

АЛГОРИТМ УЛУЧШЕНИЯ ТРАССИРОВКИ ПЕЧАТНЫХ ПЛАТ

А. М. Бершадский¹, П. А. Гудков², Е. М. Подмарькова³

^{1, 2, 3} Пензенский государственный университет, Пенза, Россия
¹bam@pnzgu.ru, ²p.a.gudkov@gmail.com, ³kalibryk.m@gmail.com

Аннотация. *Актуальность и цели.* Рассматривается новый улучшающий алгоритм трассировки, основанный на моделировании динамики частиц. *Материалы и методы.* В качестве входных данных берется имеющийся вариант трассировки, который улучшается в ходе выполнения алгоритма. Для этого предлагается новый алгоритм, когда каждая трасса заменяется цепочкой атомарных частиц, которые, перемещаясь по пространству печатной платы, отведенному для трассировки, испытывают влияние сил притяжения и отталкивания от проходящих рядом трасс и близлежащих элементов. Процесс заканчивается по достижении состояния равновесия получившейся системы элементов. *Результаты.* В результате применения предложенного алгоритма сокращается суммарная длина трасс, учитываются различного рода ограничения на расположение трасс и элементов, уменьшается общая площадь печатной платы за счет более плотной разводки трасс. *Выводы.* Рассмотренный алгоритм предоставляет широкие возможности по оптимизации имеющихся вариантов трассировки. Например, если под действием сил будут перемещаться не только трассы и переходные отверстия, но и сами элементы схемы, то это позволит учесть ограничения на взаимное расположение элементов и тем самым повысить качество проектируемого устройства.

Ключевые слова: алгоритм трассировки, компьютерное моделирование, взаимодействие частиц, печатная плата

Для цитирования: Бершадский А. М., Гудков П. А., Подмарькова Е. М. Алгоритм улучшения трассировки печатных плат // Модели, системы, сети в экономике, технике, природе и обществе. 2021. № 3. С. 80–90. doi:10.21685/2227-8486-2021-3-8

THE IMPROVING ALGORITHM OF PCB TRACING

A.M. Bershadsky¹, P.A. Gudkov², E.M. Podmar'kova³

^{1, 2, 3} Penza State University, Penza, Russia
¹bam@pnzgu.ru, ²p.a.gudkov@gmail.com, ³kalibryk.m@gmail.com

Abstract. *Background.* The new improving algorithm of PCB tracing is considered, which is based on the modeling of the particle dynamics. *Materials and methods.* An existing circuit board traces is taken as input data, and then is improved during the algorithm execution. For this purpose, the authors suggest using a new approach. Each trace is replaced with a chain of elementary particles, which moves through a circuit board tracing space and experience the attraction forces and repulsion forces from nearby traces and other elements. When the resulting system of elements comes to the equilibrium state, the process ends. *Results.* The result of the algorithm is a reduced overall length of traces, the optimal placement of traces and elements, taking into account various restrictions, a more

dense layout of the elements on the printed circuit board. *Conclusions.* The considered approach provides great opportunities for optimizing the existing PCB tracing. For example, it is possible to improve the quality of the designed device, if not only the traces are moved when the forces are influenced, but also the circuit board elements themselves. It will also allow to take into account restrictions on the mutual placement of the elements and therefore improve the quality of the designed device.

Keywords: tracing algorithm, computer simulation, particles interaction, printed circuit board

For citation: Bershadsky A.M., Gudkov P.A., Podmar'kova E.M. The improving algorithm of PCB tracing. *Modeli, sistemy, seti v ekonomike, tekhnike, prirode i obshchestve = Models, systems, networks in economics, technology, nature and society.* 2021;(3):80–90. (In Russ.). doi:10.21685/2227-8486-2021-3-8

Введение

С развитием полупроводниковых технологий постоянно возрастают и требования проектировщиков к алгоритмам трассировки для проектирования печатного монтажа. Трассировкой называется процесс нахождения геометрического расположения всех цепей на печатной плате. Исходными данными для трассировки является список цепей, определяющий требуемые соединения между контактами элементов, а также расположение элементов схемы на печатной плате, определяемое на этапе размещения.

