

УДК 519.1

UDC 519.1

**МНОГОКРИТЕРИАЛЬНАЯ ПОСТАНОВКА  
ЗАДАЧИ ВЫБОРА ПРОЕКТОВ ЦЕЛЕВЫХ  
ПРОГРАММ**

**MULTICRITERIA PROBLEM DEFINITION OF  
A SELECTION OF PROJECTS TARGETED  
PROGRAMS**

Кунижева Лариса Адамовна  
*Северо-Кавказская государственная  
гуманитарно-технологическая академия,  
Черкесск, Россия*

Kunizheva Larisa Adamovna  
*North Caucasian State Technological Academy of the  
Humanities, Cherkessk, Russia*

Кочкаров Расул Ахматович  
к.ф.-м.н., доцент  
*Заместитель директора по информационным  
технологиям, Финансовый университет при  
правительстве РФ, Москва, Россия*

Kochkarov Rasul Ahmatovich  
Dr.Sci.Phys.–Math., associate professor  
*Deputy director for information technologies,  
Financial university at the government of the RF,  
Moscow, Russia*

В статье рассмотрена экономико-математическая модель выбора проектов целевых программ на примере Российской Федерации. Определены критерии оптимальности их распределения. Предложен параллельный алгоритм с гарантированными оценками выделения  $2l$  дольного графа

In the article, we have considered the economic-mathematical model of selection of projects targeted programs on the example of the Russian Federation. The article also defines criteria of their distribution. We have proposed the parallel algorithm with the guaranteed estimates of the allocation of  $2l$  graph

Ключевые слова: ЦЕЛЕВЫЕ ПРОГРАММЫ, ЗАТРАВКА, ФРАКТАЛЬНЫЙ И ВЗВЕШЕННЫЙ ГРАФЫ, ПАРАЛЛЕЛЬНЫЙ АЛГОРИТМ

Keywords: TARGET PROGRAMS, PRIMER, FRACTAL AND WEIGHED GRAPH, PARALLEL ALGORITHM

В настоящее время Российская Федерация входит в состав ВТО, в связи с чем, для устойчивого развития, для надежности, для стойкости [1, 2] появляется необходимость поддержания регионов страны в разных направлениях экономики (промышленность, сельское хозяйство, торговля и т.д.). Государство выделяет деньги под так называемые целевые программы [3]. Каждая государственная целевая программа содержит в своем тексте перечень целевых показателей и индикаторов. Целевые показатели и индикаторы позволяют контролировать ход реализации или текущую результативность программы на этапах, а после завершения реализации - оценивать результативность, успешность выполнения программы [4].

Набор целевых показателей и индикаторов, или по – другому, система целевых показателей и индикаторов, описывает состояние программы в ходе ее исполнения, но никак не влияет на исполнение.

Чтобы влиять на исполнение целевой программы на стадии распределения и выбора проектов, необходимо построение экономико - математической модели (ЭММ). ЭММ позволяет формализовать некоторые целевые показатели, которые потом можно оптимизировать. Для построения ЭММ распределения проектов целевых программ опишем структуру распределения средств на целевые программы на примере РФ.

Средства на целевые программы распределяются первоначально по Федеральным Округам, на следующем этапе Федеральный Округ распределяет эти средства по субъектам Округа, далее по городам и районам субъекта. Вследствие этого возникает вопрос, как оптимально распределить эти средства, как оценить результативность исполнения проекта целевой программы.

Аналогичная постановка задачи распределения средств возникает в Евросоюзе, где средства распределяются по государствам – членам Евросоюза, а далее по регионам этих государств.

