УДК 681.3

Анализ результатов применения алгоритма муравьиной колонии в задаче поиска пути в графе при наличии ограничений

Ватутин Э.И., Титов В.С.

Юго-Западный государственный университет, кафедра вычислительной техники, г. Курск, 305016, Россия

Аннотация. В работе приведено описание алгоритма муравьиной колонии применительно к решению комбинаторных оптимизационных задач на примере задачи поиска кратчайшего пути в графе. Предложен ряд модификаций по сравнению с классической математической моделью М. Дориго, позволяющих сократить вычислительные затраты при поиске решения и повысить его качество. Приведены результаты сравнения качества решений с известными методами для графов с различным числом вершин и различной плотностью.

Ключевые слова: дискретная комбинаторная оптимизация, алгоритм муравьиной колонии, теория графов, поиск кратчайшего пути в графе, эвристические методы.

Abstract. The article describes an ant colony optimization algorithm applied to the solution of combinatorial optimization problems by the example of the problem of finding the shortest path in the graph. A number of modifications compared to the classical mathematical model M. Dorigo, which reduce the computational cost in finding solutions and improve its quality, are described. The results of comparing the quality of solutions with known methods for graphs with different numbers of nodes and different densities are given.

Keywords: discrete combinatorial optimization, ant colony optimization, graph theory, shortest path problem, heuristic methods.

Существует довольно обширный класс задач дискретной комбинаторной оптимизации, для которых не существует быстрых полиномиальных алгоритмов, способных производить нахождение оптимальных решений. Это определяется либо принадлежностью задачи к классу *NP*, в таком случае разработка такого алгоритма для детерминированной машины Тьюринга теоретически невозможна (разумеется, при условии неравенства классов *P* и *NP* [1]), либо подобный алгоритм неизвестен (ситуация, характерная, например, для задачи изоморфизма графов [2]), хотя теоретически не доказана невозможность его построения. Сложившееся противоречие вынуждает использовать эвристические методы для решения ряда важных задач, которые не гарантируют отыскание оптимальных решений, однако обеспечивают получение решений неплохого качества, подходящих для использования на практике. Эвристические методы подразделяются на два больших класса [3]: последовательные и итерационные. Последовательные методы и базирующиеся на них алгоритмы не подразумевают перебора различных вариантов и обеспечивают получение одного субоптимального решения. Как правило они характеризуются относительно небольшой асимптотической временной сложностью, редко превышающей $O\left(n^3\right)$, и, как следствие, требуют относительно малых затрат вычислительного времени. В отличие от них итерационные методы базируются на ограниченном переборе множества вариантов решения с выбором наилучшего, а соответствующие алгоритмы характеризуются как более высокой асимптотической временной сложностью, так и б**о**льшими затратами вычислительного времени (как минимум на 1–2 порядка) при практической реализации. В некоторых случаях затраты времени на отыскание решения можно снизить путем перехода к использованию параллельной программной реализации, однако это удается не всегда и определяется спецификой задачи и используемого для ее решения алгоритма. Для многих задач разработаны специализированные методы [4], которые, в отличие от универсальных, имеют узкую сферу применения, что компенсируется их скоростными характеристиками и/или высоким качеством получаемых решений. При этом качество решений, полученных с использованием различных подходов для различных задач может существенно отличаться, что позволяет делать выводы о целесообразности применимости методов различных направлений для решения различных задач.

Одним из перспективных универсальных подходов к решению задач дискретной комбинаторной оптимизации является алгоритм муравьиной колонии [5], предложенный М. Дориго в 1992 г. и базирующийся на результатах изучения поведения колонии муравьев. При выборе направления движения муравей выбирает конкретное решение, руководствуясь двумя факторами: длиной пути l_i, $i=\overline{1,n}$, где n – число направлений движения из текущего положения, и количеством феромона τ_i, оставленном на *i*-м пути другими муравьями. При достижении цели муравей помечает понравившийся ему путь новой порцией феромона, причем на более коротком пути остается большее количество феромона, что при итеративном повторении (движении группы муравьев по пути в различное время) обеспечивает отметку феромоном более перспективных путей, оставляя при этом небольшую возможность для отклонения от пути.