Задача трассировки – одна из наиболее трудоемких в общей проблеме автоматизации проектирования электронных устройств. Это связано с несколькими факторами, в частности с многообразием способов конструктивно-технологической реализации соединений, для каждого из которых при алгоритмическом решении задачи применяются специфические критерии оптимизации и ограничения [1–3].

На практике при оптимизации топологии печатного монтажа часто используют следующие критерии качества: минимум суммарной длины всех соединений, минимум числа пересечений проводников, равномерность распределения проводников на печатной плате, минимальная протяженность параллельных участков соседних проводников, минимум числа изгибов проводников [4]. Так как все перечисленные критерии являются противоречивыми, при выборе обобщенного показателя качества указывают важность или очередность учета отдельных критериев оптимизации.

Имеется несколько классификаций алгоритмов трассировки. По одной из них все методы делятся на трассировку с использованием сеточного представления монтажно-коммутационного пространства (*grid based routing*) или же без него (*gridless routing*). Описываемый далее предлагаемый авторами алгоритм использует второй подход.

Типовой же подход для большинства существующих алгоритмов – разбиение монтажного поля на ячейки, размер и форма которых определяют плотность и конфигурацию печатных проводников. Наибольшее распространение на практике получило разбиение рабочего поля на правильные квадраты, что обеспечивает простую адресацию ячеек в прямоугольной системе координат и привычную форму соединений [4].

Многие алгоритмы трассировки представляют собой модификацию алгоритма Ли, предложенного еще в 1961 г. [5]. Этот алгоритм моделирует всю поверхность печатной платы как прямоугольный массив элементов, размеры которых определяются таким образом, чтобы не нарушать заданных интерва-

лов между прокладываемыми трассами. Трассы образуются по одной, занимая по очереди свободные соседние элементы, от одного контакта цепи до другого. После того, как элемент оказывается занят, он помечается как непригодный для дальнейшего использования (при прокладке оставшихся трасс).

Этапу выбора свободного соседнего элемента предшествует этап расширения «волны» от одной из вершин цепи. Это позволяет строить трассы оптимальной конфигурации, а также дает другое название этого алгоритма – волновой. Основным преимуществом алгоритма Ли является то, что он гарантированно найдет путь от одной точки до другой, если он существует. С другой стороны, из-за своей последовательной природы алгоритм не в состоянии выявить такие побочные эффекты, когда трассировка одного соединения будет влиять на другие. Соответственно, главным недостатком алгоритма является сильная зависимость результата от очередности трассировки цепей – принятие решений о расположении первых трасс приводит к невозможности трассировки последующих.

Для устранения указанного недостатка трассировку иногда выполняют в два этапа. Первый этап называется глобальной трассировкой (*global routing*) и дает лишь примерное расположение трасс для каждой цепи. Фактически он определяет список областей прохождения будущих трасс без указания их конкретного геометрического расположения. На втором этапе, который называется детальной трассировкой (*detailed routing*), выполняется уже обычная трассировка каждой цепи в назначенных областях. Представителями данного класса алгоритмов являются канальные и топологические алгоритмы трассировки [6].

Другая классификация – деление алгоритмов трассировки на последовательные и параллельные. При последовательном подходе, как следует из его названия, цепи трассируются одна за другой. Однако, как упоминалось ранее, проложенная цепь может заблокировать другие, еще не проложенные. Иными словами, эти алгоритмы очень чувствительны к порядку трассировки цепей. Для сглаживания этого недостатка цепи обычно упорядочиваются по их значимости.

На практике, помимо упорядочивания цепей, используется также этап корректировки трасс. Однако и он не может преодолеть недостаток последовательных алгоритмов. В одних алгоритмах для корректировки используется подход «разорвать трассу и проложить ее иначе» [7], в других – подход «сдвинуть трассу в сторону». В первом случае мешающие трассы разрываются и трассируются заново таким образом, чтобы можно было проложить заблокированные цепи. Во втором случае трассы отводятся в сторону без разрыва существующих соединений.

