

## Разработка больших сетей связи: оптимизационные проблемы и эвристики

Б. Ф. Мельников<sup>1</sup>, Ю. Ю. Терентьева<sup>2\*</sup>

<sup>1</sup> Совместный университет МГУ – ППИ, г. Шэньчжэнь, Китайская Народная Республика 517182, Китайская Народная Республика, провинция Гуандун, г. Шэньчжэнь, р-н Лунган, Даюнь-синьчэн, ул. Гоцзидасюеюань, д. 1

<sup>2</sup> ФГАНУ «Центр информационных технологий и систем органов исполнительной власти», г. Москва, Российская Федерация

123557, Российская Федерация, г. Москва, ул. Пресненский вал, д. 19, стр. 1

\*terjul@mail.ru

### Аннотация

Настоящая статья обобщает и продолжает предыдущие статьи авторов. Мы рассматриваем несколько моделей оптимизационных задач, возникающих при построении больших сетей связи. Топология сети связи формализуется неориентированным графом. Координаты вершин неориентированного графа (узлы сети связи), как правило, каким-то способом заданы заранее, и для этого множества вершин должно быть построено множество рёбер. Основную (и иногда единственную) цель – как наших предыдущих статей, так и настоящей статьи – можно сформулировать следующим образом: для некоторых специальных дополнительных требований нужно построить удовлетворяющее этим требованиям множество рёбер графа, имеющее минимальную суммарную длину. Другой важной идеей является модификация стандартных алгоритмов работы с графами – для того, чтобы иметь возможность рассматривать динамические модели (когда входные данные немного изменяются), причём должна быть возможность при изменении входных данных использовать результаты предыдущих вычислений.

С точки зрения теории графов все эти задачи давно решены – однако на практике реализация соответствующих алгоритмов сопряжена с большими трудностями: во-первых, в реальных условиях мы рассматриваем графы, состоящие из нескольких тысяч вершин и ребер; во-вторых, как мы уже отметили, построение больших сетей связи обычно включает в себя динамическое изменение требований. Следствием двух приведённых обстоятельств является то, что часто даже алгоритмы квадратичной сложности часто не могут быть применены.

Для каждой из наших моделей оптимизационных задач мы приводим один или несколько возможных алгоритмов (в первую очередь эвристических), предназначенных для её решения. Предлагаемые нами алгоритмы, как правило, итерационные – и этот факт хорошо вписывается в возможности, необходимые для построения алгоритмов, работающих с динамически изменяющимися данными.

**Ключевые слова:** коммуникационные сети, большая размерность, алгоритм Краскала, эвристические алгоритмы, выделенное множество вершин, anytime-алгоритмы.

*Авторы заявляют об отсутствии конфликта интересов.*

**Для цитирования:** Мельников, Б. Ф. Разработка больших сетей связи: оптимизационные проблемы и эвристики / Б. Ф. Мельников, Ю. Ю. Терентьева. – DOI 10.25559/SITITO.17.202101.727 // Современные информационные технологии и ИТ-образование. – 2021. – Т. 17, № 1. – С. 69-79.

© Мельников Б. Ф., Терентьева Ю. Ю., 2021



Контент доступен под лицензией Creative Commons Attribution 4.0 License.  
The content is available under Creative Commons Attribution 4.0 License.



## Development of Large Communication Networks: Optimization Problems and Heuristics

B. F. Melnikov<sup>a</sup>, Yu. Yu. Terentyeva<sup>b\*</sup>

<sup>a</sup> Shenzhen MSU – BIT University, Shenzhen, People's Republic of China

1 Guoji-daxueyuan St., Dayunxincheng, Longgang District, Shenzhen 517182, Guangdong Province, People's Republic of China

<sup>b</sup> Center for Information Technologies and Systems of Executive Authorities, Moscow, Russian Federation  
19-1 Presnensky Val St., Moscow 123557, Russian Federation

\* terjul@mail.ru

### Abstract

This paper summarizes and continues the previous papers of the authors. We consider some models of optimization problems that arise when building large communication networks. The topology of the communication network is formalized by an undirected graph. The coordinates of the vertices of an undirected graph (nodes of the communication network) are usually set in advance in some way, and a set of edges must be constructed for this set of vertices. The main (and sometimes the only) goal, both of our previous articles and of this article, can be formulated as follows: for some special additional requirements, it is necessary to construct a set of edges of the graph that satisfies these requirements; these edges should have the minimum possible total length. Another important idea is to modify the standard algorithms for working with graphs in order to be able to consider dynamic models (when the input data changes slightly), and it should be possible to use the results of previous calculations when changing the input data. From the point of view of graph theory, all these problems have long been solved, but in practice, the implementation of the corresponding algorithms is fraught with great difficulties: first, in real conditions, we consider graphs consisting of several thousand vertices; second, as we have already noted, the construction of large communication networks usually involves a dynamic change in requirements. The consequence of the two circumstances given is that often even algorithms of quadratic complexity often cannot be applied. For each of our models of optimization problems, we present one or more possible algorithms (primarily heuristic ones) designed to solve it. Our proposed algorithms are usually iterative, and this fact fits well into the capabilities needed to build algorithms that work with dynamically changing data.

**Keywords:** communication networks, large dimension, Kruskal algorithm, heuristic algorithms, selected set of vertices, anytime algorithms.

*The authors declare no conflict of interest.*

**For citation:** Melnikov B.F., Terentyeva Yu.Yu. Development of Large Communication Networks: Optimization Problems and Heuristics. *Sovremennyye informacionnyye tehnologii i IT-obrazovanie* = Modern Information Technologies and IT-Education. 2021; 17(1):69-79. DOI: <https://doi.org/10.25559/SITITO.17.202101.727>



## Введение

Рассматриваемые в настоящей статье задачи имеют практическую значимость и напрямую связаны с обеспечением устойчивости сети связи<sup>1</sup> [1]. Важность поиска оптимальных решений объясняется прежде всего масштабами реальных сетей связи, формальное представление которых мы рассматриваем. А именно, высокая себестоимость прокладки новых линий связи (эквивалентом которых выступают ребра графа с весовой функцией, равной метрическому расстоянию между вершинами графа) стимулирует разработку эффективных алгоритмов формирования новых ребер графа сети связи, удовлетворяющих ряду заданных требований.

Настоящая статья обобщает и продолжает статьи авторов [2-7]. Мы рассматриваем несколько моделей оптимизационных задач, возникающих при построении больших сетей связи, а также несколько возможных алгоритмов (в первую очередь эвристических), предназначенных для решения этих задач. По-видимому, основную идею статей [3-7], а также настоящей статьи, можно сформулировать следующим образом: для заданного графа необходимо построить множество ребер графа, имеющее минимальную суммарную длину и удовлетворяющее специальным дополнительным требованиям.

В большинстве требований, соответствующих рассматриваемым нами задачам, мы считаем, что координаты вершин неориентированного графа (узлы сети связи) каким-то способом заданы заранее – и для этого множества вершин должно быть построено множество ребер. Как правило, целью является разработка топологии сети с минимально возможным значением суммы весов ребер; однако в некоторых случаях (в некоторых постановках задач) мы применяем эвристические алгоритмы, при этом целью является описание алгоритмов, работающих за приемлемое время и дающих результаты, близкие к оптимальным.

