

О ДВУХ ЗАДАЧАХ АППРОКСИМАЦИИ ВЗВЕШЕННЫХ ГРАФОВ И АЛГОРИТМАХ ИХ РЕШЕНИЯ

А. Р. Ураков, Т. В. Тимеряев

Уфимский государственный авиационный технический университет, г. Уфа, Россия

E-mail: urakov@ufanet.ru, timeryaev@yandex.ru

Рассматриваются две связанные задачи аппроксимации взвешенных графов. В качестве погрешности аппроксимации рассматривается абсолютная величина максимума разностей расстояний между вершинами исходного графа и соответствующими им вершинами графа аппроксимирующего. В первой задаче требуется минимизировать погрешность аппроксимации при заданной размерности аппроксимирующего графа. Во второй — минимизировать размерность аппроксимирующего графа при заданной погрешности. Для задач приводится постановка в терминах задачи разбиения графа, описываются эвристические алгоритмы решения и производится сравнение с решением известным алгоритмом разбиения графа.

Ключевые слова: *аппроксимация графа, разбиение графа, уменьшение сложности задач.*

Введение

Под аппроксимацией графа в общем случае понимают процесс приближения некоторой исходной информации о исследуемом графе (возможно, не обладающей полнотой) графом с нужными исследователю характеристиками. В качестве исходной информации может выступать либо граф, обладающий неудобными с точки зрения рассматриваемой задачи свойствами, либо набор данных, не представленный в виде графа, но требующий такого представления для возможности формулирования и решения задач на нём с использованием аппарата теории графов.

Известные задачи аппроксимации графа можно условно разделить на следующие категории: 1) восстановление исследуемой структуры по ограниченному объёму информации в виде графа с последующим изучением её свойств и механизмов посредством решения задач на получившемся приближённом графе [1]; 2) приближение исходного графа с одним числом рёбер (его матрицы связности) графом на тех же вершинах с другим числом рёбер (другой матрицей связности), обладающим определёнными свойствами [2–4]; 3) различного рода задачи на графах специального вида [5, 6].

В одной из постановок задачи аппроксимации требуется приблизить рассматриваемый граф графом, состоящим из k компонент связности [7]. В таком виде задача аппроксимации имеет много общего с задачей разбиения. В задаче разбиения графа требуется найти представление множества вершин исходного графа V в виде некоторого числа попарно непересекающихся подмножеств. Как правило, задачи разбиения графа содержат размер получаемого «разреза» множества вершин (*cut size*) как одну из составляющих целевой функции. В соответствии с этим существуют различные постановки задачи (*min-cut*, *ratio cut*, *balanced cut* и др.) [8–10].

В данной работе рассматривается не упоминавшаяся ранее в литературе задача аппроксимации взвешенного графа большой размерности графом меньшей размерности с целевой функцией на максимум разностей расстояний между вершинами исходного

графа и соответствующими им вершинами полученного малого графа. Рассматриваются два варианта задачи, приводятся их формулировки в терминах задач аппроксимации и разбиения графа. Для задач предлагаются алгоритмы решения, приводится их сравнение на данных из открытого доступа.

1. Постановка задачи и определения

Рассматривается простой связный ненаправленный взвешенный граф $G = (V, E, w)$, где $V = \{v_1, v_2, \dots, v_n\}$ — множество вершин; $E = \{e_1, e_2, \dots, e_m\}$ — множество рёбер; $w : E \rightarrow [0, \infty)$ — неотрицательная вещественная весовая функция на рёбрах. Введём ряд обозначений и определений.

Вес ребра из вершины v_i в вершину v_j будем обозначать $w(i, j)$ (отсутствие ребра — $w(i, j) = \infty$). *Длина пути* — сумма весов входящих в него рёбер. *Расстояние* m_{ij} из вершины v_i в вершину v_j — длина кратчайшего пути из v_i в v_j .

Рассматриваемые задачи аппроксимации состоят в сопоставлении исходному графу G аппроксимирующего графа $G_a = (V_a, E_a, w_a)$ меньшей размерности: $|V_a| = n_a$, где $1 \leq n_a \leq n$. Будем называть вершины $\{v_i^a : i = 1, \dots, n_a\}$ аппроксимирующего графа *метавершинами*. Множество метавершин может быть подмножеством исходного множества вершин ($V_a \subset V$) или как частично, так и полностью не содержаться в V . Вес ребра между метавершинами v_i^a и v_j^a графа G_a будем обозначать $w_a(i, j)$, расстояние между этими же метавершинами — m_{ij}^a . Будем обозначать $v_{c(i)}^a$ метавершину графа G_a , аппроксимирующую вершину v_i графа G . *Погрешностью аппроксимации* будем называть величину $A_e = \max_{i,j=1,\dots,n} (|m_{ij} - m_{c(i)c(j)}^a|)$.