Другой подход состоит в том, чтобы сначала трассировать простые цепи, состоящие из двух-трех контактов [8]. Обычно такие цепи составляют большую часть схемы (до 75 %). Затем трассируются цепи, имеющее большее число контактов, с использованием для этого алгоритма дерева Штейнера. И в самом конце выполняется трассировка оставшихся многоконтактных цепей (таких как питание, земля, тактовые сигналы и т.д.), которых не так много.

Таким образом, выделяют последовательные алгоритмы:

- 1) двухполюсные алгоритмы:
 - волновой алгоритм;
 - лучевой алгоритм;
- 2) многопозиционные алгоритмы:
 - алгоритмы на основе дерева Штейнера.

Как уже упоминалось, волновой алгоритм использует сеточное представление монтажно-коммутационного пространства. Поэтому время его работы линейно зависит от количества ячеек, на которое разбивается пространство печатной платы. Поиск в алгоритме Ли происходит с использованием волны, распространяющейся от источника. Алгоритм выполняет симметричный поиск во всех направлениях до первого нахождения целевой ячейки-приемника. Таким образом гарантируется нахождение кратчайшего пути между любыми двумя вершинами, если такой путь существует. Наихудший вариант – когда источник расположен в центре, а приемник – в углу зоны трассировки, в этом случае будут проверены практически все ячейки, прежде чем целевая ячейка будет достигнута.

Лучевые алгоритмы были независимо разработаны Миками и Табучи в 1968 г. [9] и Хайтауэром в 1969 г. [10]. Основная идея таких алгоритмов – уменьшение объема требуемой памяти за счет использования отрезков линий вместо узлов сетки при поиске. Временная сложность лучевых алгоритмов составляет $O(L)$, где L – количество проведенных отрезков линий.

Немаловажную роль при этом играют структуры данных, применяемые при реализации алгоритма. Обычно используют два списка – один для горизонтальных линий и один для вертикальных. Использование двух отдельных списков позволяет игнорировать параллельные линии, что ускоряет поиск.

В целом же и волновой, и лучевой алгоритмы можно рассматривать как частные случаи алгоритма нахождения кратчайшего пути Дейкстры, позволяющего найти путь минимальной стоимости от одной вершины до другой. Соответственно, такие алгоритмы не подходят для глобальной трассировки многоконтактных цепей. Для решения этой проблемы было предложено несколько подходов в модификации волновых и лучевых алгоритмов. В одних алгоритмах многополюсные цепи разбиваются на несколько простых цепей (имеющих по два контакта), которые трассируются с использованием стандартных алгоритмов трассировки. Качество трассировки при этом зависит от способа разбиения цепи. Такой подход дает неоптимальные результаты, поскольку практически отсутствует какое-либо взаимодействие между фазой декомпозиции и фактической фазой трассировки. Другой подход – выполнять поиск с нескольких контактов одновременно. Это позволяет процессу распространения волны определять, какие пары выводов необходимо соединить, вместо того чтобы использовать заранее определенное разбиение цепи. Однако такие алгоритмы все равно не могут оптимально соединять контакты. Кроме того, они унаследовали большую временную сложность исходных алгоритмов трассировки.

Поэтому появился новый подход в алгоритмах трассировки – на основе дерева Штейнера. Обычно используются прямолинейные деревья Штейнера (имеющие только прямолинейные ребра). Длина дерева – это сумма длин всех ребер в нем. Ее еще называют стоимостью дерева. Проблема поиска минимальной стоимости прямолинейного дерева Штейнера является NP-полной [11]. Поэтому ввиду NP-сложности задачи было разработано несколько эвристических алгоритмов ее решения [12].

Параллельные алгоритмы используют глобальное представление всех цепей, которые должны трассироваться одновременно. Эти алгоритмы требуют использования дорогостоящих в вычислительном плане методов. Один из способов снижения вычислительных затрат – иерархическое разбиение (сверху вниз) задачи на более мелкие подзадачи, которые могут быть решены

эффективно. Затем решения объединяются для получения окончательного решения исходной задачи глобальной трассировки.

