УДК 004.021; 004.023; 519.17
doi:10.21685/2072-3040-2022-4-3

**О графовой модели для задач рефлектометрии
и некоторых алгоритмах их решения.
Часть III. Подход к генерации тестовых данных
и результаты вычислительных экспериментов**

Б. Ф. Мельников¹, Ю. Ю. Терентьева²

^{1,2}Центр информационных технологий и систем
органов исполнительной власти, Москва, Россия

¹bormel@mail.ru, ²terjul@mail.ru

Аннотация. *Актуальность и цели.* Актуальность рассматриваемой предметной области обусловлена прежде всего необходимостью минимизации стоимости так называемых рефлектометров при имеющемся ограничении на условие тотального мониторинга волоконно-оптических кабелей. Подобные задачи возникают при проектировании и/или модернизации сети связи, причем они особенно важны в тех ситуациях, когда сеть связи имеет очень большую размерность. Целью является исследование возможности применения метода ветвей и границ в нескольких схожих постановках задачи рефлектометрии. *Материалы и методы.* Применены эвристические алгоритмы искусственного интеллекта и дискретной оптимизации, объединенные в единый программный пакет, а также статистические методы анализа алгоритмов. *Результаты.* Результатами являются закономерности, полученные при применении жадной эвристики и вариантов метода ветвей и границ при решении задач рефлектометрии. *Выводы.* Были предложены алгоритмы, описывающие улучшение метода ветвей и границ с помощью подключения к нему различных вспомогательных эвристик. Однако полученное временное улучшение среднего времени работы этого алгоритма в рассмотренной нами прикладной задаче – по сравнению с жадным алгоритмом – очень невелико, и это позволяет сделать предварительные выводы о том, что в задачах рефлектометрии достаточным является применение простейших жадных алгоритмов.

Ключевые слова: эвристические алгоритмы, задачи дискретной оптимизации, модели теории графов, жадный алгоритм, метод ветвей и границ

Для цитирования: Мельников Б. Ф., Терентьева Ю. Ю. О графовой модели для задач рефлектометрии и некоторых алгоритмах их решения. Часть III. Подход к генерации тестовых данных и результаты вычислительных экспериментов // Известия высших учебных заведений. Поволжский регион. Физико-математические науки. 2022. № 4. С. 31–41. doi:10.21685/2072-3040-2022-4-3

**On a graph model for reflectometry problems and some
algorithms for their solution. Part 3. An approach to test data
generation and some results of computational experiments**

B.F. Mel'nikov¹, Yu.Yu. Terent'eva²

^{1,2}Centre of Information Technologies and Systems
for Executive Authorities, Moscow, Russia

¹bormel@mail.ru, ²terjul@mail.ru

Abstract. *Background.* The relevance of the subject area under consideration is primarily due to the need to minimize the cost of so-called reflectometers, with the existing restriction on the condition of total monitoring of fiber-optic cables. Similar tasks arise when designing and / or upgrading a communication network, and they are especially important in situations where the communication network has a very large dimension. The purpose of the study is to investigate the possibility of using the method of branches and boundaries in some similar formulations of the problem of reflectometry. *Materials and methods.* The paper uses heuristic algorithms of artificial intelligence and discrete optimization, combined into a single software package, as well as statistical methods for analyzing algorithms. *Results.* The results are regularities obtained by applying the greedy heuristics and the version of the method of branches and boundaries in solving problems of reflectometry. *Conclusions.* Algorithms have been proposed describing the improvement of the branch and boundary method by connecting various auxiliary heuristics to it. However, the obtained temporary improvement in the average operating time of this algorithm in the applied problem we have considered, compared to the greedy algorithm, is very small, and this allows us to draw preliminary conclusions that the use of the simplest greedy algorithms is sufficient in reflectometry problems.