Кроме того, в рассматриваемых нами практических задачах (в наших предыдущих статьях были сформулированы две из них, но ими область применения не исчерпывается) подобный критерий минимальности часто не является единственной целевой функцией.

Итак, мы рассматриваем несколько моделей оптимизационных задач, возникающих при построении больших сетей связи. Узлы сети связи, как правило, каким-то способом заданы заранее, и для этого множества вершин должно быть построено множество ребер. Основную (и иногда единственную) цель – как наших предыдущих статей, так и настоящей статьи – можно сформулировать следующим образом: для некоторых специальных дополнительных требований нужно построить удовлетворяющее этим требованиям множество ребер графа, имеющее минимальную суммарную длину. Другой важной идеей является модификация стандартных алгоритмов работы с графами – для того, чтобы иметь возможность рассматривать динамические модели (когда входные данные немного изменяются), причём должна быть возможность при измене-

нии входных данных использовать результаты предыдущих вычислений.

С точки зрения теории графов все эти задачи давно решены – однако на практике реализация соответствующих алгоритмов сопряжена с большими трудностями: во-первых, в реальных условиях мы рассматриваем графы, состоящие из нескольких тысяч вершин; во-вторых, как мы уже отметили, построение больших сетей связи обычно включает в себя динамическое изменение требований. Следствием двух приведённых обстоятельств является то, что часто даже алгоритмы квадратичной сложности часто не могут быть применены.

Для каждой из наших моделей оптимизационных задач мы приводим один или несколько возможных алгоритмов (в первую очередь эвристических), предназначенных для её решения. Предлагаемые нами алгоритмы, как правило, итерационные – и этот факт хорошо вписывается в возможности, необходимые для построения алгоритмов, работающих с динамически изменяющимися данными.

Краткое содержание статьи таково. В разделе 2 мы рассматриваем т. н. «первую» модель – построение минимального остовного дерева – а для неё приводим два возможных алгоритма. В разделе 3 рассматриваем «вторую» модель: построение пути между двумя заданными вершинами графа, также с минимальной суммарной длиной ребер; приводится алгоритм её решения, который можно рассматривать как специальную модификацию алгоритма Краскала. В разделе 4 мы рассматриваем «третью» модель: соединение вершины с выделенным множеством вершин одним возможным путём; мы также приводим возможный эвристический алгоритм её решения. В разделе 5 рассматриваем «четвёртую» модель: соединение вершины с выделенным множеством вершин несколькими возможными путями.

Итак, разделом 5 кончаются наши описания моделей задач, связанных с алгоритмами построения сетей связи, а также описания этих алгоритмов. Далее в разделе 6 мы приводим некоторые результаты вычислительных экспериментов. Раздел 7 – заключение, в нём мы приводим некоторые направления дальнейших исследований, связанные с рассмотренной в статье тематикой.

## «Первая» модель: построение минимального остовного дерева

В этом разделе с некоторыми изменениями приведена модель из [3]; она несколько отличается от описанного в предыдущем разделе – но, конечно, мы имеем в виду построение связного графа с минимальной суммарной длиной ребер.

Некоторые временные улучшения практической работы алгоритма получаются в результате различных вариантов использования сложных структур данных. Например, мы можем каким-то образом хранить некоторое количество неиспользуемых ребер небольшой длины, а при необходимости – сортировать эти ребра, добавлять к ним новые и т. д. Однако такой

<sup>1</sup> ГОСТ Р 53111-2008. Устойчивость функционирования сети связи общего пользования. Требования и методы проверки: национальный стандарт РФ: издание официальное: утвержден и введен в действие Приказом Федерального агентства по техническому регулированию и метрологии от 18 декабря 2008 г. № 529-ст: введен впервые: дата введения 2009-10-01 / подготовлен ОАО «МТТ», ФГУП «ЦНИИС», ФГУ «ГНИИИ ПТЗИ ФСТЭК России». Москва: Стандартинформ, 2009.



подход не является «панaceей», поскольку в худшем случае оценки сложности (и времени работы алгоритмов) совпадают. Всё это формулирует необходимость рассмотрения и реализации эвристических алгоритмов – вместо точных, «переборных». Поэтому предмет настоящей статьи можно сформулировать следующим образом: Мы переходим от точных алгоритмов к их некоторым эвристическим аналогам. Более того, для большинства рассматриваемых нами исходных задач мы вообще не можем реализовать реально работающие программы – без применения некоторых эвристик.

**Задача 1.** (Построение связанного графа для заданного множества вершин с минимальной суммарной длиной рёбер)

Входные данные:

- $N$ ;
- $V = \{v_1, \dots, v_N\}$  – множество вершин (будущего) графа; примечание: здесь мы не разделяем вершины на классы (обозначаемые нами  $A, B, C, I$  <sup>2</sup>);
- $\text{Coord} : V \rightarrow \mathbf{R} \times \mathbf{R}$  – координаты каждой из вершин.

Объект построения (допустимое решение):

- $E = \{e = (v', v'') \mid v', v'' \in V\}$ , при котором  $(V, E)$  – связный (неориентированный) граф.

Целевая функция:

- $g(E) = \sum_{e \in E} |v', v''|$ , где  $|v', v''|$  = (формула теоремы Пифагора – корня квадратного их суммы квадратов разностей соответствующих координат  $\text{Coord}$ ).

Цель:  $g(E) \rightarrow \min$ .

(Конец описания задачи.)

Оптимальное решение задачи находится с помощью самого простого жадного алгоритма. А именно, среди пар  $(v', v'') \in V$ , ещё не включённых в множество  $V$  (рассматриваемое как множество пар вершин) выбирается такая, что во-первых,  $v'$  и  $v''$  в уже построенном графе <sup>3</sup> принадлежат разным компонентам связности, и, во-вторых, среди всех таких пар выбирается минимальная по длине (длине образуемого ей ребра).

**Алгоритм 1.1** (для задачи 1)

1. Формирование исходного множества рёбер  $E = \emptyset$ .
2. Формирование исходного отображения  $\text{pom} : V \rightarrow N$  (текущий номер компоненты связности) следующим образом: для некоторой (любой) нумерации вершин множества  $V$  от 1 до  $N$  считаем  $\text{pom}(v_i) = i$ .
3. Выбор пары  $(v', v'')$ , такой что:
  - $(v', v'') \notin E$ ,
  - $\text{pom}(v') \neq \text{pom}(v'') \notin E$ ,
  - (формула теоремы Пифагора) для этой пары имеет минимально возможное значение (при выполнении двух предыдущих подпунктов).
4. Включение выбранной пары в множество  $E$ .
5. Для всех  $v \in V$ , таких что  $\text{pom}(v) = \text{pom}(v'')$ , переприсваиваем:  $\text{pom}(v) := \text{pom}(v')$ .
6. Если  $|\text{pom}_{v \in V}(v)| \geq 2$  (т. е. при применении функции  $\text{pom}$  к элементам  $V$  мы получаем более 1 варианта), то переход на п.3.
7. Выход из алгоритма со значением суммы длин (каждая из которых – тоже по теореме Пифагора считается).