В работе рассматриваются две задачи аппроксимации: 1) уменьшение погрешности A_e при заданной размерности G_a и 2) уменьшение размерности G_a так, чтобы погрешность A_e не превосходила заданной величины.

Задача 1. Даны простой связный ненаправленный взвешенный граф $G = (V, E, w)$ с неотрицательной вещественной весовой функцией на рёбрах $w : E \rightarrow [0, \infty)$ и вещественное число $e_{\max} > 0$. Найти граф $G_a = (V_a, E_a, w_a)$ и функцию $c : \{1, \dots, n\} \rightarrow \{1, \dots, n_a\}$, такие, что подграфы $G_j(V_j)$, где $V_j = \{v_i \in V : c(i) = j\}$, $j = 1, \dots, n_a$, связны, $A_e \leq e_{\max}$ и $|V_a| \rightarrow \min$.

Задача 2. Даны простой связный ненаправленный взвешенный граф $G = (V, E, w)$, $|V| = n$, с неотрицательной вещественной весовой функцией на рёбрах $w : E \rightarrow [0, \infty)$ и целое число k , $n > k > 0$. Найти граф $G_a = (V_a, E_a, w_a)$, $|V_a| = k$, и функцию соответствия $c : \{1, \dots, n\} \rightarrow \{1, \dots, k\}$, такие, что подграфы $G_j(V_j)$, где $V_j = \{v_i \in V : c(i) = j\}$, связны и $A_e \rightarrow \min$.

Обе рассматриваемые задачи являются NP-трудными. Эти задачи при некоторых допущениях могут быть сформулированы в терминах задачи разбиения графа. Введём обозначения: *диаметр множества вершин* $d(V_i) = \max_{j,k:v_j, v_k \in V_i} m_{jk}$; *радиус множества*

вершин $r(V_i) = \min_{j:v_j \in V_i} \left(\max_{k:v_k \in V_i} m_{jk} \right)$; вершину, на которой достигается $r(V_i)$, будем называть *центром множества вершин* V_i и обозначать $c(V_i)$.

Будем аппроксимировать вершины подмножеств разбиения V_i центрами этих подмножеств $v_{c_i} = c(V_i)$. Веса рёбер в графе G_a между этими центрами будем полагать равными расстояниям между центрами в исходном графе G : $w_a(c_i, c_j) = m_{c_i c_j}$. Добавим для каждой вершины G_a петлю с весом, равным половине диаметра соответствующего множества разбиения: $w_a(c_i, c_i) = d(V_i)/2$. Дополнительно будем требовать связность всех подграфов разбиения $G_i(V_i)$.

При выполнении всех этих условий для погрешности аппроксимации справедливо $d_{\max}/2 \leq A_e \leq r_{m_1} + r_{m_2}$. Здесь $d_{\max} = \max_i d(V_i)$, r_{m_1} — максимальный, а r_{m_2} — второй по величине радиусы множеств разбиения V_i . Действительно, в этом случае погрешность аппроксимации внутри подграфов не превосходит $d_{\max}/2 \leq r_{m_1}$, а погрешность аппроксимации между вершинами v_i и v_j из разных подграфов не превосходит суммы расстояний от этих вершин до центров соответствующих подграфов, т. е. $A_e \leq r_{m_1} + r_{m_2}$.

При такой оценке для A_e задачи разбиения графа, соответствующие поставленным задачам аппроксимации, выглядят следующим образом.

Задача 1*. Даны простой связный ненаправленный взвешенный граф $G = (V, E, w)$ с неотрицательной вещественной весовой функцией на рёбрах $w : E \rightarrow [0, \infty)$ и вещественное число $e_{\max} > 0$. Найти разбиение множества вершин графа на попарно непересекающиеся подмножества $V_1, V_2, \dots, V_k$, такое, что все подграфы $G_i(V_i)$, порождённые этими множествами, связны, $r_{m_1} + r_{m_2} \leq e_{\max}$ и $k \rightarrow \min$.