Направление движения электронного муравья в каждом конкретном случае выбирается пропорционально вероятности

$$P_i=\frac{\eta_i^{\alpha}\tau_i^{1-\alpha}}{\sum\limits_{j=1}^{n}\eta_j^{\alpha}\tau_j^{1-\alpha}},\ i=\overline{1,n}, \qquad (1)$$

где $\eta_i=\dfrac{1}{l_i}$ в рассматриваемой задаче, $\alpha\in[0;1]$ – настроечный параметр алгоритма, выбираемый в каждом конкретном случае индивидуально и характеризующий его «жадность» (в предельных случаях при $\alpha=1$ алгоритм превращается в жадный, т.к. электронный муравей руководствуется только длиной пути, а при $\alpha = 0$ – только количеством феромона). При достижении цели, критерием чего в природе является нахождение маршрута к источнику пищи, а в рассматриваемой цифровой модели – нахождение решения, электронный муравей помечает пройденный путь, добавляя к имеющемуся количеству феромона на каждом из участков пути прибавку $\Delta\tau = \frac{Q}{L}$, где Q – некая константа (настроечный параметр алгоритма), $L = \sum_i l_i$ – длина найденного пути (качество решения). При этом обычно учитывается испарение феромона

$$\tau_i^{(t)} = \beta\tau_i^{(t-1)} + \Delta\tau, \qquad (2)$$

где t – дискретное время (номер итерации), $\beta \in [0;1]$ – настроечный параметр, влияющий на скорость испарения: при $\beta = 0$ происходит полное испарение феромона прошлых итераций, при $\beta = 1$ – сохранение значения феромона с «момента сотворения мира» (иногда вместо β в литературе используется параметр $p = 1 - \beta$, обладающий обратным смыслом и не меняющий сути алгоритма). При реализации алгоритма на практике обычно вводится понятие колонии из M муравьев, поведение каждого из которых в простейшем случае определяется рассмотренными выше правилами.

Существование в природе на протяжении миллионов лет муравьиных колоний, достигающих с использованием группового интеллекта явных успехов в приспособляемости к условиям обитания, является неоспоримым эмпирическим доказательством правильности используемого муравьями подхода, что стимулирует попытки адаптации рассмотренного подхода к решению различных практически важных задач. Рассмотрим возможность применения указанного алгоритма для решения задачи поиска кратчайшего пути в графе $G = \langle A, V \rangle$, где $A = \{a_1, a_2, ..., a_N\}$ – множество вершин, $N = |A|$ – число вершин, $V = \{v_1, v_2, ..., v_M\} \subseteq A \times A$ – множество дуг, $M = |V|$ – число дуг.

Данная задача является широко известной и в графах без отрицательных весов точно решается с использованием алгоритма Дейкстры [6] за время $O\left(N^2\right)$, что позволяет использовать ее как достаточно простую модель для тестирования различных эвристических подходов. Исходные данные задаются в виде матрицы смежности графа $M = \left\| m_{ij} \right\|$, $i, j = \overline{1, N}$, элементы $m_{ij} = l\left(a_i, a_j\right)$ которой представляют собой вещественное число, определяющее длину пути между парой вершин a_i и a_j. В случае отсутствия пути между парой вершин соответствующее значение матрицы смежности условно принимается равным бесконечности (на практике для идентификации подобной ситуации можно использовать нулевое, отрицательное или заведомо большое значение). Поиск решения сводится к отысканию такого пути $P = \left[a_{i_1}, a_{i_2}, ..., a_{i_m}\right]$ между начальной $a_{i_1} = a_{нач}$ и конечной $a_{i_m} = a_{кон}$, что его длина $L = \sum_{j=1}^{m-1} m_{i_j, i_{j+1}} \to \min$. Все вершины в составе пути должны быть различны $\left(\overline{\exists} j, k, j \neq k \colon a_{i_j} = a_{i_k}\right)$, т.к. тривиально доказывается, что повторное посещение уже посещенных ранее вершин не приводит к нахождению кратчайшего пути. Решение считается некорректным, если в составе пути присутствует пара вершин, переход между которыми запрещен по условию задачи. Пример графа, соответствующего ему комбинаторного дерева и нескольких путей приведен на рис. 1.