Keywords: heuristic algorithms, discrete optimization problems, graph theory models, greedy algorithm, method of branches and boundaries

For citation: Mel'nikov B.F., Terent'eva Yu.Yu. On a graph model for reflectometry problems and some algorithms for their solution. Part III. An approach to test data generation and some results of computational experiments. *Izvestiya vysshikh uchebnykh zavedeniy. Povolzhskiy region. Fiziko-matematicheskie nauki = University proceedings. Volga region. Physical and mathematical sciences.* 2022;(4):31–41. (In Russ.). doi:10.21685/2072-3040-2022-4-3

В представляемой части III настоящей статьи мы заканчиваем рассматривать специальные оптимизационные алгоритмы, связанные с задачами рефлектометрии. Мы приводим возможный подход к генерации репрезентативных тестовых данных, а также предварительные результаты проведенных вычислительных экспериментов.

8. Возможный подход к генерации входных данных для вычислительных экспериментов

В начале этого раздела добавим информацию – дополнительно к приведенной в части II настоящей статьи – об использованных в программах структурах данных. Мы стараемся употреблять общий вариант организации массивов с переменными границами – вариант, в чем-то близкий к классу `SArray` (и ему подобным классам) из библиотеки MFC¹. Более того, все двумерные массивы (в частности, любые варианты матрицы смежности) мы тоже представляем как *одномерный* массив с переменным числом элементов.

Специально отметим, что в подобных структурах данных (в частности, в классах, необходимых для полей кратко описанного в части II класса `Net z`) мы практически всегда используем индексацию начиная с 1, очень удобную для математических алгоритмов – в связи с чем используем специальную вспомогательную функцию индексации. При этом для двумерных массивов,

¹ См. <https://docs.microsoft.com/en-us/cpp/mfc/> и мн. др. При этом непосредственно классы этой библиотеки мы обычно не употребляем, но пытаемся соблюдать ее общую идеологию.

а также иногда встречающихся в наших прикладных задачах трехмерных, мы «убиваем двух зайцев»: с помощью такой функции индексации мы, во-первых, все сводим к одномерным массивам; во-вторых, рассматриваем индексацию начиная с 1, о которой было сказано ранее. (Подобную «идеологию», оформленную как «надстройку» над стандартными классами, мы ранее использовали и непосредственно в МФС.)

Очень важным является следующее обстоятельство: в отличие от большинства рассматривавшихся нами ранее прикладных задач дискретной оптимизации, одной из основных составляющих *именно этой задачи* (а также предмета настоящей статьи) является *генерация входных данных*. По поводу подобной генерации и ее отражении в публикациях наше мнение таково:

1) в подавляющем большинстве этих публикаций генерации внимания не уделено (часто говорится что-то вроде: «...на случайных входных данных были получены следующие результаты...»);

2) имеется подход, в котором авторы предлагают применять очень сложные математические теории. Причем гораздо более сложные, чем, например, оценки сложности алгоритмов «в среднем» и «в худшем», используемые классиками: Н. Виртом [1], Д. Кнутом [2], Т. Корменом с соавторами [3] и др. Среди них в первую очередь отметим работы Л. Левина ([4]¹ и последующие по этой тематике) и Ю. Гуревича ([5] и последующие).

Как можно понять, мы пытаемся «поместиться между этими двумя полюсами». Однако при этом целью является построение некоторой *применимой на практике теории сравнительной оценки эффективности* алгоритмов и соответствующих компьютерных программ.

Одной из важнейших задач для этой теории является создание подхода к генерации входных данных для вычислительных экспериментов – предмет этого раздела. Всегда нужны *репрезентативные* входные данные – и это мы считаем частью того, что принято называть *адекватностью* математической модели, см. [6–9]. Согласно сказанному в этих статьях (конкретно – в [8]), при генерации исходных данных следует учитывать как можно больше особенностей предметной области и адекватно их формализовывать. В рассматриваемой нами задаче для верификации алгоритмов необходимо выполнить следующие этапы генерации тестовых исходных данных:

- определить количество вершин графа модели сети связи;
- определить координаты на плоскости (или реальные географические координаты) для вершин графа;
- определить виды так называемых «лампочек» (рефлектометров) и параметров дальности «просвечивания» этих лампочек.

