

Zargaryan Elena Valerevna – e-mail: evzargaryan@sfnedu.ru; the department of automatic control systems; cand. of eng. sc.; associate professor.

Koshensky Vadim Igorevich – e-mail: koshenskii@sfnedu.ru; the department of automatic control systems; bachelor 4 years of study.

Kirsanov Kirill Olegovich – e-mail: kkirsanov@sfnedu.ru; the department of automatic control systems; bachelor 4 years of study.

УДК 007.52:629.3.05

DOI 10.18522/2311-3103-2023-5-34-48

В.А. Костюков, М.Ю. Медведев, В.Х. Пшихопов**АЛГОРИТМ ПЛАНИРОВАНИЯ ПУТИ В ДВУХМЕРНОЙ СРЕДЕ
С ПОЛИГОНАЛЬНЫМИ ПРЕПЯТСТВИЯМИ НА КЛАССЕ
КУСОЧНО-ЛОМАНЫХ ТРАЕКТОРИЙ***

Актуальной проблемой, возникающей при разработке алгоритмов автоматического планирования пути, является рост вычислительных затрат при увеличении сложности среды функционирования. Не лишены этого недостатка графовые методы планирования, в частности, метод диаграмм видимости. Он позволяет сформировать в качестве узлов графа вершины полигональных границ каждого препятствия, а ребрами графа являются все те отрезки, соединяющие эти вершины, которые не имеют пересечений с препятствиями. При увеличении количества препятствий возрастает сложность такого графа, причем этот рост очень быстрый. Поэтому наиболее важной задачей становятся приемы сокращения сложности графа видимости. В данной статье предлагается гибридный алгоритм, строящийся на методе диаграмм видимости и семействе Виг-алгоритмов. Виг-алгоритм относится к классу локальных, поскольку каждый раз имеет дело с огибанием одного препятствия, появляющегося на пути следования робота, и этот алгоритм не может предсказать заранее, какое следующее препятствие придется обходить. Предлагаемый в данной статье метод планирования траектории движения сочетает графовый алгоритм с Виг-алгоритмами, что позволяет построить специальный граф с узлами в виде характерных точек препятствий. При этом Виг-алгоритм является шагом итерационного процесса оптимизации на графе, позволяющего за конечное число шагов прийти к оптимальному решению на классе кусочно-ломанных кривых. Предлагаемый метод решает задачу глобального поиска пути на классе кусочно-линейных траекторий с полигональными препятствиями; а в отличие от классических методов диаграмм прямой видимости, существенно снижает размерность графа за счет специального выбора ограниченного количества характерных точек соответствующих препятствий. В статье проводится разработка и теоретическое обоснование предлагаемого метода. Приводятся расчетные соотношения алгоритма, обосновывается оптимальность получаемой траектории. Аналитические соотношения подтверждаются результатами численного моделирования в различных средах, заполненных полигональными препятствиями. При этом эффективность предлагаемых алгоритмов подтверждается на примерах среды, заполненной препятствиями до 70–80%. Показано, что для прокладывания пути на сценах лабиринтного типа с одним распространенным видом препятствий рассматриваемый алгоритм на 10% превосходит оптимальный алгоритм Дейкстры.

Планирование пути; сложные полигональные препятствия; поиск на графах; Виг-алгоритм; гибридный алгоритм планирования.

* Исследование выполнено за счет гранта Российского научного фонда № 22-29-00337 «Разработка вероятностных методов оптимизации траекторий группы подвижных робототехнических комплексов в априори неопределенной среде», <https://rscf.ru/project/22-29-00337/> на базе АО «НКБ Робототехники и систем управления».