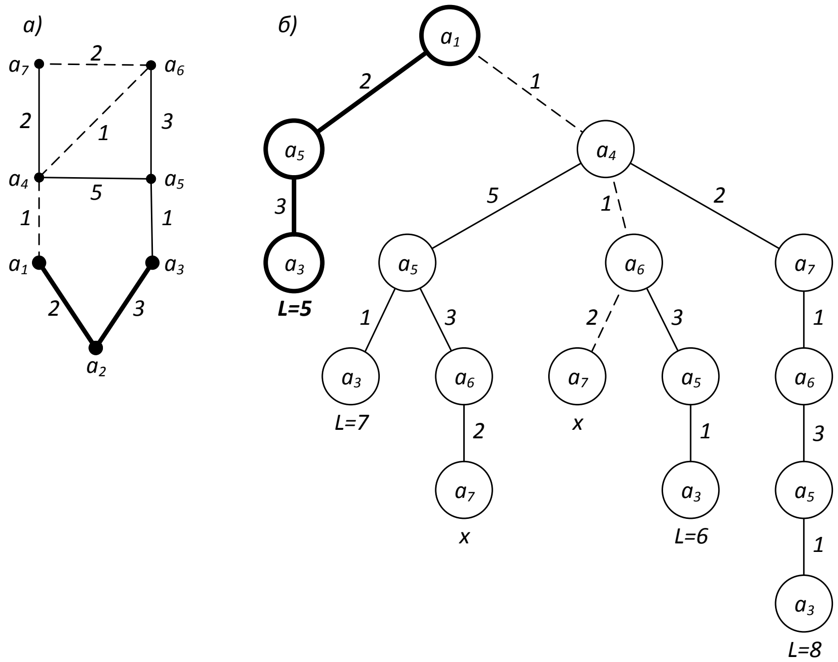

Рис. 1. Пример графа (а) и соответствующее ему дерево возможных путей между вершинами a_1 и a_3 (б). Крестами отмечены пути, приводящие в тупик, жирным выделено оптимальное решение, штриховкой показано решение, получаемое с использованием жадной стратегии

Известные эвристические подходы, такие как жадный алгоритм или случайный перебор [10], отличаются друг от друга способом выбора направления перехода из текущей вершины: в первом случае выбирается кратчайший путь, а во втором – равновероятное случайное направление движения в независимости от длины предстоящего участка пути. Подобные методы не подразумевают возвратов (в отличие от метода полного перебора, реализующего обход комбинаторного дерева в глубину), что с одной стороны обеспечивает быстрое нахождение решений, однако с другой может приводить в тупик, когда из текущей вершины, являющейся последней в формируемом пути, невозможно перейти ни в одну из еще не рассмотренных вершин, причем заданная конечная вершина не достигнута. Вероятность попадания в тупики увеличивается по мере увеличения числа запретов на движение по дугам графа, при этом рассматриваемая задача из задачи на доказательство оптимальности решения трансформируется в задачу на существование решения, являющуюся гораздо более сложной.

При использовании алгоритма муравьиной колонии выбор направления движения из текущей вершины определяется пропорционально вероятности (1). С учетом того, что нормирующий знаменатель $\sum_{j=1}^{n}\eta_j^{\alpha}\tau_j^{1-\alpha}$ в формуле (1) является константой, его можно исключить из рассмотрения, осуществляя выбор направления движения с использованием эвристики

$$\eta_i^{\alpha}\tau_i^{1-\alpha}r_k \to \max\,, \tag{3}$$

где r_k – очередное k-е псевдослучайное число, формируемое генератором псевдослучайных чисел. Максимальное значение эвристики (3) будет соответствовать предпочтительному направлению движения. При этом в ходе работы алгоритма достаточно $M=1$ муравья, который проделывает путь по одной из ветвей комбинаторного дерева от корня к одному из листьев заданное число итераций C, помечая его феромоном в соответствии с (2) случае попадания в результирующую вершину. В случае попадания муравья в тупик пометка пути феромоном не производится.

Введем понятие плотности графа, определяемой как отношение числа дуг к максимально возможному числу дуг для графа из N вершин:

$$d(G)=\frac{M}{N(N-1)}\,. \tag{4}$$

Введенное понятие плотности не следует путать с одноименным понятием, встречающимся в некоторых источниках и выражаемым через операцию стягивания и число Хардвигера [7]. Несложно заметить, что $0\le d(G)\le 1$, причем случай $d(G)=0$ отвечает графу, состоящему из изолированных вершин при полном отсутствии в графе дуг, а $d(G)=1$ – полносвязному графу без петель. При больших значениях плотности $d(G)$ б**о**льшая часть ветвей комбинаторного дерева будут являться корректными решениями, при малых значениях плотности возрастает вероятность попадания в тупик, что на практике в некоторых случаях выливается в невозможность отыскания решения с использованием эвристических методов.