¹ Вот наш перевод части аннотации к статье [4] (курсив тоже наш):

«...Развивается теория алгоритмической сложности Колмогорова. *Определение случайности модифицировано* таким образом, чтобы полученное понятие удовлетворяло *сильным свойствам инвариантности* (так называемым неравенствам сохранения). Эта модификация позволяет для некоторых бесконечных последовательностей значений определять такие понятия, как *взаимная информация*. Рассматриваются приложения полученной теории к нескольким областям, таким как теория вероятностей, теория алгоритмов...»

По-видимому, даже из приведенного фрагмента понятно, что подобную теорию *практически* невозможно применять к сравнительной *практической* оценке качества работы алгоритмов и соответствующих компьютерных программ.

Отдельного внимания заслуживает вопрос распределения точек-вершин графа на плоскости (или на реальном геопространстве). Как правило, равномерное и нормальное распределение не дают адекватных тестовых наборов входных данных для верификации исследуемых алгоритмов на графах. Это происходит в первую очередь потому, что реальные исходные данные представляют собой *композицию* различных случайных величин с комбинированными распределениями. Отсюда следует, что производительность алгоритма на реальных данных может в общем случае быть существенно ниже, чем на модельных – при заданных модельных параметрах; этот вопрос напрямую касается точности выбираемой модели.

В задачах рефлектотрии мы предлагаем рассматривать как простое равномерное распределение точек-вершин модельного графа, так и вариант с неравномерным распределением, которое будет определяться множеством точек пространства, соответствующих «населенным пунктам», с характеристиками модели, зависящими от численности населения этих пунктов (поскольку на территории население распределено неравномерно, то этим свойством можно воспользоваться для генерации неравномерного расположения точек в геопространстве).

При генерации вершин графа задается верхняя грань параметра численности населения и коэффициент разброса. Варьируя подобные параметры, мы можем получать неравномерное распределение точек на геопространстве, приближенное к реальному распределению.

В рамках рассматриваемой задачи следует проводить тестирование алгоритмов на различных сгенерированных модельных данных – с различными сочетаниями таких параметров.

Например, возможен такой конкретный вариант репрезентативной генерации входных данных. Мы выбираем некоторое множество населенных пунктов на рассматриваемой территории, причем с численностью выше некоторого заданного значения; точки (вершины графа) можно располагать только в этих пунктах. В качестве дополнительного параметра используется некоторое значение, показывающее, насколько далеко от выбранных пунктов можно размещать точку (если, например, задать этот параметр очень большим, то можно приблизиться к равномерному распределению точек-вершин на выделенном пространстве).

Репрезентативность генерации исходных данных может быть исследована с точки зрения анализа случайного процесса показателей тестирования. В частности, может быть рассмотрена случайная величина, меняющаяся во времени, представляющая собой разность целевых функций при использовании двух сравниваемых алгоритмов.

Существует и более широкий смысл понятия «репрезентативность» – и применительно к оценке качества алгоритмов мы об этом также писали в упомянутых статьях, см., в частности, [6, 7]. *Репрезентативность* – одно из базовых понятий в математической статистике. Оно характеризует те параметры, которые обеспечат максимально адекватный результат анализа выборочной совокупности. В широком смысле понятие репрезентативности граничит с мерой соответствия генеральной совокупности и выборкой – насколько точно выборка описывает особенности генеральной исследуемой

совокупности. При этом репрезентативность – величина *измеряемая*, она может быть определена так называемой ошибкой репрезентативности, т.е. разностью между специально выбираемыми характеристиками выборочной и генеральной совокупностей. Окончательным количественным значением репрезентативности обычно является среднее квадратичное полученных ее возможных значений.