Подобная задача появляется при реализации крупного научного проекта. Любой крупный научно-исследовательский проект делится на части, и предлагается для реализации научно-исследовательским институтам (НИИ) и научно-производственным объединениям (НПО). Часто, эти учреждения не в состоянии справиться с поставленными задачами в одиночку, и поэтому выбирают среди всевозможных учреждений аналогичного профиля наиболее полезные с точки зрения сотрудничества. Каждое из учреждений высшего уровня делит свою часть проекта и распределяет по ведомственным или проектным институтам, т.е. учреждениям более низкого по масштабам выполняемой работе уровня. Всем этим учреждениям приходится искать удовлетворяющие поставленным требованиям учреждения для сотрудничества уже из своего уровня,

Возвращаясь к задаче оптимального распределения средств на проекты целевых программ, необходимо определить критерии оптимальности их распределения: выделяемые средства необходимо распределить так, чтобы минимизировать суммарное время их освоения, максимизировать объем освоенных средств, уменьшить дополнительные расходы, связанные с созданием рабочих мест.

Чтобы ответить на поставленный вопрос построим математическую модель распределения средств. Для построения математической модели определим инструментарий. Введем необходимые определения и понятия.

Термином *затравка* [5] условимся называть какой-либо связный граф  $H = (W, Q)$ . Для определения *фрактального (предфрактального) графа* [5] 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  $\tilde{V} = \{\tilde{v}_j\} \subseteq V$ ,  $j = \overline{1, |\tilde{V}|}$ , смежных ей вершин. Далее из графа  $G$  удаляется вершина  $\tilde{v}$  и все инцидентные ей ребра. Затем каждая вершина  $\tilde{v}_j \in \tilde{V}$ ,  $j = \overline{1, |\tilde{V}|}$  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  $l = \overline{1, L-1}$  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.

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.

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

Потенциальных исполнителей освоения средств будем называть объектами (на первом этапе – это Федеральные Округа, на следующем – субъекты Округа, далее города и районы субъекта). Каждому исследуемому объекту ставим в соответствие вершину  $v_i, i = \overline{1, n}$  графа  $H = (W, Q)$ . Тогда на первом шаге распределения средств Федеральным Округам (или государствам Евросоюза, или научно - исследовательским институтам) поставим в соответствие вершины  $v_i, i = \overline{1, n}$ ,  $v_i \in H = (W, Q)$ . Вершины  $v_i$  и  $v_j$  будут соединены ребром  $e_{ij}$ , если между объектами, соответствующими этим вершинам могут существовать взаимоотношения (под взаимоотношениями будем понимать ситуацию, когда исследуемый объект может подключать для выполнения программы другой объект по договору). Если такое взаимоотношение существует, и если  $v_i$  исполнитель подключает  $v_j$  исполнителя, то будем говорить, что между  $v_i$  и  $v_j$  существует ресурс  $a_{ij}^1, i = \overline{1, n}, j = \overline{1, n}$ . Под ресурсом будем понимать скорость освоения выделенных средств, объемы предварительных договоров на поставку техники для исполнения тех или иных работ. Ребрам  $e_{ij}^1$  соответствуют ресурсы  $a_{ij}^1$  из промежутка  $[a, b]$ , которые определяются из

практических соображений. Вершинам  $v_i^1 \in V_1$  графа  $G_1 = (V_1, E_1)$  приписываем веса  $b_i^1, t=1, \dots, n, b_i^1 \in [a, b]$  (собственный ресурс исполнителя).