<sup>2</sup> Подробнее ниже.

<sup>3</sup> Т. е. – в итерационных алгоритмах – в построенной части графа.

(Конец описания алгоритма.)

Несложно доказывается, что этот жадный алгоритм действительно даёт оптимальное решение.

Немного забегаая вперёд, отметим, что мы будем применять похожие жадные алгоритмы в каждом из следующих случаев:

- 1) в самом простом – т. е. при выборе связанной сети минимальной суммарной длины с заданными координатами вершин графа;
- 2) при наличии «ранжирования» вершин: в этом случае мы строим связанную сеть «на вершинах  $A$ » (употребляя, конечно, и остальные вершины в качестве «промежуточных»), потом – на вершинах  $B$ , и в последнюю очередь – на (оставшихся) вершинах  $C$ ;
- 3) при наличии подмножества вершин, которые уже (заранее) объединены в сеть, с которой мы не работаем: все эти вершины для этого варианта мы можем считать связанными между собой рёбрами длины 0;
- 4) при требовании нескольких путей между некоторыми парами вершин (отметим, что в наших постановках количество таких путей всегда считается на основе надёжности или живучести).

При этом жадный алгоритм обеспечивает оптимальный вариант конфигурации будущей сети в большинстве вариантов задания сети (т. е. при различных формулировках дополнительных требований). Во всех этих случаях работа жадных алгоритмов заключается в выборе ребра минимальной стоимости (в подавляющем большинстве случаев эту стоимость ребра надо считать равной его длине); это «минимальное» ребро выбирается среди всех возможных (а подмножества возможных рёбер – они различаются в каждой из рассматриваемых задач).

Однако при простейшей реализации именно такого варианта, т. е. алгоритма 1, – возникают временные проблемы. А именно, перебор пар вершин занимает порядка  $N^2$  временных шагов – т. е. таковой является оценка времени для одного шага алгоритма (п.п.3–5 алгоритма; точнее п.3); поэтому общее выполнение алгоритма (поскольку число получаемых рёбер в  $E$  будет практически совпадать с числом вершин в  $V$ ,  $|E| = |V| - 1$ ), то общая сложность алгоритма будет порядка  $N^3$ . Это для  $N=3000$  даст более 30 млрд элементарных шагов; как показывает практика, на современной технике такие вычисления (для  $N=3000$ ) в принципе возможны, однако занимают 2–3 часа машинного времени и поэтому вряд ли приемлемы. (Поскольку входные данные нередко меняются при работе над одним и тем же проектом.)

В связи с последним нами предлагается такое усовершенствование предыдущего алгоритма – выполняющего фактически ту же самую работу.

**Алгоритм 1.2** (для задачи 1)

1. Формирование исходного множества рёбер  $E = \emptyset$ .
2. Формирование исходного отображения  $\text{pom} : V \rightarrow N$  (текущий номер компоненты связности) следующим образом: для некоторой (любой) нумерации вершин множества  $V$  от 1 до  $N$  считаем  $\text{pom}(v_i) = i$ .



3. Формирование исходного множества ближайших вершин  $\text{near} : V \rightarrow V$ .<sup>4</sup>

4. Выбор пары  $(v', v'')$ , такой что:

–  $(v', v'') \notin E$ ,

–  $\text{pom}(v') \neq \text{pom}(v'') \notin E$ ,

– (формула теоремы Пифагора) для этой пары имеет минимально возможное значение (при выполнении двух предыдущих подпунктов); используем функцию  $\text{near}$ .

5. Включение выбранной пары в множество  $E$ .

6. Для всех  $v \in V$ , таких что  $\text{pom}(v) = \text{pom}(v'')$ , переписываем:  $\text{pom}(v) := \text{pom}(v')$ .

7. Корректировка значений  $\text{near}(v')$  и  $\text{near}(v'')$ ; новые значения (ближайшие вершины) выбираем только среди вершин с несовпадающими значениями  $\text{pom}$ .

8. Если  $|\text{pom}_{v \in V}(v)| \geq 2$  (т. е. при применении функции  $\text{pom}$  к элементам  $V$  мы получаем более 1 варианта), то переход на п.4.

9. Выход из алгоритма со значением суммы длин (каждая из которых – тоже считается по теореме Пифагора).

(Конец описания алгоритма.)

Как несложно видеть, вычислительная сложность последнего алгоритма  $\sim N^2$  (поскольку сложность шага, т. е. пунктов 4–7, есть  $N$ ).

И, как мы уже отмечали, некоторые результаты вычислительных экспериментов будут приведены ниже в разделе 6.

## «Вторая» модель: построение пути между двумя заданными вершинами графа с минимальной суммарной длиной рёбер

Несколько дальнейших постановок задач (причём постановок) будут включать не создание связных графов – а построение пути из одной вершины в другую (точнее, не столько их создание, сколько построение). Это построение будет рассматриваться как следующий вспомогательный алгоритм; отметим ещё раз, что он будет использован в нескольких последующих задачах. При этом путь желательно получить оптимальный (которым, как правило, будем называть минимально возможный по суммарной длине входящих в него рёбер) – однако, согласно, например, [8, 9, 10] и др., даже алгоритм Дейкстры способен выполнить это за время, пропорциональное  $N^3$  (что, конечно, неприемлемо). Поэтому при решении этой задачи, по-видимому, необходимы только эвристические алгоритмы. Сначала формально опишем саму постановку задачи; отметим заранее, что в ней появляются т. н. «табуированные» рёбра (т. е. те, которые в строимый путь включать нельзя) – однако, конечно, постановка задачи общая: при необходимости множество табуированных дуг может быть пустым.

В этом разделе – также с некоторыми изменениями – приведена модель из [3].

Задача 2. (построение пути между двумя заданными вершинами графа с минимальной суммарной длиной рёбер)

Входные данные:

- $N$ ;
- $V = \{v_1, \dots, v_N\}$  – множество вершин (строимого) графа; примечание: как и ранее, мы не разделяем вершины на классы  $(A, B, C, I)$
- $ET$  – множество «табуированных» рёбер<sup>5</sup>
- $\text{Coord} : V \rightarrow \mathbf{R} \times \mathbf{R}$  – координаты каждой из вершин
- $v', v''$  – начальная и конечная вершины строимого пути

Объект построения (допустимое решение):

- $M$ ;  $E = \{ (v' = v^1, v^2), (v^2, v^3), \dots, (v^{M-1}, v^M = v'') \mid v^i, v^{i+1} \in V, (v^i, v^{i+1}) \notin ET \}$ .

Целевая функция:

- $g(E) = \sum_{e \in E} |v^i, v^{i+1}|$ , где  $|v^i, v^{i+1}|$  = (формула теоремы Пифагора – корня квадратного их суммы квадратов разностей соответствующих координат  $\text{Coord}$ ).

Цель:  $g(E) \rightarrow \min$ .

(Конец описания задачи.)

Как несложно убедиться, точный алгоритм здесь крайне нежелателен (тем более, что обычно это задача – вспомогательная, поэтому сам алгоритм для этой задачи будет применяться порядка  $N$  раз, а возможно и более). Самые простые эвристики, пригодные для применения при создании алгоритмов решения этой задачи, очевидны: выбрать новую вершину (и по ней добавлять ребро) «рядом» с некоторой рассматриваемой нами вершиной графа.