Тем самым параллельный подход позволяет избежать проблемы упорядочивания цепей, характерной для последовательных алгоритмов.

Предлагаемый алгоритм

Для решения задачи оптимальной трассировки авторы предлагают использовать следующий алгоритм:

1. За основу берется вариант трассировки, выполненный с использованием одного из стандартных алгоритмов, используемых в современных САПР (Diptrace, Altium Designer и пр.). При этом не предъявляется особых требований к качеству трассировки, поскольку в дальнейшем этот вариант будет улучшаться.

2. Каждая трасса заменяется цепочкой атомарных частиц (четырёхугольных элементов), которые будут перемещаться по пространству, отведенному для трассировки, испытывая влияние сил притяжения и отталкивания от проходящих рядом трасс и близлежащих элементов.

3. В качестве результата трассировки будет взято расположение трасс, получившееся после того, как система придет в состояние равновесия. Условие равновесия для отдельного элемента трассы определяется следующей формулой:

$$\sum_{i=1}^N F_i = 0,$$

где F_i – силы, действующие на элемент. На практике применима формула

$$\sum_{i=1}^N F_i \leq \varepsilon,$$

где ε – некоторая допустимая погрешность, которая подбирается исходя из требуемой точности позиционирования трасс. После того как данное условие будет выполнено для всех элементов, работа алгоритма завершается.

Входные данные для первоначального варианта трассировки задаются в виде графа $G(V, E)$, где $V\{x_i, y_i\}$ – множество координат узловых точек, E – множество связей, определяющих наличие фрагмента трассы между соответствующими точками. Множество V , в свою очередь, состоит из непересекающихся подмножеств $V_1, V_2, \dots$, соответствующих различным цепям электрической схемы.

При загрузке входных данных из файла, экспортированного из какой-либо САПР в формате DSN, сначала выполняется подготовительный этап, на котором происходит «дробление» протяженных участков трасс. При этом каждый отрезок трассы заменяется цепочкой связанных элементов, показанных на рис. 1. После этого начинается основной этап работы алгоритма.

Решаемая авторами задача улучшения трассировки может быть сформулирована следующим образом. Найти такое местоположение элементов на печатной плате, при котором достигается минимум целевой функции:

$$F = \sum_{i=1}^N \sum_{j=1}^N c_{ij} \cdot d_{ij},$$

где d_{ij} – расстояние между i -м и j -м элементами, c_{ij} – значение коэффициента взвешенной связности, принимающего в простейшем случае значения '0' (отсутствие связи) и '1' (элементы являются соседями в одной трассе).

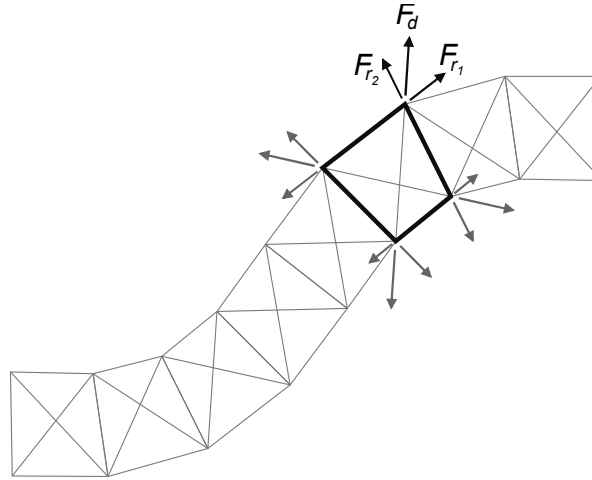


Рис. 1. Силы, действующие на элементы трасс

При этом накладывается ряд ограничений, таких как $d_{ij} \geq d_{\min}$, для обеспечения требуемого технологического зазора между трассами. Также возможно задание каких-то специфичных ограничений, например, чтобы минимизировать тепловое влияние сильно нагревающихся элементов схемы на своих соседей.