Таким образом, на первом шаге распределения средств получаем вершинно - реберно - взвешенный граф  $G_1$ . На втором шаге каждый объект, в свою очередь, является распределителем средств по субъектам Федерального Округа (аналогично распределителем средств будет государство Евросоюза или научно- исследовательский институт). То есть вместо каждой вершины  $v_i^1 \in V_1$  на последующем шаге графа  $G_1 = (V_1, E_1)$  будем рассматривать граф  $G_2^i = (V_2^i, E_2^i) = H = (W, Q), i = \overline{1, n}$ , определенный с той лишь разницей, что ресурсы  $a_{ij}^i$  будут взяты из промежутка  $[qa, qb], 0 \leq q \leq 1$ , а в вершины  $v_j^i \in V_2$  графа  $G_2^i$  входят ребра  $e_{ij}^1$  предыдущего ранга. Как и на предыдущем шаге, вершине  $v_j^i \in V_2^i$  приписываем вес  $b_j^2 \in [qa, qb], 0 \leq q \leq 1$  причем  $\sum b_j^2 = b_j^1$ . Таким образом, получаем взвешенный граф  $G_2$ . Инцидентность ребер первого ранга к вершинам  $G_2^i$  определяется из практических соображений. На  $l$ - том шаге  $l = \overline{1, L}$  ( $L$  определяем из практических соображений), проводя аналогичные рассуждения, получаем взвешенный граф  $G_l$ , с той лишь разницей, что ребра  $e_{ij}^l$  взвешиваются из интервала  $[q^{l-1}a, q^{l-1}b], 0 \leq q \leq 1$ , а вершинам  $v_j^i \in V_l^i, i = \overline{1, n}, l = \overline{1, L}$  приписываются веса  $b_j^l$ , в точности также, как и на предыдущих шагах.

Таким образом, мы получили структуру последовательного распределения средств по объектам.

Следующим этапом формализации задачи является определение решения задачи, то есть введения переменной, с помощью которой мы будем математически записывать критерии оптимальности. Поэтому введем понятие допустимого решения. Допустимым решением  $x$  поставленной задачи является  $2l$  - дольный граф  $D = (V_1^x, V_2^x, \dots, V_l^x, V_1, V_2, \dots, V_l, E_x)$  на графе  $G_l = (V_l, E_l)$ ,

где

$$V_1^x \cup V_2^x \cup \dots \cup V_l^x \cup V_1 \cup V_2 \cup \dots \cup V_l = V,$$

$$V_1^x \cap V_2^x \cap \dots \cap V_l^x \cap V_1 \cap V_2 \cap \dots \cap V_l = \emptyset$$

$$E_x \subset E$$

Множество всех допустимых решений обозначим через  $X$ . На множестве  $X$  определим критерии  $F_1(x), F_2(x), \dots, F_5(x), F_6(x)$ :

$$F_1(x) = V / \left( \sum_{ij \in E_x} a_{ij} + \sum_{V_i \in V_n^x, n=1, l} b_i \right) \rightarrow \min \quad (1)$$

(общее время освоения средств, выделенных на проекты);

$$F_2(x) = \sum_{e_{ij} \in E_x} t_{ij} a_{ij} + \sum_{V_i \in V_n^x, n=1, l} t_i b_i \rightarrow \max \quad (2)$$

(объем освоенных средств), где  $t_{ij}$  - время освоения или исполнения объема на ребре  $e_{ij}$ ,  $t_i$  - время освоения объема на вершине  $v_i$  )

$$F_3(x) = \sum_{n=1}^l |V_n^x| \rightarrow \min \quad (3)$$

(количество главных исполнителей);

$$F_4(x) = |x| \rightarrow \min \quad (4)$$

(количество компонент);

$$F_5(x) = \max_{v_i \in V_n^x} \deg_{V_n^x} v_i \rightarrow \min \quad (5)$$

(степень вершин указывает на количество соисполнителей у главного исполнителя).

$$F_6(x) = |E_x| \rightarrow \max \quad (6)$$

(общее количество связей).

Решением задачи (1) - (6) является Парето-Оптимальное (ПО) [6] множество  $\tilde{X}$  из  $X$ .

На практике структура объекта связей исследуемой задачи, математическую модель которой мы построили, представляет собой предфрактальный граф, порожденный различными затравками.

Математически, в таком виде задача сильно усложняется, поэтому мы рассмотрим более упрощенный вариант этой задачи. В дальнейшем, оговаривая каждый раз, мы будем рассматривать предфрактальный граф, порожденный одной, двумя или несколькими затравками.

Пусть дан предфрактальный граф  $G_L = (V_L, E_L)$ , порожденный одной затравкой, для которого будем искать решение поставленной задачи. Различие исследуемой задачи и модели, которую мы предлагаем, заключается в том, что на каждом шаге мы берем некоторую усложненную затравку порождения, хотя на практике на каждом шаге могут быть различные затравки порождения.