Однако такой подход, будучи применённым на практике (т. е. к реальным данным, возникающим в рассматриваемых нами задачах), дал плохие результаты: при поиске вершин, находящихся близко к первой заданной (в начале работы – обычно  $v'$ ), но при этом уменьшающих оставшееся расстояние до второй заданной (в начале работы – обычно  $v''$ ), мы нередко (на практике в наших задачах – в 20% случаев и более, что обычно неприемлемо) попадаем в ситуацию, когда выбрать такую «улучшающую» вершину не представляется возможным: все они уже включены

- либо в табуированное множество  $ET$ ,
- либо в уже построенную часть пути.

Немного помогает (но тоже не всегда) эвристика, аналогичная т. н. «шейкерной» сортировке ([8, 10] и др): если в качестве стартовой выбирать любую из двух вершин, между которыми строится путь, согласно некоторым дополнительным предпочтениям, – то мы снижаем вероятность неудачного исхода, но эта вероятность всё равно остаётся равной порядка 10–15% случаев, что тоже неприемлемо.

Поэтому на практике (опять же для рассматриваемых нами данных) мы обычно применяем эвристический алгоритм, аналогичный методу деления пополам – что делается в следующем алгоритме.

### Алгоритм 2.1 (для задачи 2)

Обозначения. Будем считать параметры  $v', v''$  локальными, а все остальные – глобальными. В связи с этим вызов алгоритма будем записывать как  $E := A3(v', v'')$  – при этом  $M$  определяется по построенному множеству  $E$ .

<sup>4</sup> Сложность этого шага –  $N^2$ , однако он применяется один раз, т. е. не увеличивает сложность всего алгоритма, также равную  $N^2$ .

<sup>5</sup> Мы здесь не уточняем причину, по которой ребро не может быть включено в строимый путь. Одна из возможных причин – наличие другого, уже построенного пути, содержащего это ребро – при необходимости построения нескольких независимых путей. Другая возможная причина – обе вершины ребра принадлежат классу  $I$ .



1. Вычисление оптимальных координат промежуточной вершины – как полусумму соответствующих координат вершин  $v'$  и  $v$ .
2. Поиск ближайшей вершины к этой середине; пусть это  $v''$ .
3. Если  $A3(v', v'')$  и  $A3(v'', v)$  успешно завершают работу, то выход из алгоритма со значением  $A3(v', v'') \cup A3(v'', v)$ .
4. Если  $|v', v''|$  не превышает заданную константу (мы пользуемся 100 км / 1.3), то выход из алгоритма со значением  $\{(v', v'')\}$ .
5. Замена вершины  $v''$  на «вторую ближайшую» к выбранной ранее середине.
6. Если  $A3(v', v'')$  и  $A3(v'', v)$  успешно завершают работу, то выход из алгоритма со значением  $A3(v', v'') \cup A3(v'', v)$ .
7. Выход из алгоритма с отрицательным результатом. (Конец описания алгоритма.)

Отметим, что на каждом шаге мы применяем не более двух вариантов построения вершины, «ближайшей к середине» – этот вариант алгоритма был выбран на основе проведённых нами вычислительных экспериментов с реальными данными.

Также отметим, что, конечно, легко задать такое табуированное множество ET, при которых эта эвристика приемлемых решений не даёт (аналогично более простым версиям алгоритмов решения этой задачи, о которых мы упомянули выше, но не приводили). Однако на практике – т. е. для конкретных входных данных и рассматриваемых нами для них задач построения графов – мы таких случаев не наблюдали.

Одно из возможных применений последнего алгоритма (но, как мы уже отмечали, не единственное применение) – это «соединение» (возможно, несколькими рёбрами) вершин класса A с вершинами класса И. При этом возникают табуированные рёбра: при каждом следующем таком «соединении» это – рёбра всех предыдущих «соединений», плюс все рёбра, обе вершины которых принадлежат исходному классу И.

Таким образом, мы будем решать задачи, сформулированные в следующих моделях 3 и 4. (В них нужно обратить внимание на то, что, в отличие от предыдущих задач, некоторое исходное множество рёбер уже имеется.)

### «Третья» модель: соединение вершины с выделенным множеством вершин одним возможным путём

на каждом шаге мы применяем не более двух вариантов построения вершины, «ближайшей к середине» – этот вариант алгоритма был выбран на основе проведённых нами вычислительных экспериментов с реальными данными.

Отметим, что, конечно, легко задать такое табуированное множество ET, при которых эта эвристика приемлемых решений не даёт (аналогично более простым версиям алгоритмов решения этой задачи, о которых мы упомянули выше, но не приводили). Однако на практике (для конкретных входных данных и рассматриваемых нами для них задач построения графов) мы таких случаев не наблюдали.

Одно из возможных применений последнего алгоритма (но, как мы уже отмечали, не единственное применение) – это

«подвязка» вершин класса A к вершинам класса И. При этом возникают табуированные рёбра: при каждой следующей «подвязке» это – рёбра всех предыдущих «подвязок», плюс все рёбра, обе вершины которых принадлежат классу И.

Таким образом, будем решать следующие задачи 3 и 4. (Нужно обратить внимание на то, что, в отличие от предыдущих задач, некоторое исходное множество рёбер уже имеется.)

Задача 3. (Соединение вершины с выделенным множеством одним возможным путём)

Входные данные:

- N;
- $V = \{v_1, \dots, v_N\}$  – множество вершин (строимого) графа
- EE – множество уже имеющихся (априори заданных) рёбер
- ET – множество «табуированных» рёбер
- $\text{Coord} : V \rightarrow \mathbf{R} \times \mathbf{R}$  – координаты каждой из вершин
- $\text{Class} : V \rightarrow \{A, B, C, I\}$  – класс каждой из вершин
- $v'$  – начальная вершина строимого пути

Объект построения (допустимое решение):

- M; E =  $\{ (v' = v^1, v^2), (v^2, v^3), \dots, (v^{M-1}, v^M = v) \mid v^i, v^{i+1} \in V, (v^i, v^{i+1}) \notin ET \}$ .

Целевая функция:

- $g(E) = \sum_{e \in E, e \notin EE} |v^i, v^{i+1}|$ , где  $|v^i, v^{i+1}|$  = (формула теоремы Пифагора – корня квадратного их суммы квадратов разностей соответствующих координат Coord); примечание: важное дополнение к предыдущему – добавленные нижние индексы  $e \notin EE$ , которые показывают, что ранее построенные рёбра (имеющиеся к началу работы алгоритма) – не дают вклада в целевую функцию

Цель:  $g(E) \rightarrow \min$ .

(Конец описания задачи.)

Для решения мы используем естественную модификацию предыдущего алгоритма.

Алгоритм 3.1 (для задачи 3)

Обозначения и соглашения – совпадают с предыдущими.