Используемая модель атомарных элементов трасс показана на рис. 1. Это четырехугольник, соединенный противоположными сторонами с соседними такими же элементами. Каждая из вершин четырехугольника испытывает помимо внешних сил действие трех внутренних сил (F_{r1} , F_{r2} – вдоль ребер и F_d – по диагонали). Они необходимы для поддержания фиксированной ширины трасс, а также плавности их изгибов при встрече с препятствиями. Кроме того, силы F_d , приложенные к диагоналям, предотвращают «схлопывание» четырехугольника в ромб.

Как известно, движение отдельного элемента под действием приложенных к нему сил можно описать дифференциальным уравнением [13]. Соответственно, имея множество взаимодействующих друг с другом элементов, мы получаем систему дифференциальных уравнений, решив которую, можно найти оптимальное расположение элементов в пространстве, получаемое по достижении системой состояния равновесия:

$$\begin{cases} x_i'' = F_{x_i}(\bar{x}, \bar{y}, x_i', y_i', t), \\ y_i'' = F_{y_i}(\bar{x}, \bar{y}, x_i', y_i', t), \end{cases}$$

где x_i , y_i – координаты i -го элемента; F_{x_i} и F_{y_i} – проекции сил на оси x и y ; t – модельное время.

В разрабатываемом прототипе, реализующем данный алгоритм, используются приближенные методы решения систем дифференциальных уравнений (одна из реализаций метода Эйлера), что позволяет существенно сократить временные затраты и упростить программный код.

Блок-схема алгоритма представлена на рис. 2. Основной цикл выполняется до тех пор, пока система не придет в состояние равновесия. Обычно весь процесс занимает несколько секунд и может визуализироваться в реальном времени.