Под решением  $x$  поставленной задачи будем понимать выделение на предфрактальном графе  $G_L = (V_L, E_L)$  суграфа  $G^x = (V_L, E_L^x)$ , где каждая компонента связности графа  $G^x$  является  $2l$ -дольным графом,  $l = \overline{1, L}$ .

Для решения поставленной задачи в данной работе предлагается параллельный алгоритм  $a_1$  с гарантированными оценками.

### **Параллельный алгоритм $a_1$ выделения $2l$ -дольного графа**

Рассмотрим взвешенный предфрактальный граф  $G_L = (V_L, E_L)$ , порожденный затравкой  $H = (W, Q)$  и  $k$  процессоров  $p_1, p_2, \dots, p_k$ , где  $k = \overline{1, n^{L-1}}$ .

Параллельный алгоритм  $a_1$  выделения  $2l$ -дольного графа основан на процедуре  $b$ , которая выделяет двудольный граф, состоящий из двух частей.

Процедура  $b$  состоит из двух шагов.

На первом шаге выделяем максимальное паросочетание максимального веса (МПМВ) используя алгоритм, предложенный Эдмондсом [7].

Второй шаг состоит в следующем.

В МПВП входят множество ребер  $M = \{e_{ij} | e_{ij} \in E\}$ . Разобьем вершины  $v_i \in V_L$  на множества  $W_1, W_2, W_3$  по принципу: одну вершину, инцидентную ребру  $e_{ij}$  включим во множество  $W_1$ , другую - во множество  $W_2$ , оставшиеся вершины графа  $G_L$  включим во множество  $W_3$ . Множества  $W_1, W_2$  и выделенное множество ребер  $M$  задают некоторый двудольный граф  $D_0 = (W_1, W_2, M)$ . Ребра  $e_{ij}'$ , инцидентные вершинам множества  $W_1$  и  $e_{ij}''$ , инцидентные вершинам  $W_2$  удаляем.

Множество вершин  $w_i \in W_3, i = \overline{1, m}, m = |W_3|$  объединяем с множеством  $W_1$  или  $W_2$  двудольного графа  $D_0$  по принципу:

1. Если все вершины  $w_i \in W_3, i = \overline{1, m}, m = |W_3|$  смежны с вершинами только множества  $W_1$  или только множества  $W_2$ , то множество вершин  $w_i \in W_3$  объединяем с множеством  $W_1$  или множеством  $W_2$  соответственно. При этом инцидентные им ребра сохраняем.

2. Вершины  $w_i \in W_3, i = \overline{1, m}, m = |W_3|$  могут быть смежны с вершинами и множества  $W_1$  и множества  $W_2$ . Тогда считаем сумму весов  $a_{ij}$  ребер  $e_{ij}$ , инцидентных этой вершине, смежных с вершинами множества  $W_1$  и аналогичную сумму соответственно для множества  $W_2$ . Выбираем ребра  $e_{ij}$ , суммарный вес которых максимален. Включая эти ребра в двудольный граф  $D_0$ , получая тем самым новый двудольный граф  $D$ , определяем принадлежность этих вершин к долям  $W_1, W_2$ .

Опишем принцип работы алгоритма  $a_1$ .

Количество процессоров равно количеству всех затравок  $L$ -ого ранга, то есть  $k = n^{L-1}$ . Тогда каждый из процессоров  $p_s$  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  $p_1, p_2, \dots, p_k$  параллельно независимо друг от