Задача 2*. Даны простой связный ненаправленный взвешенный граф $G = (V, E, w)$, $|V| = n$, с неотрицательной вещественной весовой функцией на ребрах $w : E \rightarrow [0, \infty)$ и целое число k , $n > k > 0$. Найти разбиение множества вершин графа на k попарно непересекающихся подмножеств $V_1, V_2, \dots, V_k$, такое, что все подграфы $G_i(V_i)$, порождённые этими множествами, связны и $r_{m_1} + r_{m_2} \rightarrow \min$.

Предлагаются эвристические алгоритмы решения задач 1 и 2.

2. Алгоритм для задачи 1 (A1)

Алгоритм состоит в последовательном «удалении» вершин графа. Удаляемым вершинам ставится в соответствие в качестве метавершины одна из смежных вершин и выполняется пересчёт весов рёбер получившегося графа меньшей размерности.

Пусть на возможность удаления первой рассматривается вершина v_i степени m , смежная с вершинами v_{i_z} , $z = 1, \dots, m$. Удаление v_i влечёт необходимость изменения весов рёбер только между смежными с ней вершинами. Пересчёт весов данных рёбер будем производить по формуле

$$w(i_j, i_l) = \begin{cases} w(i_j, i) + w(i, i_l), & \text{если } w(i_j, i) + w(i, i_l) < w(i_j, i_l), \\ w(i_j, i_l) & \text{иначе.} \end{cases}$$

После такого пересчёта погрешность аппроксимации между всеми неудалёнными вершинами останется равной нулю. Поэтому для минимизации общей погрешности аппроксимации необходимо поставить в соответствие удалённой вершине v_i такую метавершину $v_{c(i)}$, которая сделает как можно меньшим максимум абсолютной разности расстояний между v_i , $v_{c(i)}$ и вершинами исходного графа: $\max_j (|m_{ij} - m_{c(i)j}|) \rightarrow \min$. В исходном графе расстояния от v_i до любой вершины графа v_j определяются формулой $m_{ij} = \min_z (w(i, i_z) + m_{i_z j})$; после удаления v_i расстояния между неудалёнными вершинами $m_{i_z j}$ не изменились. В качестве $v_{c(i)}$ будем выбирать ближайшую смежную с v_i вершину $v_{c(i)} = v_{i_j} : w(i, i_j) = \min_z w(i, i_z)$, при этом погрешность аппроксимации увеличится не более чем на вес ребра $w(i, i_j)$.

В задаче 1 требуется, чтобы погрешность аппроксимации не превосходила некоторого заданного числа $A_e \leq e_{\max}$. Для достижения этого необходимо хранить информацию о погрешности, вносимой каждой из оставшихся вершин v_l . Пусть A_e^l — максимально возможная погрешность аппроксимации между вершинами, аппроксимируемыми вершиной v_l ; сначала все $A_e^l = 0$. Удаление вершины v_i изменяет погрешность

аппроксимирующей её вершины $v_{c(i)}$ по правилу $A_e^{c(i)} = A_e^{c(i)} + A_e^i + w(i, c(i))$. В силу сохранения расстояний между оставшимися вершинами для выполнения $A_e \leq e_{\max}$ необходимо, чтобы для всех пар оставшихся вершин (v_i, v_j) выполнялось $A_e^i + A_e^j \leq e_{\max}$, это справедливо, как только $A_e^j \leq e_{\max}/2$ для всех j . Согласно этому, вершину v_i можно удалять, когда для аппроксимирующей её вершины $v_{c(i)}$ выполняется $A_e^{c(i)} + A_e^i + w(i, c(i)) \leq e_{\max}/2$.

Рассмотрение и удаление вершин может производиться различными способами. Одной из очевидных стратегий является удаление вершин в порядке роста добавляемой этими вершинами погрешности. В п. 4 приводятся результаты тестирования алгоритма для реализации, в которой вершины рассматриваются в порядке роста их степеней. Такая стратегия показала наилучшие результаты на тестовых графах.

Полученная таким образом аппроксимация не гарантирует связности подграфов $G_j(V_j)$, $V_j = \{v_i \in V : c(i) = j\}$, исходного графа, вершины которых аппроксимируются одной метавершиной. Существует правило переопределения функции c , гарантирующее связность таких подграфов и не увеличивающее погрешность аппроксимации.