1. Все вершины класса И специальными флажками помечаем как «ещё не обработанные».
2. Выбор  $v''$  – ближайшей к  $v'$  вершины класса И, которая ещё не обработана. Если таковой нет – то выход из алгоритма с отрицательным результатом.
3. Если  $v'$  и  $v''$  соединены рёбрами из множества EE<sup>6</sup>, то выход из алгоритма с ответом  $\emptyset$  (т. е. мы не добавляем новые рёбра).
4. Вычисление оптимальных координат промежуточной вершины – как полусумму соответствующих координат вершин  $v'$  и  $v''$ .
5. Поиск ближайшей вершины к этой середине; пусть это  $v'''$ .
6. Если  $A3(v', v''')$  и  $A3(v''', v)$  успешно завершают работу, то выход из алгоритма со значением  $A3(v', v''') \cup A3(v''', v)$ .
7. Если  $|v', v'''|$  не превышает заданную константу (мы пользуемся 100 км / 1.3), то выход из алгоритма со значением  $\{(v', v''')\}$ .
8. Замена вершины  $v'''$  на «вторую ближайшую» к выбранной ранее середине.
9. Если  $A3(v', v''')$  и  $A3(v''', v)$  успешно завершают рабо-

<sup>6</sup> Это проверяется несложным вспомогательным алгоритмом.



ту, то выход из алгоритма со значением  $A3(v', v'') \cup A3(v, v'')$ .

10. Переход на п.2.

(Конец описания алгоритма.)

### «Четвёртая» модель: соединение вершины с выделенным множеством вершин несколькими возможными путями

Формулировка этой модели очевидна (поскольку она просто следует из названия раздела и может быть получена небольшим усложнением формулировки, приведённой в предыдущем разделе) – и поэтому мы не будем её приводить. Аналогично – для соответствующего алгоритма: требуется выполнить предыдущий алгоритм, изменяя при этом (после каждой его итерации) множество табуированных вершин ET.

Задача 4. (Соединение вершины с выделенным множеством несколькими возможными путями – сокращённое описание)

Дополнительные входные данные:

- K – число необходимых путей для «подвязки»

(Конец описания задачи.)

Алгоритм 4.1 (для задачи 4)

1.  $EEE := \emptyset$  (формируемое выходное множество рёбер).
2. В цикле от 1 до K: begin
  - выполнить алгоритм 4;
  - в случае неудачного его окончания – выход из алгоритма с отрицательным ответом;
  - добавить ответ (рёбра) выхода алгоритма 4 в множество ET;
  - добавить ответ (рёбра) выхода алгоритма 4 в множество EEE;
 end
3. Выход из алгоритма с ответом EEE.

(Конец описания алгоритма.)

Однако очевидно, что такой алгоритм вряд ли интересен: он слишком прост. В будущем мы предполагаем описать более сложные алгоритмы для этой задачи – см. заключение.

### Некоторые результаты вычислительных экспериментов

Результаты вычислений, приведённые в этом разделе в следующей таблице, ранее частично были даны в [5].

Таблица 1. Результаты экспериментов

Table 1. Experimental results

	1-я эвристика (1.1)		«Препроцессор»		Алгоритм 1.2	
	Время	Badness	Время	Badness	Время	Badness
512	0,076	1,16	0,277	> 4	0,078	1,15
1024	0,269	1,13	0,312	> 4	0,260	1,13
1536	0,566	1,11	0,517	> 4	0,561	1,10
2048	0,979	1,12	0,925	> 4	0,923	1,12
3072	2,29	1,09	1,18	> 4	2,12	1,09
4096	4,00	1,10	1,59	> 4	3,81	1,10
6144	6,12	1,10	2,02	> 3	5,93	1,09

Вот некоторые комментарии к приведённой таблице:

- 1-й столбец – размерность;
- 2-й и 3-й столбцы – применение алгоритма 1.1 (по терминологии из [5] – первой эвристики); отметим, что результаты [5] несколько улучшены (за счёт изменения алгоритма плюс, возможно, «более удачных» случайно сгенерированных входных данных) – но незначительно;
- 4-й и 5-й столбцы («Препроцессор») – то, что в [5] было названо «второй эвристикой»; видно, что результаты хуже, чем в алгоритме 1.1 – однако, по нашему мнению, применение быстрой препроцессорной обработки<sup>7</sup> может дать положительные результаты;
- 6-й и 7-й столбцы – применение алгоритмы 1.2;

- всюду время считается в секундах, «Badness» – ухудшение результата по сравнению с оптимальным ответом.

### Заключение. Некоторые направления дальнейших исследований

Таким образом, у нас возникает несколько похожих задач, которые удаётся решать схожими алгоритмами. Большинство из этих алгоритмов – повторим их основную идею – построены на том, чтобы вызвать вспомогательный алгоритм построения одного пути между двумя заданными вершинами графа. При этом такое построение для разных задач и разных алгоритмов осуществляется с разными ограничениями

<sup>7</sup> Это фактически последовательность правых задач, про которую будет кратко сказано ниже. И понятно.



- либо изначально заданными,
- либо формируемыми в процессе решения задач.

Однако уже выше, при самом описании этих задач, мы сформулировали некоторые проблемы; все эти проблемы возникают из-за возможного «плохого» расположения точек на плоскости – точнее, из-за плохого выбора множества точек, представляющего собой вершины формируемого графа. Мы уже описали проблему, приводящую к заклиниванию при выборе вершин вблизи рассматриваемой – и в реальных задачах процент неудач при формировании пути может достигать 20%. Этот процент в рассматриваемых нами задачах удалось существенно снизить (причём мы продолжали применять жадные эвристики) – за счёт применения метода «деления пополам», аналога дихотомии.

Однако в некоторых ситуациях хочется добиться ещё большего улучшения. Не исключены ситуации<sup>8</sup>, в которых оптимальные пути должны идти «в обход» центра, находимого между некоторыми двумя важными вершинами (этот центр может быть на «пустой» территории). Поэтому в качестве альтернативного варианта мы предлагаем применение муравьиных алгоритмов ([11] – «классика»; см. также [12,13,14,15,16,17,18]) – которые необходимо «подключить» к рассматриваемым нами задачам<sup>9,10</sup>. Стоит отметить, что эти алгоритмы хорошо отражают построение сети с требуемой «живучестью».

Как отмечается в теоретических работах, важное преимущество муравьиных алгоритмов по сравнению с другими алгоритмами оптимизации состоит в возможности адаптации к динамическим средам – что полезно и для рассматриваемого нами класса задач, поскольку входные данные – не постоянны, а меняются в процессе разработки алгоритмов решения задач: частые изменения входных данных в нашем случае обычны, и к ним приходится приспосабливаться [19-23].

В нашей ситуации мы собираемся применять<sup>11</sup> муравьиные алгоритмы в качестве дополнительных эвристик, применяемых в случае необходимости точного расчёта живучести. В отличие от жадных алгоритмов (в каждом из которых, как было отмечено ранее, «минимальное» ребро всегда выбирается среди всех возможных, т. е. алгоритм выбора – линейный, поэтому сложность общего алгоритма, каждого, – квадратная), название «муравьиный» требует значительно большего уточнения. Приведём сначала *неформальное описание* одного конкретного варианта – а именно, для задачи построения сети с требуемой живучестью важных направлений; мы сводим её к задаче построения такой сети, в которой для каждого из этих направлений имеется информация о числе путей. Причём:

- это число зависит только от входных данных
- практически одинаковые задачи возникают в случае,

когда надёжность считается обычным образом, по рёбрам, – или другим способом, по вершинам;

- при всех известных нам входных данных нам необходимо составить 5 независимых путей.