друга находят двудольный граф  $D^l$ , каждый на назначенной ему заправке  $z_s^{(L)}$ . Поиск двудольного графа на отдельно взятой подграф-заправке осуществляется с помощью процедуры  $b$ . Далее каждую подграф-заправку  $z_s^{(L)}$ ,  $s = \overline{1, n^{L-1}}$  стягиваем в одну вершину. Тогда мы получаем новый предфрактальный граф  $G_{L-1} = (V_{L-1}, E_{L-1})$ . Аналогично предыдущему шагу, каждый процессор назначаем на одну из заравок  $z_{s-1}^{(L)}$ ,  $s = \overline{1, n^{L-1}}$ . Аналогично, применяя процедуру  $b$ , находим двудольный граф  $D_s^{l-1}, s = \overline{1, n^{l-2}}$ . Общий  $l$ -тый шаг работы алгоритма представляет собой выделение двудольных графов  $D_s^l, s = \overline{1, n^{L-l+1}}, l = \overline{1, L}$  на заправках  $z_s^{(L)}$ ,  $s = \overline{1, n^{L-1}}$ .

Далее на предфрактальном графе  $G_L = (V_L, E_L)$  работа алгоритма на каждом  $l$ -том шаге заключается в объединении выделенных двудольных графов по принципу смежности вершин. Объединение двудольных графов  $D_s^{l-1}, s = \overline{1, n^{l-2}}$  будем обозначать  $\tilde{D}$ .

Алгоритм заканчивает свою работу на  $L$ -м шаге, когда на последней исходной заправке выделяем двудольный граф. Тем самым, объединив все выделенные  $D^l$  двудольные заправки на всех шагах работы алгоритма, получим  $2l$ -дольный граф.

#### АЛГОРИТМ $a_1$

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

ВЫХОД:  $2l$ -дольный граф  $D$

ШАГ 1. Параллельно и независимо друг от друга  $k$  процессоров  $p_1, p_2, \dots, p_k$  для каждой заправки  $z_s^{(L)}$ ,  $s = \overline{1, n^{L-1}}$  находят двудольный граф  $D$  с помощью процедуры  $b$ . На шаге 1 получаем двудольный граф максимальной заправки.

ШАГ 2. Объединяя эти двудольные графы получаем покрытие предфрактального графа  $G_L = (V_L, E_L)$  двудольными графами, которое дает в объединении  $2l$ -дольный граф.

ПРОЦЕДУРА  $b$ .

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

ВЫХОД: двудольный граф  $D$ .

**ТЕОРЕМА 1.** Алгоритм  $a_1$  выделяет  $2l$ -дольный граф  $D$  ( $l = \overline{1, n^{L-1}}$ ) на предфрактальном  $(n, L)$ -графе  $G_L = (V_L, E_L)$ , порожденном затравкой  $H = (W, Q)$ ,  $|W| = n$ ,  $|Q| = n$ , с коэффициентом подобия  $q < \frac{a}{b}$ .

ДОКАЗАТЕЛЬСТВО. На первом шаге  $k$  процессоров находят двудольные  $D_s^l$  графы, используя процедуру  $b$ .

На последнем шаге  $L$  построения предфрактального графа  $G_L = (V_L, E_L)$  все вершины замещаются затравкой  $H = (W, Q)$ , тогда каждая вершина  $L$ -ого ранга принадлежит какой-либо подграф-затравке  $z_s^{(L)}$ . Найдя двудольный граф  $D_L$  на затравке  $z_s^{(L)}$ , покроем все вершины, принадлежащие этой затравке.

Таким образом, на  $L$ -том шаге процессоры  $p_1, p_2, \dots, p_k$  с помощью процедуры  $b$  выделили двудольные графы на всех затравках.