Пусть для всех множеств V_j найдены радиусы $r(V_j)$ и центры $s(V_j)$ (быстрый поиск этих величин можно осуществить, используя алгоритм [11]). Если для вершины $v_i \in V_k$ отсутствует путь до центра $s(V_k)$ в графе $G_k(V_k)$, то в пути от v_i до $s(V_k)$ в исходном графе G существует вершина $v_j \in V_l$. Если принадлежность вершины v_i множеству V_l не увеличивает $r(V_l)$ и у вершины v_i принадлежность множеству не изменялась, полагаем $c(i) = l$. В противном случае все вершины v_s пути от v_i до v_j в исходном графе G делаем принадлежащими множеству V_k , полагая $c(s) = k$. После каждого изменения принадлежности множеству какой-либо из вершин центры и радиусы изменившихся множеств V_k, V_l пересчитываются и процесс повторяется снова до тех пор, пока не останется несвязных подграфов. После этого полагаем $w(c(V_j), c(V_j)) = r(V_j)$, $c(i) = c(V_j)$ для всех i , таких, что $c(i) = j$.

В худшем случае, когда для полного графа выполняется «удаление» $n - 2$ вершин, сложность алгоритма составляет $O(n^3)$.

3. Алгоритм для задачи 2 (A2)

Для решения задачи 2 предлагается эвристический итерационный алгоритм, в основе которого лежит алгоритм A1. Итеративно ищется такое значение величины $e_{\max}$ (и аппроксимация при этом значении), при котором решение задачи 1 алгоритмом A1 обладает свойством $|V_a| \leq k$, где k — ограничение из задачи 2. Для поиска такого $e_{\max}$ используется интерполяция многочленом Лагранжа и предположение о зависимости $e_{\max} = f(|V_a|)$. При этом считаем, что $f(|V|) = 0$, а для получения ещё одной точки интерполяции полагаем $e_{\max}^1 = \bar{w}(n/k)$, где $\bar{w}$ — средний вес ребра в графе.

Если в результате такого поиска аппроксимирующий граф содержит вершин меньше нужного ($|V_a| < k$), то полагаем $c(i) = c(V_j)$ для всех i , $c(i) = j$, а $k - |V_a|$ вершин v_i с максимальным удалением от центров своих множеств $s(V_j)$ переносим в отдельные множества, полагая $c(i) = i$.

4. Результаты

В качестве тестовых данных использовались взвешенные графы дорожных сетей США из открытого доступа [12]. Из них были получены связные подграфы размерностью от 10^3 до $5 \cdot 10^3$, по 10 для каждой размерности. Характеристики тестовых графов представлены в табл. 1.

Т а б л и ц а 1

Характеристики тестовых графов

Группа графов	Среднее кол-во вершин	Средняя степень вершин	Средний диаметр	Среднее расстояние	Кол-во графов
$G1$	10^3	2,48	$3,64 \cdot 10^5$	$1,39 \cdot 10^5$	10
$G2$	$2 \cdot 10^3$	2,6	$3,17 \cdot 10^5$	$1,07 \cdot 10^5$	
$G3$	$3 \cdot 10^3$	2,62	$5,63 \cdot 10^5$	$1,9 \cdot 10^5$	
$G4$	$4 \cdot 10^3$	2,68	$4,09 \cdot 10^5$	$1,45 \cdot 10^5$	
$G5$	$5 \cdot 10^3$	2,66	$4,88 \cdot 10^5$	$1,69 \cdot 10^5$	

Для сравнения с алгоритмом A2 использовалась модификация эвристического алгоритма Fiduccia — Mattheyses биразбиения графа (A2* [13], сложность одной итерации $O(|E|)$), решающая задачу в постановке 2*. Описание сущности процесса разбиения данным методом приводится в [14]. Здесь укажем лишь особенности адаптации алгоритма к рассматриваемой задаче.

В задаче требуется минимизировать сумму максимального r_{m_1} и второго по величине r_{m_2} радиусов множества вершин получившихся подграфов разбиения — $r_{m_1} + r_{m_2} \rightarrow \min$. После начального разбиения выбор пары подграфов (G_i, G_j) для его улучшения производится так, чтобы радиус множества вершин G_i был равен r_{m_1} или r_{m_2} . При этом G_j выбирается как подграф, имеющий минимальный радиус множества вершин среди подграфов, связанных с G_i хотя бы одним ребром и изменённых со времени последнего совместного рассмотрения с G_i . Число вершин — претендентов на перемещение — ограничивается требованием сохранения связности подграфов после перемещения. Улучшение пары (G_i, G_j) останавливается, если $\max_i g_i < 0$ (g_i — разница целевых функций до и после возможного перемещения v_i) или радиусы множеств вершин обоих подграфов меньше r_{m_2} .