Однако фактическая (действительная) величина указанной разности на практике обычно остается неизвестной. Поэтому моделировать структурные объекты – и при этом обеспечивать репрезентативность модели – очень непростая задача.

Для исследования поведения структур и алгоритмов их обработки обычно проводится моделирование составляющих комбинаторных структур, основанное на случайных генерациях комплекса параметров. При этом проверять репрезентативность сгенерированных структур следует с помощью *статистических критериев, специально адаптируемых к данным задачам*. Адекватная модель структуры генерирует объекты, приближенные к реальным. *Степень приближения* также является отдельным предметом исследования процесса моделирования различных процессов и предметных областей. К примеру, генерация конечных автоматов при описании контекстно-свободных языков, проводимая в [6], учитывала широкий спектр параметров моделирования. Это, в частности, необходимо для обеспечения сферы применимости исследуемых алгоритмов (например, при их использовании в алгоритмах LR-анализа). Кроме того, в [6] учитывалось использование конечных автоматов в лексических анализаторах – при компиляции языков, а также в процессах тестирования программного обеспечения на основе исследуемых моделей.

В большинстве других рассматривавшихся нами ранее задачах ([10–13] и др.) мы на репрезентативную генерацию тестовых данных обращали существенно меньше внимания. При этом в процитированных здесь статьях, прежде всего в [12], имеются в виду компьютерные программы, соответствующие рассматриваемым задачам. Подробных публикаций о самих таких программах мы еще практически не приводили, исключением являются те статьи, в которых либо понятия репрезентативности (либо адекватности модели) были вынесены в заглавие [6–9], либо исследуются задачи с заведомо негенерируемыми (в том числе с «однократными») входными данными, которые берутся исключительно из внешних источников ([14–16] и др.); вследствие этого понятие тестовой генерации входных данных в них вряд ли может иметь какой-либо смысл.

Во всех остальных ситуациях мы считаем репрезентативную генерацию входных данных, подобную описанной в этом разделе, необходимой.

Перейдем к непосредственному описанию примененных нами алгоритмов генерации псевдослучайных данных в задачах рефлектотрии. Нами был выбран подход, в котором значения и цены лампочек подбирались так, чтобы в выдаваемый жадным алгоритмом ответ вошли бы все типы лампочек (об этом алгоритме мы писали в части II настоящей статьи). То есть мы *вручную*, причем *одновременно*, выбирали как мощность, так и стоимость лампочек (их количество было выбрано равным 4), а также параметры следующего метода класса Netz (сеть):

```

void InitRnd(int minDia, int maxDia, double proDia);
// minDia - минимальное число для диапазона для добавления рёбер
// maxDia - максимальное число для этого диапазона
// (после выбора конкретного числа из этого диапазона -
// столько попыток добавления делаем)
// proDia - вероятность добавления рассматриваемого ребра;
// рёбра рассматриваются в порядке возрастания длин

```

Те же самые входные данные подавались на вход и более сложным алгоритмам, т.е. более сложным, чем описанный ранее жадный, в частности, различным вариантам метода ветвей и границ (напомним, что краткое описание класса `Netz`, а также описание жадного алгоритма и одного из вариантов алгоритма ветвей и границ, были приведены в части II).

Приведем сначала текст используемого нами метода. Сам алгоритм построения сети при предварительной случайной генерации координат точек, являющихся вершинами сети, заключается в нескольких для каждой вершины попытках добавления ребер. В каждой такой попытке мы выбираем ближайшую к ней из еще нерассмотренных вершин и при «удачном» значении генерируемой равномерной случайной величины действительно добавляем такое ребро.