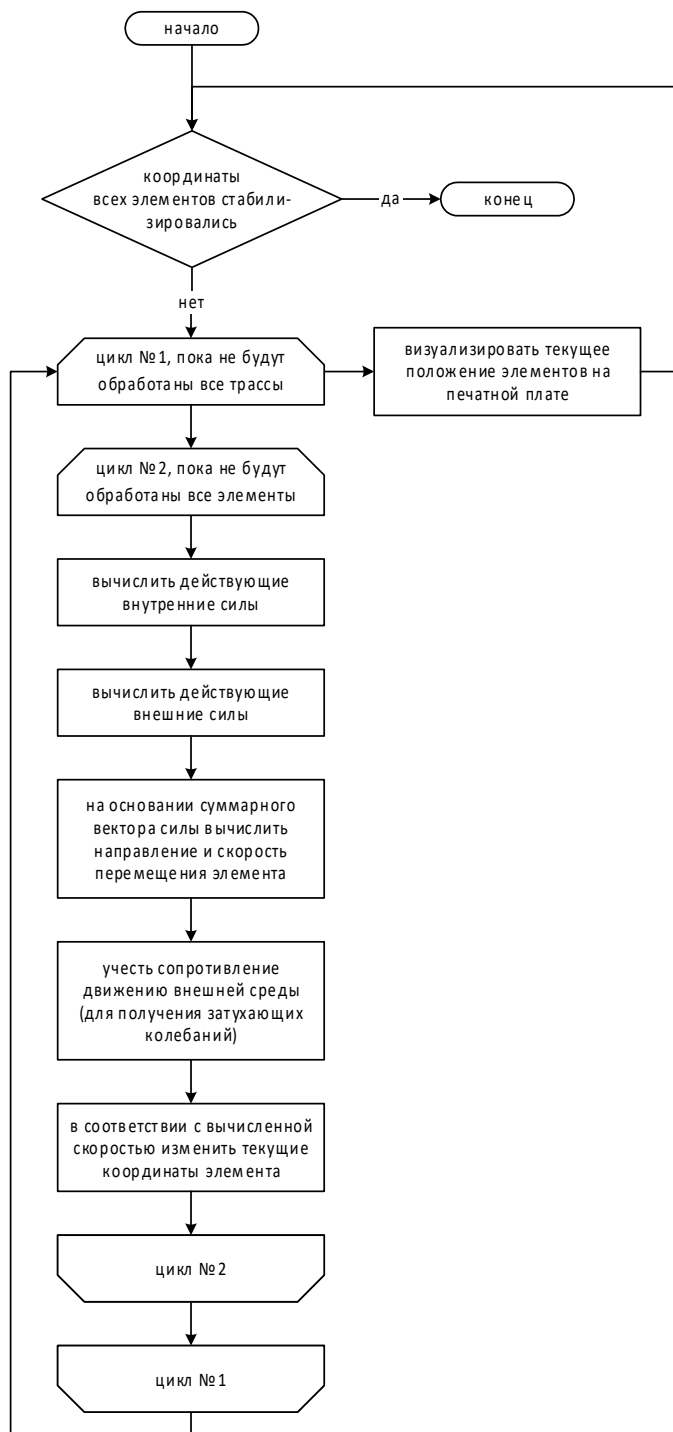


Рис. 2. Блок-схема предлагаемого алгоритма

Результат работы алгоритма (улучшенный вариант трассировки) можно сохранить в формате SES, пригодном для импорта в различные САПР, с целью дальнейшей работы над проектом печатной платы устройства.

Кроме того, как показала практика, при большом временном шаге моделирования возможны затухающие колебательные процессы, иногда приводящие к «перекручиваниям» трасс. Факт такого перекручивания обнаруживается путем вычисления обычного векторного произведения. По знаку произведения можно определить, расположен ли элемент трассы нормально или же «обратной» стороной вверх. Периодически просматривая списки элементов трасс, можно своевременно выявлять такие искажения.

Поскольку такие «перекручивания» трасс устраняются элементарно, путем выполнения перестановок соседних вершин, было принято решение не усложнять алгоритм для предотвращения образования подобных искажений, а устранять их по мере появления. При общем времени работы алгоритма 5–10 с проверки на «перекручивания» трасс можно выполнять, например, один раз в секунду, что практически никак не скажется на производительности алгоритма.

Результат применения предлагаемого алгоритма показан на рис. 3 и 4 на примере схемы преобразователя уровней сигналов RS232-TTL. Как видно из этих рисунков, разводка трасс визуально стала более плотной, а суммарная длина трасс уменьшилась на 14 %, что является неплохим показателем.

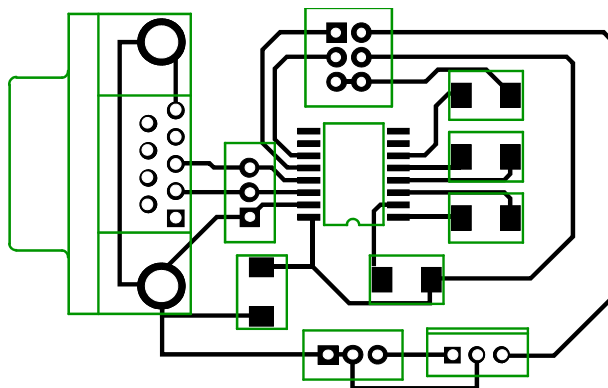


Рис. 3. Исходный вариант трассировки, экспортированный из САПР DipTrace

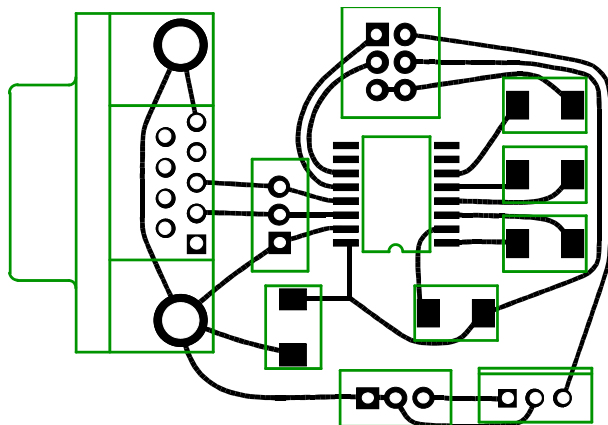


Рис. 4. Результирующая трассировка

Также стоит обратить внимание на тот факт, что поддержание заданного зазора между проводниками выполняется автоматически. За счет того, что элементы различных цепей испытывают взаимные силы отталкивания, полностью исключается не только их пересечение, но и приближение друг к другу сверх указанного расстояния.