После того как дальнейшее улучшение разбиения становится невозможным, для каждого подграфа G_i находится центр v_{c_i} и диаметр $d(G_i)$. Функция соответствия s задается как $s(j) = i$ для всех j , таких, что $v_j \in V_i$. Подграф G_a строится из k вершин, веса его рёбер задаются формулами $w_a(i, j) = m(c_i, c_j)$ и $w_a(i, i) = d(G_i)/2$.

Результаты экспериментов приведены в табл. 2 и 3.

Заключение

Рассмотрены две задачи аппроксимации на уменьшение размерности графа при ограничениях на величину изменения расстояний между исходными вершинами и вершинами аппроксимирующего графа. Полезность постановки этих задач обнаруживается в практических приложениях: транспортной логистике, мониторинге вычислительных сетей и т. д. Полученный малый аппроксимирующий граф может использоваться при решении NP-трудных задач на графах, позволяя: а) решать задачи быстрее, используя методы для графов большой размерности, б) получать более точные результаты, применяя более вычислительно сложные алгоритмы.

Рассматриваемые задачи аппроксимации являются частным случаем задачи разбиения графа. Функцию s , ставящую в соответствие вершинам исходного графа вершины аппроксимирующего графа, можно рассматривать как способ распределения вершин по множествам разбиения V_i . Такое отношение между задачами аппроксимации и разбиения графа позволяет использовать методы решения одной задачи при решении другой, и наоборот.

Т а б л и ц а 2

Результаты решения задачи 1 алгоритмом A1

Группа графов	$\bar{V}_n$	A_e фактическая	Время, с
$e_{\max} = 10^3$			
$G1$	897	975	1,4
$G2$	$1,8 \cdot 10^3$	987	5,8
$G3$	$2,7 \cdot 10^3$	991	14
$G4$	$3,5 \cdot 10^3$	995	25
$G5$	$4,4 \cdot 10^3$	996	41
$e_{\max} = 2 \cdot 10^3$			
$G1$	765	$2 \cdot 10^3$	1,4
$G2$	$1,4 \cdot 10^3$	$2 \cdot 10^3$	5,8
$G3$	$2,2 \cdot 10^3$	$2 \cdot 10^3$	14
$G4$	$2,8 \cdot 10^3$	$2 \cdot 10^3$	25
$G5$	$3,4 \cdot 10^3$	$2 \cdot 10^3$	41
$e_{\max} = 5 \cdot 10^3$			
$G1$	558	$4,9 \cdot 10^3$	1,3
$G2$	945	$4,9 \cdot 10^3$	5,4
$G3$	$1,5 \cdot 10^3$	$4,9 \cdot 10^3$	13
$G4$	$1,8 \cdot 10^3$	$5 \cdot 10^3$	24
$G5$	$2,2 \cdot 10^3$	$5 \cdot 10^3$	38

Т а б л и ц а 3

Результаты решения задачи 2 (Δ_1 — отношение A_e к диаметру графа, Δ_2 — отношение A_e к среднему расстоянию в графе)

Группа графов	$A2$				$A2^*$			
	A_e	Δ_1	Δ_2	Время, с	A_e	Δ_1	Δ_2	Время, с
$k = \lfloor 0,1 \cdot V \rfloor$								
$G1$	$4,6 \cdot 10^4$	0,14	0,35	0,52	$1,3 \cdot 10^5$	0,35	0,92	1,7
$G2$	$3 \cdot 10^4$	0,1	0,29	2,3	$9,6 \cdot 10^4$	0,31	0,91	8,4
$G3$	$4,3 \cdot 10^4$	0,09	0,26	5,4	$1,8 \cdot 10^5$	0,32	0,95	23
$G4$	$3,2 \cdot 10^4$	0,08	0,24	9,9	$1,4 \cdot 10^5$	0,33	0,97	44
$G5$	$2,8 \cdot 10^4$	0,06	0,19	16	$1,7 \cdot 10^5$	0,32	0,95	81
$k = \lfloor 0,2 \cdot V \rfloor$								
$G1$	$2,4 \cdot 10^4$	0,08	0,2	0,83	$1,2 \cdot 10^5$	0,31	0,82	1,9
$G2$	$1,7 \cdot 10^4$	0,06	0,16	3,6	$8,1 \cdot 10^4$	0,26	0,76	11
$G3$	$2,3 \cdot 10^4$	0,05	0,13	8,5	$1,6 \cdot 10^5$	0,28	0,83	30
$G4$	$1,6 \cdot 10^4$	0,04	0,12	15	$1,2 \cdot 10^5$	0,29	0,85	62
$G5$	$1,6 \cdot 10^4$	0,04	0,11	26	$1,5 \cdot 10^5$	0,28	0,82	112
$k = \lfloor 0,5 \cdot V \rfloor$								
$G1$	$7 \cdot 10^3$	0,02	0,06	1,3	$8 \cdot 10^4$	0,23	0,58	2,7
$G2$	$4,9 \cdot 10^3$	0,02	0,05	5,5	$5,8 \cdot 10^4$	0,18	0,53	17
$G3$	$5,9 \cdot 10^3$	0,01	0,04	13	$1 \cdot 10^5$	0,19	0,55	50
$G4$	$4,3 \cdot 10^3$	0,01	0,03	24	$8,7 \cdot 10^4$	0,19	0,55	108
$G5$	$4,3 \cdot 10^3$	0,01	0,03	39	$8,4 \cdot 10^4$	0,18	0,53	199

