгоритмический подход. – М.: Мир, 1978.
2. Кочкаров А.М. Распознавание фрактальных графов. Алгоритмический подход. – Нижний Архыз: РАН САО, 1998.
3. Майника Э. Алгоритмы оптимизации на графах и сетях. – М.: Мир, 1981.
4. Асанов М.О., Баранский В.А., Расин В.В. Дискретная математика: графы, матроиды, алгоритмы. – Ижевск: НИЦ "РХД", 2001.
5. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. – М.: Мир, 1982.
6. Емеличев В.А., Мельников О.И., Сарванов В.И., Тышкевич Р.И. Лекции по теории графов. – М.: Наука, 1990.

7. *Узденов А.А., Кочкаров Р.А.* Алгоритм поиска внешне-внутреннего центра предфрактального графа с сохранением смежности старых ребер // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. – СПб.: СПбГПУ, 2010. - № 3(101). – С. 145-149.
8. *Воеводин В.В., Воеводин Вл.В.* Параллельные вычисления. – СПб.: БХВ-Петербург, 2002.
9. *Кочкаров А.А., Кочкаров Р.А.* Параллельные алгоритмы на предфрактальных графах: Препринт ИПМ им. М.В. Келдыша РАН № 84. М., 2003. 20 с.
10. *Кочкаров А.А., Кочкаров Р.А.* Параллельный алгоритм поиска кратчайшего пути на предфрактальном графе // Журнал вычисл. матем. и матем. физики. – 2004. – Т. 44, № 6. – С. 1157-1162.