Дальнейшая работа алгоритма: на шаге  $L-1$  используя те же процессоры, закрепляя их затравках предпоследнего ранга, применяя процедуру  $b$  к каждой затравке  $z_i^{l-2}$ , опять получим двудольный граф. Будем помнить, что каждая вершина этой затравки  $z_s^{l-1}$ ,  $s = 1, 2, \dots, L-2$  содержит внутри себя двудольный граф. Тогда если мы развернем эти двудольные графы, находящиеся внутри затравки  $z_s^{l-1}$ , то у нас в покрытии получатся  $4x$ -дольные графы. Продолжая этот процесс, на  $l$ -том шаге ( $l = \overline{1, L}$ ) получаем, что, на затравках  $z_s^{L-l+1}$  применяя тот же принцип распределения процессоров и процедуру  $b$ , опять получаем двудольные

графы. Тогда на последнем шаге будет выделена последняя двудольная затравка и, объединяя выделенные затравки мы получим  $2l$ -дольный граф ( $k = \overline{1, n^{L-1}}$ ). Тем самым мы доказали, что алгоритм  $a_1$  выделяет  $2l$ -дольный граф. ◀

**ТЕОРЕМА 2.** *Вычислительная сложность параллельного алгоритма  $a_1$  выделения  $2l$ -дольного графа ( $l = \overline{1, n^{L-1}}$ )  $D$  на предфрактальном  $(n, L)$ -графе  $G_L = (V_L, E_L)$ , порожденном затравкой  $H = (W, Q)$ ,  $|W| = n$ ,  $|Q| = n$ , где  $|V_L| = N = n^L$ , равна  $O(Nn^3)$ .*

**ДОКАЗАТЕЛЬСТВО.** Алгоритм  $a_1$  представляет собой, по существу многократное выполнение шага 1, т.е. поиск двудольного графа для каждой подграф-затравки, а их  $n^{L-1}$ . Шаг 1 потребует выполнения  $O(n^4)$  операций на каждой подграф-затравке – столько операций выполняет алгоритм Эдмондса.

Тогда  $O(n^{L-1}n^4) = O(n^L n^3) = O(Nn^3)$ . Таким образом, вычислительная сложность алгоритма  $a_1$  равна  $O(Nn^3)$ .

Сравнив вычислительную сложность  $O(Nn^3)$  параллельного алгоритма  $a_1$  с вычислительной сложностью  $O(N^4)$  алгоритма Эдмондса, получим:  $O(Nn^3) < O(N^4)$ , то есть ускорение вычисления параллельного алгоритма  $a_1$  относительно алгоритма Эдмондса  $\sim N^3$ . ◀

**ПРИМЕЧАНИЕ 1.** *Вычислительная сложность параллельного алгоритма  $a_1$  меньше вычислительной сложности алгоритма Эдмондса, в  $N^3$  раз на предфрактальном  $(n, L)$ -графе  $G_L = (V_L, E_L)$ .* ◀

**ТЕОРЕМА 3.** *Временная сложность параллельного алгоритма  $a_1$ , где имеется  $k$  процессоров  $p_1, p_2, \dots, p_k$ ,  $k = n^{L-1}$ , равна  $O(n^4)$ .*

**ДОКАЗАТЕЛЬСТВО.** Алгоритм  $a_1$  выполняет шаг 1 параллельно, каждым из  $l$  процессоров. Для выполнения шага 1 потребуется  $O(n^4)$  [7]

времени (столько времени требуют процедура Эдмондса). Но все процессоры работают параллельно, то есть они закончат свою работу: «выполнение шага 1» в худшем случае за время  $O(n^4)$ . Таким образом, временная сложность алгоритма  $a_1$  равна  $O(n^4)$ . ◀

Сравнив временную сложность  $O(n^4)$  параллельного алгоритма  $a_1$  с вычислительной сложностью  $O(N^4)$  алгоритма Эдмондса, получим:  $O(n^4) < O(N^4)$ .