Необходимо отметить, что в практической деятельности связность подграфов $G_j(V_j)$, приводимая в постановках рассматриваемых задач, часто не является необходимой. Это связано с тем, что граф разбивается лишь логически, связность его не нарушается. С исключением этого условия в приведённых алгоритмах отпадает необходимость в процедуре изменения принадлежности вершин множествам V_j и вычислении $r(V_j)$ и $s(V_j)$, что значительно уменьшает время аппроксимации данными методами.

ЛИТЕРАТУРА

1. *Bonneau J., Anderson J., et al.* Eight friends are enough: social graph approximation via public listings // Proc. Second ACM EuroSys Workshop on Social Network Systems. Nuremberg, 2009. P. 13–18.
2. *Koutis I., Levin A., et al.* Improved spectral sparsification and numerical algorithms for SDD matrices // Proc. STACS. Paris, 2012. P. 266–277.
3. *Long B., Xiaoyun X., et al.* Community learning by graph approximation // Proc. Seventh IEEE Intern. Conf. on Data Mining. Omaha, 2007. P. 232–241.
4. *Jacob J., Jentsch M., et al.* Detecting hierarchical structure in molecular characteristics of disease using transitive approximations of directed graphs // Bioinformatics. 2008. V. 24(7). P. 995–1001.
5. *Kuhn F., Moscibroda T., et al.* Unit disk graph approximation // Proc. Joint Workshop on Foundations of Mobile Computing. Philadelphia, 2004. P. 17–23.
6. *Fertin G., Hermelin D., et al.* Common structured patterns in linear graphs: approximation and combinatorics // Proc. 18th Annual Conf. on Combinatorial Pattern Matching. Berlin, 2007. P. 241–252.
7. *Ильев В. П., Ильева С. Д. и др.* Приближенные алгоритмы для задач аппроксимации графов // Дискретный анализ и исследование операций. 2007. Т. 18(1). С. 41–60.
8. *Kernighan B. and Lin S.* An efficient heuristic procedure for partitioning graphs // The Bell System Techn. J. 1970. V. 40(1). P. 291–307.
9. *Hagen L. and Kahng A.* New spectral methods for ratio cut partitioning and clustering // IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems. 1992. V. 11(9). P. 1074–1085.
10. *Andreev K. and Racke H.* Balanced graph partitioning // Proc. sixteenth Annual ACM Symposium on Parallelism in Algorithms and Architectures. Barcelona, 2004. P. 120–124.
11. *Ураков А. Р., Тимеряев Т. В.* Алгоритмы быстрого поиска для двух задач о метрических характеристиках взвешенных графов // Управление большими системами. 2013. Вып. 42. С. 153–172.
12. <http://www.dis.uniroma1.it/challenge9/download.shtml> — 9th DIMACS Implementation Challenge — Shortest Paths (дата обращения: май 2013).
13. *Fiduccia C. and Mattheyses R.* A linear time heuristic for improving network partitions // DAC'82. Proc. 19th Design Automation Conf. Las Vegas, 1982. P. 175–181.
14. *Ураков А. Р., Тимеряев Т. В.* Многоуровневый алгоритм разбиения графов по критерию средней длины // Информационные технологии. 2012. № 4. С. 22–25.