С целью подробного исследования потенциала алгоритма муравьиной колонии с учетом наличия ограничений на движение по дугам была разработана программная реализация [11], с использованием которой организована серия вычислительных экспериментов. В ходе каждого сравнения методов решения поставленной задачи была произведена генерация выборки $\Lambda = \{G_1, G_2, ..., G_K\}$ из K псевдослучайных матриц смежности для взвешенных неориентированных графов с заданным числом вершин N и заданной плотностью $d(G)$, длины дуг которых выбирались равновероятно из диапазона $[0;1]$. Для каждого из графов осуществлялся поиск кратчайшего пути с использованием заданного алгоритма, при этом производилась оценка следующих средневыборочных параметров:

- средняя длина пути $\overline{L} = \frac{\sum_{i=1}^{K} L(G_i)\phi(G_i)}{K}$, где $\phi(G)=1$ в случае отыскания решения и $\phi(G)=0$ в случае попадания в тупик;
- среднее отклонение длины пути от минимальной $L_{\min}(G_i)$, полученной с использованием алгоритма Дейкстры $\Delta\overline{L} = \frac{\sum_{i=1}^{K} \left(L(G_i) - L_{\min}(G_i)\right)\phi(G_i)}{K}$;
- вероятность нахождения решения $\overline{p} = \frac{\sum_{i=1}^{K} \phi(G_i)}{K}$;
- вероятность нахождения оптимального решения $\overline{p}_{opt} = \frac{\sum_{i=1}^{K} \theta(G_i)}{K}$, где $\theta(G)=1$ в случае отыскания оптимального решения $\left(L(G_i) = L_{\min}(G_i)\right)$ и $\theta(G)=0$ в противном случае;

- среднее число итераций $\bar{C} = \frac{\sum_{i=1}^{K} C(G_i)}{K}$, необходимое итерационному методу для получения условно оптимального решения, где $C(G_i)$ – номер итерации, на которой получено наилучшее решение (для последовательных подходов $\bar{C} \equiv 1$).

Для алгоритма Дейкстры применительно к связным графам по определению $\Delta\bar{L} = 0$, $\bar{p} = 1$, $\bar{p}_{opt} = 1$. Лучшим будем считать алгоритм, обеспечивающий нахождение решений, для которых $\bar{L} \to \min$, $\Delta\bar{L} \to \min$, $\bar{p} \to \max$, $\bar{p}_{opt} \to \max$, $\bar{C} \to \min$.

При реализации алгоритма муравьиной колонии в условиях наличия тупиков возможно использование двух стратегий. Согласно первой из них при попадании в тупик дальнейшее движение муравья прекращается, а пройденный им путь не помечается феромоном как не приведший к искомой цели. Частично обойти ограничение на тупики возможно обойти путем возврата на один или несколько уровней комбинаторного дерева вверх с запретом движения текущего муравья в тупиковый город, однако данный подход может потребовать большего числа действий, особенно при прокладке первого маршрута. Согласно второй стратегии муравью разрешается движение по запрещенным путям, однако соответствующие значения длин путей для запрещенных переходов выбираются равными заведомо большому значению *V*. При этом путь, включающий в своем составе запрещенные переходы, будет обладать гораздо большей длиной *L* по сравнению с допустимыми путями, что обеспечит для него малую прибавку феромона $\Delta\tau$ и, как следствие, малую вероятность использования в дальнейшем.

Перед практической реализацией рассмотренных выше стратегий необходимо определиться со значениями настроечных параметров $\alpha, Q, \tau^{(0)}, \beta, V$ алгоритма муравьиной колонии, где $\tau^{(0)}$ – начальное значение феромона на дугах графа. Для этого был проведена вычислительных экспериментов, в рамках которой для выборки графов с параметрами $K = 1\,000$, $N = 10$, $d = 0{,}5$ был реализован метод покоординатного спуска с целью оптимизации значения критерия $\overline{L} \to \min$. В результате выполненной параметрической оптимизации были получены следующие значения настроечных параметров, используемые в дальнейших экспериментах: $\alpha^* = 0{,}28$, $Q^* = 1$, $\tau^{(0)*} = 100$, $\beta^* = 0{,}9$, $V^* = 1\,000$.