Дополнительные комментарии:

- `minDia` и `maxDia` задают диапазон, в нем и выбирается (как равномерно распределенная целая случайная величина) число попыток построения ребра. В каждой из этих попыток мы пытаемся добавить ребро к ближайшей из непросмотренных вершин (в случае неудачной попытки вершина считается просмотренной). При этом каждый раз, переходя к некоторой новой вершине, мы все остальные вершины снова объявляем непросмотренными – и снова выполняем заранее оговоренное число попыток добавления ребра:

```

void Netz::InitRnd(int minDia, int maxDia, double proDia) {
    pTA->InitRnd();
    pGR->InitNulla();
    Array* pArr = new Array(nDim);
    for (int i=1; i<=nDim; i++) { // от этой вершины строим новых соседей
        int nKol = RndDiap(minDia,maxDia);
        // число попыток добавления рёбер - начиная от ближайшего
        pArr->InitNulla();
        for (int j=1; j<=nDim; j++) if (j!=i && !pGR->Get(i,j)) pArr->Set1(j);
        // возможная для просмотра вершина
        for (int k=1; k<=nKol; k++) { // k-я попытка добавления
            double rMin = 2000000;
            // текущее минимальное расстояние до i-й вершины
            int jMin = 0;
            for (int j=1; j<=nDim; j++) {
                if (j==i) continue;
                if (pArr->Get(j)==0) continue;
                double r = pTA->Delta(i,j);
                if (r>=rMin) continue;
                rMin = r;
                jMin = j;
            }
            if (jMin==0) continue;
            pArr->Set0(jMin);
            if (!RndBool(proDia)) continue;
            pGR->SetTrue(i,jMin); pGR->SetTrue(jMin,i);
        }
    }
    delete pArr;
}

```

- $proDia$ – сама вероятность добавления ребра (от этого значения, как и от двух предыдущих параметров, зависит «насыщенность» всей сети связи).

- Расстояния представляются целыми значениями от 0 до 10^6 (считаем, что последнее значение соответствует 1, т.е. мы просто рассматриваем единичный квадрат). В связи с этим мы и употребляем константу $2 \cdot 10^6$ (заведомо превышающую диагональ единичного квадрата).

- Сами ближайшие вершины мы ищем простейшим переборным алгоритмом; специальной структуры данных для этого мы не заводим: описанные здесь построения выполняются для каждой сети однократно.

В *конкретных* примененных алгоритмах генерации входных данных все вершины первоначально полагаются доступными для любого вида ламп, и мы задавали четыре вида ламп (у каждого вида – свои яркость и стоимость).

9. Кратко о результатах вычислительных экспериментов, предварительные выводы и заключение

В этом разделе мы обобщаем полученные результаты вычислительных экспериментов. Однако стоит сразу отметить, что в действительности временное улучшение работы метода ветвей и границ по сравнению с жадным алгоритмом ни разу не превысило 6 %.

Проведенные нами вычислительные эксперименты можно кратко описать следующим образом:

- Мы генерируем графы «средней насыщенности», и при этом новые ребра добавляются в первую очередь «к более-менее близким вершинам», т.е. все делается примерно так, как в вычислительных экспериментах многих предыдущих тем, связанных с использованием случайных графов ([8–11] и др.).

- Конкретные применявшиеся константы: размерности 29 и 99; четыре вида ламп.

- Стоимость и мощность этих ламп *вручную* подбирались таким образом, чтобы описанный в части I жадный алгоритм для подавляющего большинства сгенерированных частных случаев проблемы (95 % случаев и более для таких «графов средней насыщенности») использовал бы все четыре вида ламп.

- Этот жадный алгоритм для размерности 99 обрабатывает все сгенерированные частные случаи проблемы менее чем за 1 мин каждый. При этом мы применяли компьютер «средней мощности»: Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz.