Заключение

Укажем преимущества рассмотренного алгоритма:

1. Поскольку предлагаемый алгоритм относится к классу параллельных алгоритмов трассировки и применяется ко всем трассам схемы одновременно, потенциально получается лучший результат, чем исходный, выполненный с применением одного из стандартных алгоритмов, используемых в современных САПР (DipTrace, Altium Designer и пр.). В данной статье рассматривался типовой пример, поставляемый с САПР DipTrace. На других примерах возможны другие показатели улучшения качества трассировки.

2. Возможна модификация алгоритма, при которой под действием сил будут перемещаться не только трассы и переходные отверстия, но и сами элементы схемы, что позволит еще более сократить суммарную длину трасс и общую площадь печатной платы (за счет более компактного расположения элементов и трасс), а также учесть ограничения на взаимное расположение элементов схемы.

Список литературы

1. Ланин В. Л., Хмыль А. А. Технология производства электронных средств : учеб. пособие. Минск : Вышэйшая школа, 2019. 455 с.
2. Ларистов А. И., Лячек Ю. Т., Мусаид А. М. О., Мустафа Ахмед Б. М. Сравнение систем трассировки печатных плат // Международная конференция по мягким вычислениям и измерениям (SCM'2017). 2017. Т. 2. С. 101–105.
3. Морозов К. К., Одинокое В. Г., Курейчик В. М. Автоматизированное проектирование конструкций радиоэлектронной аппаратуры. М. : Радио и связь, 1983. 280 с.
4. Деньдобренко Б. Н., Малика А. С. Автоматизация конструирования РЭА : учебник для вузов. М. : Высшая школа, 1980. 384 с.
5. Lee C. Y. An Algorithm for Path Connections and Its Applications // IRE Transactions on Electronic Computers. 1961. Vol. EC-10, no. 3. P. 346–365.
6. Sherwani N. Algorithms for VLSI Physical Design Automation. 3d ed. New York ; Boston ; Dordrecht ; London ; Moscow : Kluwer Academic Publishers, 1998. 603 p.
7. Raith M., Bartholomeus M. A new hypergraph-based rip-up and reroute strategy // 28th ACM/IEEE Design Automation Conference (DAC '91) : proceedings. New York : Association for Computing Machinery, 1991. P. 54–59.
8. De V. K. A Heuristic Global Router for Polycell Layout : PhD thesis. Durham, North Carolina : Duke University, 1986.
9. Mikami K., Tabuchi K. A computer program for optimal routing of printed circuit connectors // IFIPS Proc., H47. 1968. P. 1475–1478.
10. Hightower D. W. A solution to line-routing problem on the continuous plane // 6-th Design Automation Workshop : proceedings. 1969. P. 12–24.
11. Garey M. R., Johnson D. S. The rectilinear Steiner tree problem is NP-complete // SIAM Journal Applied Mathematics. 1977. Vol. 32. P. 826–834.

12. De Souza C. C., Ribeiro C. C. Heuristics for the minimum rectilinear Steiner tree problem: new algorithms and a computational study // *Discrete Applied Mathematics*. 1993. Vol. 45. P. 205–220.
13. Бутенин Н. В., Лунц Я. Л., Меркин Д. Р. Курс теоретической механики : учеб. пособие для ВО : в 2 т. 12-е изд., стер. СПб. : Лань, 2020. 732 с.