Итак, для построенной (или строимой) сети нам надо найти независимые пути – по которым фактически и определяются и надёжность и живучесть.

В самой простой реализации алгоритма<sup>12</sup> для вычисления числа путей мы запускаем рой муравьёв из одной вершины важного направления в другую. Они (объекты-муравьи) в процессе работы алгоритма «знают», куда идти быстрее всего (эту информацию необходимо построить первоначально; возможно и заполнение соответствующей матрицы «вручную») – но ведь не всё (и не у всех объектов-муравьёв) у них получается: либо занята дуга, «кто-то другой по ней пошёл» ... (Мы используем обычное соглашение: по одной конкретной дуге в один конкретный момент дискретного времени может идти только один муравей.)

Поэтому после того, как очередной объект-муравей выберет, куда надо идти, после ограничений вычисляем то, куда он всё-таки на самом деле пошёл; и на этом ребре он оставляет т. н. феромонный след. Именно из этих следов – и образуется первый оптимальный путь – в котором, вероятность описанных выше «проблем» (неудачной работы алгоритма), по-видимому, должна быть равна 0 (точнее, равна 0 в случае, если рассматриваемая задача вообще имеет хоть какое-то допустимое решение). Поэтому мы просто выбираем путь с наибольшим количеством феромона.

А далее (т. е. для построения нескольких независимых путей) возможны такие два варианта:

- либо выбрать «второй» по количеству феромона путь;
- либо<sup>13</sup> мы применяем тот же самый муравьиный алгоритм, имея на входе оставшуюся часть графа и уже выбранные рёбра в качестве табуированных.

Также кратко перечислим и другие возможные задачи, про которые мы предполагаем в будущем привести статьи. Во-первых – это построение т. н. матрицы надёжности – причём *одновременно* с построением всей сети. Во-вторых – создание других алгоритмов для приведённой нами 4-й модели – более сложных, чем последовательное применение нескольких алгоритмов для 3-й модели, но, однако, более простых, чем муравьиные. И – это, видимо, самое главное направление дальнейших работ – добавление к любой из описанных нами задач и любой из описанных для неё эвристик метода ветвей и границ – для которого эта наша эвристика может быть рассмотрена как построение т. н. последовательности правых задач,

<sup>8</sup> Мы в процессе решения (работы программ) их не наблюдали – однако, по-видимому, они не исключены...

<sup>9</sup> Bullnheimer B., Hartl R.F., Strauss C. A New Rank Based Version of the Ant System. A Computational Study. Working Paper No. 1. Wien, Austria, 1997. URL: <https://epub.wu.ac.at/616> (дата обращения: 05.02.2021).

<sup>10</sup> Stützle T., López-Ibáñez M., Pellegrini P., Maur M., Montes de Oca M., Birattari M., Maur M., Dorigo M. Parameter Adaptation in Ant Colony Optimization. Technical Report No. TR/IRIDIA/2010-002. Université Libre de Bruxelles, 2010. URL: <https://iridia.ulb.ac.be/IridiaTrSeries/link/IridiaTr2010-002.pdf> (дата обращения: 05.02.2021).

<sup>11</sup> Реально такое применение уже выполняется, мы предполагаем написать о нём в одной из следующих публикаций.

<sup>12</sup> Конечно, возможны и другие варианты; нужно выполнить конкретные реализации и сравнить их на практике, т. е. посмотреть, какой именно вариант лучше.

<sup>13</sup> Если есть достаточное количество рабочего (процессорного) времени. По-видимому, это время есть всегда: в реальных задачах мы строим не более 5 независимых путей. При этом значение 5 получается по сложным формулам, в которых осуществляется подсчёт надёжности строимой сети связи.



[24-25] и др.

Кроме того, следует обозначить еще одну актуальную задачу, которая важна в сетях связи и которая в то же время не является широко распространенной задачей в теории графов. Это задача нахождения максимального числа независимых путей, а также самих путей между заданными вершинами графа. Это NP-сложная задача, и ее решение потребует разработки новых оптимизационных алгоритмов. Важность этой задачи заключается еще и в том, что пути, построенные рассмотренными ранее алгоритмами (в частности, «четвертой» модели, см. п.5), необходимо эффективно находить в произвольный момент времени. Как показывает практика, для реальных сильно разветвленных сетей связи традиционные алгоритмы нахождения кратчайших путей не дают удовлетворительных результатов, поскольку «теряют» пути, столь важные для устойчивости сети. Поэтому данная задача также будет предметом дальнейших исследований и разработок.