Два из полученных нами случайно сгенерированных частных случаев проблемы для размерностей 29 и 99 и параметров функции генерации таковы:

```
pNT->InitRnd(2, 3, 0.7);
```

Мы вручную для четырех видов ламп подобрали значения мощностей (длин) равные соответственно $40 \cdot 10^4$, $18 \cdot 10^4$, $8 \cdot 10^4$ и $2 \cdot 10^4$ (обозначаем длины сторон единичного квадрата как 10^6 , поэтому значение $40 \cdot 10^4$ фактически представляет собой $40 \cdot 10^{4-6} = 0,4$); при применении этих значений жадный алгоритм практически всегда приводил к использованию всех четырех видов ламп. Полученные нами характерные – «*медианные*» – примеры для вышеуказанных двух размерностей таковы:

dim=29, число освещенных=29, cost=196
использовано ламп по типам: 1-1 2-2 3-2 4-5

и

dim=99, число освещенных=99, cost=311
использовано ламп по типам: 1-2 2-3 3-6 4-2

Можно также заметить, что существенное увеличение размерности задачи не приводит ни к значительному увеличению числа ламп в ответе, ни к значительному же увеличению стоимости этого ответа – по крайней мере, при таком способе случайной генерации входных данных. Стоит также отметить, что использование жадного алгоритма почти совпадает с одним из вспомогательных алгоритмов, применяемых в мультиэвристическом подходе к задачам дискретной оптимизации; этот подход фактически является расширением метода ветвей и границ. А именно имеется в виду построение так называемой последовательности правых задач (см. [10, 14] и др.). В предыдущих публикациях мы отмечали, что применение этого вспомогательного алгоритма не приводит к замедлению времени работы метода ветвей и границ.

Использование всех четырех видов ламп дает основания говорить о репрезентативности входных данных – и, сделать *предварительные* выводы о применении аналогичных алгоритмов для значительно больших размерностей. Как мы уже сказали, в наших вычислениях для метода ветвей и границ практически не было получено улучшения работы по сравнению с жадным алгоритмом. У метода ветвей и границ не было и временных задержек (они по сравнению с жадным алгоритмом не превышали 10 %), но не было и значительного улучшения качества решения: так, для размерности 99 для нескольких десятков рассмотренных примеров максимальное улучшение стоимости решения ни разу не превысило 6 %. Заметим еще, что, в части II и в настоящей части III мы фактически привели все важные фрагменты программы – и поэтому повторить наши вычисления можно очень легко – должны получиться результаты, практически совпадающие с нашими. Вследствие этого более подробных результатов вычислений мы не приводим.

Итак, полученное временное улучшение среднего времени работы этого алгоритма в рассмотренной нами прикладной задаче – по сравнению с жадным алгоритмом – *очень невелико*; это позволяет сделать предварительные выводы о том, что *в задачах рефлектометрии достаточным является применение простейших жадных алгоритмов, а не более сложных алгоритмов*, в частности метода ветвей и границ.

Список литературы

1. Вирт Н. Алгоритмы и структуры данных. М. : ДМК Пресс, 2010. 272 с.
2. Кнут Д. Искусство программирования. Т. 2. Полученные алгоритмы. М. : Вильямс, 2013. 832 с.
3. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы. Построение и анализ. М. : Вильямс, 2013. 1324 с.
4. Levin L. Randomness conservation inequalities; information and independence in mathematical theories // Information and Control. 1984. Vol. 61. P. 15–37.
5. Gurevich Y. A logic for constant-depth circuits // Information and Control. 1984. Vol. 61. P. 65–74.
6. Мельников Б. Ф., Пивнева С. В., Рогова О. А. Репрезентативность случайно сгенерированных недетерминированных конечных автоматов с точки зрения соот-