При сравнительном анализе качества получаемых решений были использованы алгоритм Дейкстры (сокр. O, от англ. **O**ptimal), жадный алгоритм (сокр. G, **G**reedy), алгоритм случайного перебора (сокр. RM, **R**andom search with **M** iterations) [10], алгоритм случайного перебора с возвратами (сокр. RMR, **R**andom search **M** iterations with **R**eturns) [9], алгоритм взвешенного случайного перебора (сокр. WRM, **W**eighted **R**andom search **M** iterations) [8], алгоритм взвешенного случайного перебора с выходами из тупиков (сокр. WRMR, **W**eighted **R**andom search **M** iterations with **R**eturns) и алгоритм муравьиной колонии (сокр. AC, **A**nt **C**olony) с использованием двух рассмотренных выше стратегий реакции на тупики (AC – без реакции на тупики, ACR – с возвратом из тупика, UVLP (сокр. **U**se **V**ery **L**ong **P**athes) – с разрешением на перемещение по запрещенным дугам). Вычислительные эксперименты выполнены на выборках графов объемом $K = 10\,000$, полученные результаты приведены в таблицах 1–4.

Таблица 1. Результаты вычислительного эксперимента при $N = 10$, $d = 0{,}9$

Метод	$\overline{L}$	$\Delta\overline{L}$	$\overline{p}$	$\overline{p}_{opt}$	$\overline{C}$
O	0,2897	0	1	1	1
G	0,8213	0,5325	0,9918	0,2711	1
RM	0,2950	0,0054	0,9918	0,9237	117
RMR	0,2952	0,0055	1	0,9307	117

WRM	0,2901	0,0006	0,9940	0,9860	29
WRMR	0,2901	0,0006	0,9940	0,9860	29
AC	0,2992	0,0096	1	0,8966	12
ACR	0,2993	0,0097	1	0,9855	12
UVLP	0,3000	0,0104	1	0,8944	12

Таблица 2. Результаты вычислительного эксперимента при $N = 10$, $d = 0{,}5$

Метод	$\overline{L}$	$\Delta\overline{L}$	$\overline{p}$	$\overline{p}_{opt}$	$\overline{C}$
O	0,4864	0	1	1	1
G	1,1019	0,6322	0,8558	0,2810	1
RM	0,4821*	0,0004	0,8535	0,8500	108
RMR	0,4866	0,0002	1	0,9975	108
WRM	0,4765*	0,0010	0,8479	0,8436	22
WRMR	0,4873	0,0009	0,9998	0,9940	24
AC	0,4933	0,0069	0,9998	0,9504	11
ACR	0,4965	0,0101	0,9998	0,9367	10
UVLP	0,4964	0,0101	0,9997	0,9352	10

* Замечание. Кажущееся противоречие, заключающееся в том, что средняя длина пути для эвристических методов оказывается меньше, чем у оптимального, легко разрешается, т.к. она в данном случае считается только для тех графов, на которых было найдено решение. В наиболее сложных случаях эвристические методы не всегда обеспечивают отыскание решения, чему соответствует значение $\overline{p} < 1$.

Таблица 3. Результаты вычислительного эксперимента при $N = 10$, $d = 0{,}2$

Метод	$\overline{L}$	$\Delta\overline{L}$	$\overline{p}$	$\overline{p}_{opt}$	$\overline{C}$
O	1,0412	0	0,8179*	0,8179*	1
G	1,0577	0,2513	0,4052	0,2747	1
RM	0,9535	0,0629	0,4042	0,3291	73
RMR	1,0412	0,0000	0,8179	0,8179	73
WRM	0,8846	0,0422	0,4069	0,3485	8
WRMR	1,0423	0,0011	0,8179	0,8166	5
AC	1,0415	0,0003	0,8179	0,8166	7
ACR	1,0417	0,0004	0,8179	0,8170	3
UVLP	1,0436	0,0027	0,8173	0,8107	7

* Замечание. Для выбранных условий тестирования не для всех графов возможно отыскание требуемого пути, т.к. некоторые из сгенерированных псевдослучайным образом графов оказываются несвязными: для них решение не существует.