References

1. Lanin V.L., Khmyl' A.A. *Tekhnologiya proizvodstva elektronnykh sredstv: ucheb. posobie = Technology of electronic means production : textbook*. Minsk: Vysheyshaya shkola, 2019:455. (In Russ.)
2. Laristov A.I., Lyachek Yu.T., Musaid A.M.O., Mustafa Akhmed B.M. Comparison of PCB tracing systems. *Mezhdunarodnaya konferentsiya po myagkim vychisleniyam i izmereniyam (SCM'2017) = International Conference on Soft Computing and Measurements (SCM'2017)*. 2017;2:101–105. (In Russ.)
3. Morozov K.K., Odinkov V.G., Kureychik V.M. *Avtomatizirovannoe proektirovanie konstruktivnykh radioelektronnoy apparatury = Automated design of radio-electronic equipment structures*. Moscow: Radio i svyaz', 1983:280. (In Russ.)
4. Den'dobrenko B.N., Malika A.S. *Avtomatizatsiya konstruirovaniya REA: uchebnik dlya vuzov = Automation of REA design : textbook for universities*. Moscow: Vysshaya shkola, 1980:384. (In Russ.)
5. Lee C.Y. An Algorithm for Path Connections and Its Applications. *IRE Transactions on Electronic Computers*. 1961;EC-10(3):346–365.
6. Sherwani N. *Algorithms for VLSI Physical Design Automation*. 3d ed. New York; Boston; Dordrecht; London; Moscow: Kluwer Academic Publishers, 1998:603.
7. Raith M., Bartholomeus M. A new hypergraph-based rip-up and reroute strategy. *28th ACM/IEEE Design Automation Conference (DAC '91): proceedings*. New York: Association for Computing Machinery, 1991:54–59.
8. De V.K. *A Heuristic Global Router for Polycell Layout: PhD thesis*. Durham, North Carolina: Duke University, 1986.
9. Mikami K., Tabuchi K. A computer program for optimal routing of printed circuit connectors. *IFIPS Proc., H47*. 1968:1475–1478.
10. Hightower D.W. A solution to line-routing problem on the continuous plane. *6-th Design Automation Workshop: proceedings*. 1969:12–24.
11. Garey M.R., Johnson D.S. The rectilinear Steiner tree problem is NP-complete. *SIAM Journal Applied Mathematics*. 1977;32:826–834.
12. De Souza C.C., Ribeiro C.C. Heuristics for the minimum rectilinear Steiner tree problem: new algorithms and a computational study. *Discrete Applied Mathematics*. 1993;45:205–220.
13. Butenin N.V., Lunts Ya.L., Merkin D.R. *Kurs teoreticheskoy mekhaniki: ucheb. posobie dlya VO: v 2 t. 12-e izd., ster. = Course of theoretical mechanics: textbook. manual for military education : in 2 vols. 12th ed., ster.*. Saint Petersburg: Lan', 2020:732. (In Russ.)

Информация об авторах / Information about the authors

Александр Моисеевич Бершадский
доктор технических наук, профессор,
заведующий кафедрой систем
автоматизированного проектирования,
Пензенский государственный университет
(Россия, г. Пенза, ул. Красная, 40)
E-mail: bam@pnzgu.ru

Alexander M. Bershadsky
Doctor of technical sciences, professor,
head of the sub-departments
of computer-aided design systems,
Penza State University
(40 Krasnaya street, Penza, Russia)

Павел Анатольевич Гудков

кандидат технических наук, доцент,
доцент кафедры систем
автоматизированного проектирования,
Пензенский государственный университет
(Россия, г. Пенза, ул. Красная, 40)
E-mail: p.a.gudkov@gmail.com

Pavel A. Gudkov

candidate of technical sciences,
associate professor, associate professor
of the sub-departments
of computer-aided design systems,
Penza State University
(40 Krasnaya street, Penza, Russia)

Екатерина Михайловна Подмарькова

кандидат технических наук, доцент,
доцент кафедры систем
автоматизированного проектирования,
Пензенский государственный университет
(Россия, г. Пенза, ул. Красная, 40)
E-mail: kalibryk.m@gmail.com

Ekaterina M. Podmar'kova

candidate of technical sciences,
associate professor, associate professor
of the sub-departments
of computer-aided design systems,
Penza State University
(40 Krasnaya street, Penza, Russia)

Авторы заявляют об отсутствии конфликта интересов /

The authors declare no conflicts of interests.

Поступила в редакцию/Received 31.05.2021

Поступила после рецензирования/Revised 24.06.2021

Принята к публикации/Accepted 12.07.2021