**ПРИМЕЧАНИЕ 2.** *Временная сложность параллельного алгоритма  $a_1$  меньше временной сложности алгоритма Эдмондса, в  $N^3 \cdot n^{L-2}$  раз.* ◀

**ТЕОРЕМА 4.** *Временная сложность параллельного алгоритма  $a_1$ , где используется  $r$  процессоров  $p_1, p_2, \dots, p_r$ ,  $1 < r \leq k$ ,  $k = n^{L-1}$  равна  $O\left(\frac{1}{r} \cdot N \cdot n^3\right)$ .*

**ДОКАЗАТЕЛЬСТВО.** Алгоритм  $a_1$  выполняет шаг 1 параллельно, каждым из  $r$  процессоров. Для выполнения шага 1 требуется  $O(n^4)$  времени. Но все процессоры работают параллельно, т.е. все  $r$  процессоров закончат свою работу: «выполнение шага 1» – одновременно, за время  $O(n^4)$ . Шаг 1 будет выполнен ровно  $\frac{k}{r}$  раз. Тогда, получим:

$$O\left(\frac{k}{r} \cdot n^4\right) = O\left(\frac{1}{r} \cdot n^{L-1} \cdot n^4\right) = O\left(\frac{1}{r} \cdot n^L \cdot n^3\right) = O\left(\frac{1}{r} \cdot N \cdot n^3\right).$$

Получаем, что временная сложность алгоритма  $a_1$ , в случае использования  $r$  процессоров, равна  $O\left(\frac{1}{r} \cdot N \cdot n^3\right)$ . ◀

**ПРИМЕЧАНИЕ 3.** *Временная сложность параллельного алгоритма  $a_1$ , где используется  $r$  процессоров  $p_1, p_2, \dots, p_r$ ,  $1 < r \leq k$ ,  $k = n^{L-1}$  меньше временной сложности алгоритма Эдмондса, в  $rNn^{L-1}$  раз.* ◀

**ТЕОРЕМА 5.** *Алгоритм  $a_1$  выделяет покрытие  $x_1$  на предфрактальном  $(n, L)$  - графе  $G_L = (V_L, E_L)$ , порожденном полной*

$n$  – вершинной заправкой  $H=(W,Q)$ ,  $|W|=n$ ,  $|Q|=n$ , оптимальное по критериям  $F_4(x)$ ,  $F_6(x)$  и оцениваемое по третьему  $F_3(x_1) \leq \frac{n^l}{2} + n^{l-1}$ ,  $l = \overline{1, L}$  и пятому  $F_5(x_1) \leq \left( \left\lfloor \frac{n}{2} \right\rfloor + 1 \right) l$ ,  $l = \overline{1, L}$ . ( $[a]$  – целая часть числа  $a$ )

**ДОКАЗАТЕЛЬСТВО.** Доказательство оптимальности по критерию  $F_4(x)$  следует из структуры работы алгоритма  $a_1$ . Алгоритм  $a_1$  выделяет один  $2l$  – дольный граф, то есть  $F_4(x)$  состоит из одной компоненты.

Для того, чтобы доказать оптимальность по критерию  $F_6(x)$  покажем, что на полном графе  $H=(W,Q)$  максимальное число ребер выделенного двудольного графа  $D$  достигается, если доли равны (в случае, если  $n$  делится пополам) или же доли будут равны  $\frac{n}{2}$  и  $\frac{n}{2}-1$  (если  $n$  не делится пополам).

Для доказательства этого утверждения сформулируем и докажем предложение.

**Предложение.** *Двудольный граф  $D_1=(V_1, V_2, E)$  на графе  $H=(W, Q)$ ,  $|W|=n, |Q|=n$  будет наибольшим по количеству ребер, если  $|V_1|=|V_2|=\frac{n}{2}$  (в случае, когда  $n$  четно) и если  $|V_1|=\frac{n-1}{2}, |V_2|=\frac{n+1}{2}$  (в случае, когда  $n$  нечетно).*

**ДОКАЗАТЕЛЬСТВО.** Докажем первую часть предложения.

Если  $n$  – четно, обозначим  $|V_{1_k}|=k$ , тогда  $|V_{2_k}|=n-k$ ,  $k = \overline{1, n}$  и максимальное количество ребер  $|E_x^k|=k(n-k)$  достигается при  $k = \frac{n}{2}$ .