Таблица 4. Результаты вычислительного эксперимента при $N = 50$, $d = 0{,}5$

Метод	$\overline{L}$	$\Delta\overline{L}$	$\overline{p}$	$\overline{p}_{opt}$	$\overline{C}$
O	0,1634	0	1	1	1
G	1,5576	1,3952	0,9803	0,0839	1
RM	0,2876	0,1243	0,9787	0,2788	536
RMR	0,2874	0,1240	1	0,2853	536
WRM	0,1721	0,0089	0,9806	0,7892	197
WRMR	0,1727	0,0092	1	0,8023	200
AC	0,2060	0,0403	1	0,5429	25
ACR	0,2073	0,0421	1	0,5331	25
UVLP	0,2069	0,0436	1	0,5164	25

Полученные результаты позволяют сделать ряд выводов. Прежде всего следует отметить, что жадный подход демонстрирует наихудшие результаты по средней длине пути в найденных решениях. Методы случайного и взвешенного случайного перебора демонстрируют неплохое качество получаемых решений и вероятности их отыскания. Алгоритм муравьиной колонии демонстрирует качество решений, сопоставимое с более простыми методами случайного перебора, причем первая стратегия (прекращение движения при попадании в тупик) демонстрирует чуть более высокое качество получаемых решений. По видимому это связано с тем, что большая часть некорректных решений отбрасывается на более ранних итерациях, что обеспечивает уменьшение арности узлов комбинаторного дерева и, как следствие, увеличение вероятности отыскания корректного решения, не содержащего в своем составе запрещенных переходов между вершинами. Модификация алгоритма путем добавления возможности выхода из тупиков оказывает незначительное влияние на качество получаемых решений и в данном подходе не является целесообразной. При наличии ограничений на движение по дугам (малые значения плотности графа d) и сравнительно большой размерности задачи N алгоритм муравьиной колонии уступает методу взвешенного случайного перебора с возвратами.

В перспективе дальнейших исследований необходимо выполнение скрупулезного анализа графов как с различным числом вершин, так и с различной плотностью с целью убедиться в справедливости сформулированных выводов и получить экспериментальные зависимости значений средневыборочных параметров от указанных параметров псевдослучайных графов, на основании чего можно сформулировать рекомендации об областях преимущественной применимости рассматриваемых эвристических методов. Данная задача потребует существенно б**о**льших затрат вычислительного времени и, учитывая ее слабую связность, может быть эффективно реализована с использованием грид.

Библиографический список

1. https://ru.wikipedia.org/wiki/Равенство_классов_P_и_NP
2. Валяев В.В., Ватутин Э.И. Метод определения изоморфизма графов общего вида за полиномиальное время // Известия Юго-Западного государственного университета. Серия: Управление, вычислительная техника, информатика. Медицинское приборостроение. № 2. Ч. 1. 2012. С. 200–206.
3. Ватутин Э.И. Проектирование логических мультиконтроллеров. Синтез разбиений параллельных граф-схем алгоритмов. Saarbrucken: Lambert Academic Publishing, 2011 г. 292 с.
4. Ватутин Э.И., Зотов И.В. Метод формирования субоптимальных разбиений параллельных управляющих алгоритмов // Параллельные вычисления и задачи управления PACO'04. М.: ИПУ РАН, 2004. С. 884–917.
5. Dorigo M. Optimization, Learning and Natural Algorithms // PhD thesis. Politecnico di Milano, Italie, 1992.
6. Dijkstra E. W. A note on two problems in connexion with graphs // Numerische Mathematik. V. 1 (1959), PP. 269–271.
7. Зыков А. А. Основы теории графов. М.: Наука, 1986. 384 с.
8. Ватутин Э.И., Дремов Е.Н., Мартынов И.А., Титов В.С. Метод взвешенного случайного перебора для решения задач дискретной комбинаторной оптимизации // Известия Волгоградского государственного технического университета. Серия: Электроника, измерительная техника, радиотехника и связь. № 10 (137). Вып. 9. 2014. с. 59–64.
9. Ватутин Э.И., Мартынов И.А., Титов В.С. Способ обхода тупиков при решении задач дискретной оптимизации с ограничениями // Перспективные информационные технологии (ПИТ-2014). Самара: изд-во Самарского научного центра РАН. С. 313–317.
10. Ватутин Э.И., Колясников Д.В., Мартынов И.А., Титов В.С. Метод случайного перебора в задаче построения разбиений граф-схем

параллельных алгоритмов // Многоядерные процессоры, параллельное программирование, ПЛИС, системы обработки сигналов. Барнаул: Барнаул, 2014. С. 115–125.

11. Ватутин Э.И., Валяев С.Ю., Дремов Е.Н., Мартынов И.А., Титов В.С. Расчетный модуль для тестирования комбинаторных оптимизационных алгоритмов в задаче поиска кратчайшего пути в графе с использованием добровольных распределенных вычислений // Свидетельство о государственной регистрации программы для ЭВМ № 2014619797 от 22.09.14.