### Список использованных источников

- [1] Dutta, B. The stability and efficiency of directed communication networks / B. Dutta, M. Jackson. – DOI 10.1007/PL00013688 // Review of Economic Design. – 2000. – Vol. 5, issue 3. – Pp. 251-272.
- [2] Melnikov, V. F. Implementation of optimality criteria in the design of communication networks / V. F. Melnikov, V. Y. Meshchanin, Y. Y. Terentyeva. – DOI 10.1088/1742-6596/1515/4/042093 // Journal of Physics: Conference Series. Engineering and Innovative Technologies. – 2020. – Vol. 1515. – Article 042093.
- [3] Булынин, А. Г. Оптимизационные задачи, возникающие при проектировании сетей связи высокой размерности, и некоторые эвристические методы их решения / А. Г. Булынин, Б. Ф. Мельников, В. Ю. Мещанин, Ю. Ю. Терентьева. – DOI 10.34219/2078-8320-2020-11-1-34-40 // Информатизация и связь. – 2020. – № 1. – С. 34-40. – Рез. англ.
- [4] Булынин, А. Г. Алгоритмы проектирования сетей связи с применением жадных эвристик разных типов // Информационные технологии и нанотехнологии (ИТНТ-2020) / А. Г. Булынин, Б. Ф. Мельников, В. Ю. Мещанин, Ю. Ю. Терентьева // Информационные технологии и нанотехнологии (ИТНТ-2020): Сборник трудов по материалам VI Международной конференции и молодежной школы / Под ред. В. А. Фурсова. – Самара: Самарский университет, 2020. – С. 856-860. – URL: <https://www.elibrary.ru/item.asp?id=43576630> (дата обращения: 05.02.2021). – Рез. англ.
- [5] Мельников, Б. Ф. Построение коммуникационных сетей: о применении алгоритма Краскала в задачах больших размерностей / Б. Ф. Мельников, Ю. Ю. Терентьева // International Journal of Open Information Technologies. – 2021. – Т. 9, № 1. – С. 13-21. – URL: <https://www.elibrary.ru/item.asp?id=44584129> (дата обращения: 05.02.2021). – Рез. англ.
- [6] Мельников, Б. Ф. Построение оптимального остовного дерева как инструмент для обеспечения устойчивости сети связи / Б. Ф. Мельников, Ю. Ю. Терентьева. – DOI 10.21685/2072-3059-2021-1-4 // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2021. – № 1(57). – С. 36-45. – Рез. англ.
- [7] Терентьева, Ю. Ю. Некоторые теоретические вопросы, практических алгоритмов дефрагментации сети связи / Ю. Ю. Терентьева // International Journal of Open Information Technologies. – 2021. – Т. 9, № 3. – С. 23-27. – URL: <https://www.elibrary.ru/item.asp?id=44853580> (дата обращения: 05.02.2021). – Рез. англ.
- [8] Алгоритмы: построение и анализ / Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. – М. Вильямс, 2010.
- [9] Липский, В. Комбинаторика для программистов / В. Липский. – М., Мир, 1979.
- [10] Вирт, Н. Алгоритмы и структуры данных. Новая версия для Оберона / Н. Вирт. – М.: ДМК Пресс, 2010.
- [11] Dorigo, M. Ottimizzazione, apprendimento automatico, ed algoritmi basati su metafora naturale: Doctoral dissertation in Systems and Information Electronic Engineering. – Politecnico di Milano, 1992.
- [12] Colorni, A. Distributed Optimization by Ant Colonies / A. Colorni, M. Dorigo, V. Maniezzo // Proceedings of the European Conference on Artificial Life (ECAL'91) / ed. by F. Varela, P. Bourgine. – Paris, France, Elsevier Publishing, 1991. – Pp. 134-142.
- [13] Dorigo, M. Ant system: optimization by a colony of cooperating agents / M. Dorigo, V. Maniezzo, A. Colorni. – DOI 10.1109/3477.484436 // IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics). – 1996. – Vol. 26, no. 1. – Pp. 29-41.
- [14] Gambardella, L. M. Ant-Q: A Reinforcement Learning Approach to the Traveling Salesman Problem / L. M. Gambardella, M. Dorigo. – DOI 10.1016/B978-1-55860-377-6.50039-6 // Machine Learning Proceedings. – Morgan Kaufmann, 1995. – Pp. 252-260.
- [15] Dorigo, M. Ant colony system: a cooperative learning approach to the traveling salesman problem / M. Dorigo, L. M. Gambardella. – DOI 10.1109/4235.585892 // IEEE Transactions on Evolutionary Computation. – 1997. – Vol. 1, no. 1. – Pp. 53-66.
- [16] Stützle, T. MAX-MIN Ant System and local search for the traveling salesman problem / T. Stützle, H. Hoos. – DOI 10.1109/ICEC.1997.592327 // Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC '97). – Indianapolis, IN, USA, 1997. – Pp. 309-314.
- [17] Coltorti, D. Ant Colony Optimization for Real-World Vehicle Routing Problems / D. Coltorti, A. E. Rizzoli. – DOI 10.1145/1329465.1329466 // SIGEVolution. – 2007. – Vol. 2, issue 2. – Pp. 2-9.
- [18] Stützle, T. Parameter Adaptation in Ant Colony Optimization / T. Stützle. – DOI 10.1007/978-3-642-21434-9\_8 // Autonomous Search / ed. by Y. Hamadi, E. Monfroy, F. Saubion. – Springer, Berlin, Heidelberg, 2011. – Pp. 191-215.
- [19] Мельников, Б. Ф. Мультиэвристический подход к задачам дискретной оптимизации / Б. Ф. Мельников // Кибернетика и системный анализ. – 2006. – № 3. – С. 32-43.
- [20] Amir, C. A fuzzy logic controller for ant algorithms / C. Amir, A. Badr, I. Farag // Computing and Information Systems. – 2007. – Vol. 11, issue 2. – Pp. 26-34. – URL: <https://scholar>



- su.edu eg/?q=c-ahmed/files/antcolonies.pdf (дата обращения: 05.02.2021).
- [21] Cai, Z. Ant colony optimization based on adaptive volatility rate of pheromone trail / Z. Cai, H. Huang, Y. Qin, X. Ma. – DOI 10.4236/ijcns.2009.28092 // International Journal of Communications, Network and System Sciences. – 2009. – Vol. 2, issue 8. – Pp. 792-796.
- [22] Li, Y. Adaptive ant colony optimization algorithm based on information entropy: Foundation and application / Y. Li, W. Li // Fundamenta Informaticae. – 2007. – Vol. 77, no. 3. – Pp. 229-242.
- [23] Merkle, D. Ant colony optimization for resource-constrained project scheduling / D. Merkle, M. Middendorf, H. Schmeck. – DOI 10.1109/TEVC.2002.802450 // IEEE Transactions on Evolutionary Computation. – 2002. – Vol. 6, no. 4. – Pp. 333-346.
- [24] Tavares Neto, R. F. Literature review regarding Ant Colony Optimization applied to scheduling problems: Guidelines for implementation and directions for future research / R. F. Tavares Neto, M. Godinho Filho. – DOI 10.1016/j.engappai.2012.03.011 // Engineering Applications of Artificial Intelligence. – 2013. – Vol. 26, issue 1. – Pp. 150-161.
- [25] Practical Genetic Algorithms / R. L. Haupt, S. E. Haupt. – 2nd ed. – John Wiley & Sons, Inc., 2004.
- [3] Bulynin A.G., Melnikov B.F., Meshchanin, V.Y., Terentyeva. Y.Y. Optimization problem, arising in the development of high-dimensional communication networks, and some heuristic methods for solving them. *Informatization and Communication*. 2020; (1):34-40. (In Russ., abstract in Eng.) DOI: <https://doi.org/10.34219/2078-8320-2020-11-1-34-40>
- [4] Bulynin A.G., Melnikov B.F., Meshchanin, V.Y. and Terentyeva. Y.Y. Algorithms for designing communication networks using greedy heuristics of different types. *Proceedings of the International Conference on Information Technology and Nanotechnology (ITNT-2020)*. Samara University, Samara; 2020. p. 856-860. Available at: <https://www.elibrary.ru/item.asp?id=43576630> (accessed 05.02.2021). (In Russ., abstract in Eng.)
- [5] Melnikov B.F., Terentyeva. Y.Y. Building communication networks: on the application of the Kruskal's algorithm in the problems of large dimensions. *International Journal of Open Information Technologies*. 2021; 9(1):13-21. Available at: <https://www.elibrary.ru/item.asp?id=44584129> (accessed 05.02.2021). (In Russ., abstract in Eng.)
- [6] Melnikov B.F., Terentyeva. Y.Y. Construction of an optimal spanning tree as a tool to ensure the stability of the communication network. *University proceedings. Volga region. Technical sciences*. 2021; (1):36-45. (In Russ., abstract in Eng.) DOI: <https://doi.org/10.21685/2072-3059-2021-1-4>
- [7] Terentyeva Y.Y. Some theoretical issues related to the description of practical algorithms for constructing spanning trees. *International Journal of Open Information Technologies*. 2021; 9(3):23-27. Available at: <https://www.elibrary.ru/item.asp?id=44853580> (accessed 05.02.2021). (In Russ., abstract in Eng.)
- [8] Cormen T., Leiserson C., Rivest R., Stein C. Introduction to Algorithms. The MIT Press; 2009. (In Eng.)
- [9] Lipsky V. *Kombinatorika dlia programmistov* [Combinatorics for programmers]. Mir, Moscow; 1979. (In Russ.)
- [10] Wirth N. Algorithms and Data Structures. Oberon version. Prentice-Hall; 2004. (In Eng.)
- [11] Dorigo M. *Ottimizzazione, apprendimento automatico, ed algoritmi basati su metafora naturale* [Optimization, Learning, and Natural Algorithms]: Doctoral dissertation in Systems and Information Electronic Engineering. Politecnico di Milano, 1992. (In Italian)
- [12] Colorni A., Dorigo M., Maniezzo V. Distributed Optimization by Ant Colonies. In: F. Varela, P. Bourguine (Eds.) *Proceedings of the European Conference on Artificial Life (ECAL'91)*. Paris, France, Elsevier Publishing; 1991. p. 134-142. (In Eng.)
- [13] Dorigo M., Maniezzo V., Colorni A. Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*. 1996; 26(1):29-41. (In Eng.) DOI: <https://doi.org/10.1109/3477.484436>
- [14] Gambardella L.M., Dorigo M. Ant-Q: A Reinforcement Learning Approach to the Traveling Salesman Problem. *Machine Learning Proceedings*. Morgan Kaufmann; 1995. p. 252-260. (In Eng.) DOI: <https://doi.org/10.1016/B978-1-55860-377-6.50039-6>
- [15] 15. Dorigo M., Gambardella L.M. Ant colony sys-