Если  $n$  – нечетно, то одну вершину выделяем в сторону. Тогда число вершин четно и доли должны быть равны  $\frac{n-1}{2}$ . Так как рассматриваем полный граф, то эту вершину относим в одну из долей и получаем максимальное число ребер. ◀

На основании доказанного предложения на каждой затравке  $z_s^l$  алгоритм  $a_1$  выделяет максимальное число ребер. На основании полученного результата и работы алгоритма  $a_1$  на всех затравках на шаге 1, выделяется наибольшее количество ребер.

Аналогично рассуждая, видим, что на каждом шаге  $l$  алгоритм  $a_1$  выделяет максимальное число ребер, что доказывает оптимальность критерия  $F_6(x_1)$ .

Дадим оценку  $F_3(x_1)$ . Согласно нашей модели, главными исполнителями является одна из долей выделенного двудольного графа. Максимальное число долей выделенного двудольного графа на затравке  $z_s^l$  будет равна  $\left\lceil \frac{n}{2} \right\rceil + 1$ , а количество всех затравок на последнем шаге равно  $n^{l-1}$ ,  $l = \overline{1, L}$ . Тогда общее количество исполнителей будет меньше либо равно  $\left( \left\lceil \frac{n}{2} \right\rceil + 1 \right) n^{l-1} \leq \left( \frac{n}{2} + 1 \right) n^{l-1} = \frac{n^l}{2} + n^{l-1}$ .

Для оценки критерия  $F_5(x)$  последовательно оценим результат критерия  $F_5(x)$  по шагам работы алгоритма.

На первом шаге работы алгоритма  $a_1$ , максимальная степень выделенного двудольного графа на затравке  $z_s^1$  будет равна  $\left\lceil \frac{n}{2} \right\rceil + 1$ . На каждом алгоритм не запрещает увеличение степени вершины на  $\left\lceil \frac{n}{2} \right\rceil + 1$ . Аналогично рассуждая, на  $l$ -м шаге максимальная степень вершины может достичь  $\left( \left\lceil \frac{n}{2} \right\rceil + 1 \right) l$ . ◀

### Литература

1. Залиханов М.Ч. Устойчивое развитие России: перспективы и угрозы // Безопасность Евразии. 2001. №2. С.518-525.
2. Кочкаров А.А., Малинецкий Г.Г. Управление безопасностью и стойкостью сложных систем в условиях внешних воздействий // Проблемы управления. 2005. №5. С.70-76.
3. Кочкаров Р.А. Целевые программы: инструментальная поддержка. М.: ЗАО «Издательство «Экономика»», 2007. 223с.

4. Пospelов Г.С. Ириков В.А. Программно-целевое планирование и управление – М.: Сов. Радио. 1976. 440с.
5. Кочкаров А.М. Распознавание фрактальных графов. Алгоритмический подход. Нижний Архыз: РАН САО, 1998
- 6 . Емеличев В.А., Перепелица В.А. Сложность дискретных многокритериальных задач // Дискретная математика. – 1994. Т. 6, вып. 1. С .3 – 33.
7. Кристофидес Н. Теория графов. Алгоритмический подход. – М.: Мир, 1978.

### References

1. Zalikhanov M.C. Sustainable development of Russia: prospects and threats // Security Evrazii.2001. № 2. S.518-525.
2. Kochkarov A.A., Malinetskii G.G. Managing security and resilience of complex systems with external influences // Control. 2005. № 5.S.70-76.
3. Kochkarov R.A. Targeted programs: instrumental support. Moscow: ZAO "Publisher" Economy ", 2007. 223s.
4. Pospelov G.S., Irikov V.A. Software-oriented planning and management - Sov. Radio. 1976. 440С.
5. Kochkarov A.M. Recognition of fractal graphs. Algorithmic approach. Nizh. Arkhyz: SAO RAS, 1998
6. Emelichev V.A., Perepelitsa V.A. Complexity of discrete multicriteria problems // Appl. Math - 1994. Т. 6, no. 1. pp .3 - 33.
7. N. Christofides, Graph Theory. Algorithmic approach. - Academic Press, 1978.