- ветствующих базисных автоматов // Стохастическая оптимизация в информатике. 2010. № 6. С. 74–82.
7. Пивнева С. В., Рогова О. А. Метод случайной генерации недетерминированных конечных автоматов и проверка репрезентативности сгенерированных структур // Вестник Воронежского государственного университета. Серия: Системный анализ и информационные технологии. 2010. № 2. С. 5–8.
 8. Макаркин С. Б., Мельников Б. Ф. Адекватность математических моделей на примере задачи коммивояжера // Философские проблемы информационных технологий и киберпространства. 2013. № 2 (6). С. 4–17.
 9. Мельников Б. Ф., Сайфуллина Е. Ф., Терентьева Ю. Ю., Чурикова Н. П. Применение алгоритмов генерации случайных графов для исследования надежности сетей связи // Информатизация и связь. 2018. № 1. С. 71–80.
 10. Мельников Б. Ф., Сайфуллина Е. Ф. Применение мультиэвристического подхода для случайной генерации графа с заданным вектором степеней // Известия высших учебных заведений. Поволжский регион. Физико-математические науки. 2013. № 3. С. 70–83.
 11. Макаркин С. Б., Мельников Б. Ф. Геометрические методы решения псевдогеометрической версии задачи коммивояжера // Стохастическая оптимизация в информатике. 2013. Т. 9, № 2. С. 54–72.
 12. Melnikov V. F., Korabelshchikova S. Yu., Dolgov V. N. On the task of extracting the root from the language // International Journal of Open Information Technologies. 2019. Vol. 7, № 3. P. 1–6.
 13. Мельников Б. Ф., Терентьева Ю. Ю. Построение оптимального остоного дерева как инструмент для обеспечения устойчивости сети связи // Известия высших учебных заведений. Поволжский регион. Технические науки. 2021. № 1. С. 36–45.
 14. Мельников Б. Ф., Панин А. Г. Параллельная реализация мультиэвристического подхода в задаче сравнения генетических последовательностей // Вектор науки Тольяттинского государственного университета. 2012. № 4 (22). С. 83–86.
 15. Нагорнов Ю. С., Золотов А. В. К подходу моделирования формирования нанокристаллов в процессе карбонизации пористого кремния // Вектор науки Тольяттинского государственного университета. 2012. № 4 (22). С. 89–93.
 16. Melnikov B., Pivneva S. On the multiple-aspect approach to the possible technique for determination of the author's literary style // CEUR Workshop Proceedings. Selected Papers of the 11-th International Scientific-Practical Conference Modern Information Technologies and IT-Education(SITITO–2016). 2016. P. 311–315.

References

1. Virt N. *Algoritmy i struktury dannykh = Algorithms and data structures*. Moscow: DMK Press, 2010:272. (In Russ.)
2. Knut D. *Iskusstvo programmirovaniya. T. 2. Poluchislennyye algoritmy = The art of programming. Volume 2. Polonumeral algorithms*. Moscow: Vi-l'yams, 2013:832. (In Russ.)
3. Kormen T., Leyzerson Ch., Rivest R., Shtayn K. *Algoritmy. Postroenie i analiz = Algorithms. Construction and analysis*. Moscow: Vil'yams, 2013:1324. (In Russ.)
4. Levin L. Randomness conservation inequalities; information and independence in mathematical theories. *Information and Control*. 1984;61:15–37.
5. Gurevich Y. A logic for constant-depth circuits. *Information and Control*. 1984;61:65–74.
6. Mel'nikov B.F., Pivneva S.V., Rogova O.A. Representativeness of randomly generated non-deterministic finite automata in terms of the corresponding basis automata. *Stokhasticheskaya optimizatsiya v informatike = Stochastic optimization in computer science*. 2010;(6):74–82. (In Russ.)