Поступила 05.02.2021; одобрена после рецензирования  
14.03.2021; принята к публикации 25.03.2021.

#### Об авторах:

**Мельников Борис Феликсович**, профессор факультета вычислительной математики и кибернетики, Совместный университет МГУ – ППИ (517182, Китайская Народная Республика, провинция Гуандун, г. Шэньчжэнь, р-н Лунган, Даюньсиньчэн, ул. Гоцзидасюэюань, д. 1), доктор физико-математических наук, профессор, **ORCID:** <http://orcid.org/0000-0002-6765-6800>, [bormel@mail.ru](mailto:bormel@mail.ru)

**Терентьева Юлия Юрьевна**, начальник управления анализа и методологии совершенствования информационных телекоммуникационных систем, ФГАНУ «Центр информационных технологий и систем органов исполнительной власти» (123557, Российская Федерация, г. Москва, ул. Пресненский вал, д. 19, стр. 1), кандидат технических наук, **ORCID:** <http://orcid.org/0000-0002-2418-003X>, [terjul@mail.ru](mailto:terjul@mail.ru)

Все авторы прочитали и одобрили окончательный вариант рукописи.

## References

- [1] Dutta B., Jackson M. The stability and efficiency of directed communication networks. *Review of Economic Design*. 2000; 5(3):251-272. (In Eng.) DOI: <https://doi.org/10.1007/PL00013688>
- [2] Melnikov B.F., Meshchanin V.Y., Terentyeva Y.Y. Implementation of optimality criteria in the design of communication networks. *Journal of Physics: Conference Series. Engineering and Innovative Technologies*. 2020; 1515:042093. (In Eng.) DOI: <https://doi.org/10.1088/1742-6596/1515/4/042093>



- tem: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*. 1997; 1(1):53-66. (In Eng.) DOI: <https://doi.org/10.1109/4235.585892>
- [16] Stützle T., Hoos H. MAX-MIN Ant System and local search for the traveling salesman problem. *Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC '97)*. Indianapolis, IN, USA; 1997. p. 309-314. (In Eng.) DOI: <https://doi.org/10.1109/ICEC.1997.592327>
- [17] Coltorti D., Rizzoli A.E. Ant Colony Optimization for Real-World Vehicle Routing Problems. *SIGEVOLUTION*. 2007; 2(2):2-9. (In Eng.) DOI: <https://doi.org/10.1145/1329465.1329466>
- [18] Stützle T. et al. Parameter Adaptation in Ant Colony Optimization. In: Y. Hamadi, E. Monfroy, F. Saubion (Eds.) *Autonomous Search*. Springer, Berlin, Heidelberg; 2011. p. 191-215. (In Eng.) DOI: [https://doi.org/10.1007/978-3-642-21434-9\\_8](https://doi.org/10.1007/978-3-642-21434-9_8)
- [19] Melnikov B.F. *Mul'tijevristicheskij podhod k zadacham diskretnoj optimizacii* [A multi-heuristic approach to discrete optimization problems]. *Cybernetics and System Analysis*. 2006; (3):32-43. (In Russ.)
- [20] Amir C., Badr A., Farag I. A fuzzy logic controller for ant algorithms. *Computing and Information Systems*. 2007; 11(2):26-34. Available at: <https://scholar.cu.edu.eg/?q=c-ahmed/files/antcolonies.pdf> (accessed 05.02.2021). (In Eng.)
- [21] Cai Z., Huang H., Qin Y., Ma X. Ant colony optimization based on adaptive volatility rate of pheromone trail. *International Journal of Communications, Network and System Sciences*. 2009; 2(8):792-796. (In Eng.) DOI: <https://doi.org/10.4236/ijcns.2009.28092>
- [22] Li Y., Li W. Adaptive ant colony optimization algorithm based on information entropy: Foundation and application. *Fundamenta Informaticae*. 2007; 77(3):229-242. (In Eng.)
- [23] Merkle D., Middendorf M., Schmeck H. Ant colony optimization for resource-constrained project scheduling. *IEEE Transactions on Evolutionary Computation*. 2002; 6(4):333-346. (In Eng.) DOI: <https://doi.org/10.1109/TEVC.2002.802450>
- [24] Tavares Neto R.F., Godinho Filho M. Literature review regarding Ant Colony Optimization applied to scheduling problems: Guidelines for implementation and directions for future research. *Engineering Applications of Artificial Intelligence*. 2013; 26(1):150-161. (In Eng.) DOI: <http://dx.doi.org/10.1016/j.engappai.2012.03.011>
- [25] Haupt R.L., Haupt S.E. *Practical Genetic Algorithms*. 2nd ed. John Wiley & Sons, Inc.; 2004. (In Eng.)

*Submitted 05.02.2021; approved after reviewing 14.03.2021;  
accepted for publication 25.03.2021.*

#### About the authors:

**Boris F. Melnikov**, Professor of the Faculty of Computational Mathematics and Cybernetics, Shenzhen MSU – BIT University (1 Guoji-daxueyuan St., Dayunxincheng, Longgang District, Shenzhen 517182, Guangdong Province, People's Republic of China), Dr.Sci. (Phys.-Math.), Professor, **ORCID:** <http://orcid.org/0000-0002-6765-6800>, [bormel@mail.ru](mailto:bormel@mail.ru)

**Yulia Yu. Terentyeva**, Head of the Department of Analysis and Methodology for Improving Information and Telecommunications Systems, Center for Information Technologies and Systems of Executive Authorities (19-1 Presnensky Val St., Moscow 123557, Russian Federation), Ph.D. (Engineering), **ORCID:** <http://orcid.org/0000-0002-2418-003X>, [terjul@mail.ru](mailto:terjul@mail.ru)

*All authors have read and approved the final manuscript.*