7. Pivneva S.V., Rogova O.A. Method of random generation of non-deterministic finite automata and check of representativeness of generated structures. *Vestnik Voronezhskogo gosudarstvennogo universiteta. Seriya: Sistemnyy analiz i informatsionnye tekhnologii = Bulletin of Voronezh State University. Series: System analysis and information technology*. 2010;(2):5–8. (In Russ.)
8. Makarkin S.B., Mel'nikov B.F. Adequacy of mathematical models on the example of the traveling salesman problem. *Filosofskie problemy informatsionnykh tekhnologiy i kiberprostranstva = Philosophical problems of information technology and cyberspace*. 2013;(2):4–17. (In Russ.)
9. Mel'nikov B.F., Sayfullina E.F., Terent'eva Yu.Yu., Churikova N.P. Application of algorithms for generating random graphs to study the reliability of communication networks. *Informatizatsiya i svyaz' = Informatization and communication*. 2018;(1):71–80. (In Russ.)
10. Mel'nikov B.F., Sayfullina E.F. Application of a multiheuristic approach for random generation of a graph with a given degree vector. *Izvestiya vysshikh uchebnykh zavedeniy. Povolzhskiy region. Fiziko-matematicheskie nauki = University proceedings. Volga region. Physical and mathematical sciences*. 2013;(3):70–83. (In Russ.)
11. Makarkin S.B., Mel'nikov B.F. Geometric methods for solving the pseudo-geometric version of the traveling salesman problem. *Stokhasticheskaya optimizatsiya v informmatike = Stochastic optimization in informatics*. 2013;9(2):54–72. (In Russ.)
12. Melnikov B.F., Korabelshchikova S.Yu., Dolgov V.N. On the task of extracting the root from the language. *International Journal of Open Information Technologies*. 2019;7(3):1–6.
13. Mel'nikov B.F., Terent'eva Yu.Yu. Building an optimal spanning tree as a tool for ensuring communication network stability. *Izvestiya vysshikh uchebnykh zavedeniy. Povolzhskiy region. Tekhnicheskie nauki = University proceedings. Volga region. Engineering sciences*. 2021;(1):36–45. (In Russ.)
14. Mel'nikov B.F., Panin A.G. Parallel implementation of the multiheuristic approach in the problem of comparing genetic sequences. *Vektor nauki Tol'yattinskogo gosudarstvennogo universiteta = Bulletin of Togliatti State University*. 2012;(4):83–86. (In Russ.)
15. Nagornov Yu.S., Zolotov A.V. To the approach of modeling the formation of nanocrystals in the process of carbonation of porous silicon. *Vektor nauki Tol'yattinskogo gosudarstvennogo universiteta = Bulletin of Togliatti State University*. 2012;(4):89–93. (In Russ.)
16. Melnikov B., Pivneva S. On the multiple-aspect approach to the possible technique for determination of the author's literary style. *CEUR Workshop Proceedings. Selected Papers of the 11-th International Scientific-Practical Conference Modern Information Technologies and IT-Education(SITITO–2016)*. 2016:311–315.

Информация об авторах / Information about the authors

Борис Феликсович Мельников

доктор физико-математических наук,
профессор, главный научный сотрудник,
Центр информационных технологий
и систем органов исполнительной
власти (Россия, Москва, ул. Пресненский
Вал, 19, стр. 1)

E-mail: bormel@mail.ru

Boris F. Mel'nikov

Doctor of physical and mathematical
sciences, professor, principal researcher,
Centre of Information Technologies
and Systems for Executive Authorities
(building 1, 19 Presnenskiy
Val street, Moscow, Russia)

Юлия Юрьевна Терентьева

кандидат технических наук, начальник
управления анализа и методологии
совершенствования информационных
телекоммуникационных систем,
Центр информационных технологий
и систем органов исполнительной
власти (Россия, Москва, ул. Пресненский
Вал, 19, стр. 1)

E-mail: terjul@mail.ru

Yuliya Yu. Terent'eva

Candidate of engineering sciences,
head of the department of analysis
and methodology for improving information
telecommunication systems, Centre
of Information Technologies and Systems
for Executive Authorities (building 1,
19 Presnenskiy Val street, Moscow, Russia)

Авторы заявляют об отсутствии конфликта интересов / The authors declare no conflicts of interests.

Поступила в редакцию / Received 01.08.2022

Поступила после рецензирования и доработки / Revised 10.09.2022

Принята к публикации / Accepted 05.10.2022