V.A. Kostyukov, M.Yu. Medvedev, V.Kh. Pshikhopov

**AN ALGORITHM FOR PATH PLANNING IN A TWO-DIMENSIONAL
ENVIRONMENT WITH POLYGONAL OBSTACLES ON A CLASS
OF PIECEWISE POLYLINE TRAJECTORIES**

An urgent problem of developing algorithms for automatic path planning is the increase in computational costs with an increase in the complexity of the operating environment. Graphic planning methods, in particular the method of visibility diagrams, are not without this drawback. As the number of obstacles increases, the complexity of a visibility graph increases, and this growth is very fast. Therefore, techniques for reducing the complexity of the visibility graph become the most important task. This article proposes a hybrid algorithm based on the method of visibility diagrams and Bug algorithms. The method of path planning proposed in this article combines a graph algorithm with Bug algorithms, which allows you to build a special graph with nodes in the form of characteristic obstacle points. In this case, the Bug algorithm is a step in the iterative optimization process on the graph, which allows for a finite number of steps to come to an optimal solution on a class of piecewise polyline curves. The proposed method solves the problem of global path search. Unlike the classical methods of line-of-sight diagrams, it significantly reduces the dimension of the graph due to a special selection of a limited number of characteristic points of the corresponding obstacles. The article develops and theoretically substantiates the proposed method. The calculated relations of the algorithm are given. The optimality of the obtained trajectory is justified. The analytical relations are confirmed by the results of numerical modeling in various environments filled with polygonal obstacles. It is shown that in order to pave the way for maze-type scenes with one common type of obstacles, the algorithm under consideration is 10% superior to the optimal Dijkstra algorithm.

Path planning; complex polygonal obstacles; graph search; Bug algorithm; hybrid planning algorithm.

Введение. Большинство существующих алгоритмов планирования пути сталкиваются с проблемой роста вычислительной сложности при увеличении сложности карты препятствий. В связи с этим актуальной является задача разработки алгоритма планирования пути, у которых сложность растет медленнее, чем у существующих алгоритмов.

Наиболее распространенные методы планирования пути используют алгоритмы поиска на графах. Каждый узел графа представляет одно состояние робота, которое может представлять собой положение, угол ориентации, скорость или ускорение робота. Переходы между состояниями отражаются ребрами графа и характеризуются функцией затрат. Это потенциально позволяет найти путь, который имеет минимальную общую стоимость достижения до целевого состояния. Для построения узлов графа может быть использован метод диаграмм видимости, который позволяет сформировать в качестве узлов вершины полигональных границ каждого препятствия, а ребрами графа являются отрезки, соединяющие эти вершины, которые не имеют пересечений с препятствиями.

При увеличении количества препятствий возрастает сложность такого графа, причем этот рост очень быстрый. Поэтому здесь наиболее важной задачей становятся приемы сокращения сложности графа видимости. Huang [1] предложил метод динамической диаграммы видимости, а Janet ввел концепцию т-вектора [2].

Другими подходами являются метод структурирования свободного пространства [3] и метод на основе диаграммы Вороного [4]. Здесь выделяются так называемые свободные звенья – отрезки, соединяющие вершины препятствий и не пересекающиеся с ними. Из свободных звеньев собираются выпуклые многогран-

ники, описывающие области свободного пространства. Эти данные оформляются в виде специального графа, а спланированный путь в обход препятствий строится алгоритмами поиска путей на графе. Использование диаграмм Вороного также приводит к построению допустимого пути, максимально удаленного от препятствий, при этом обеспечивает дополнительные возможности с точки зрения позиционирования робота. Методы структурирования свободного пространства и диаграммы Вороного дают более безопасный путь с точки зрения столкновения с препятствиями, но общая длина пути увеличивается.

К методам на основе случайного выбора относится метод быстро исследующих случайных деревьев (RRT) [5]. В этом методе процесс генерации пути состоит в построении дерева опорных точек, которое последовательно расширяется от начальной точки до целевой точки. Данный метод обладает невысокой сходимостью к оптимальному пути, поэтому направления его развития связаны с методами повышения его эффективности [6, 7].

Графовые методы поиска также используются в методах клеточной декомпозиции, предполагающих разбиение исходного пространства на дискретные ячейки [8]. Также получили распространение методы адаптивной пространственной декомпозиции [9, 10]. Пространственная декомпозиция позволяет применять методы дискретного поиска, базирующиеся на теории графов; наибольшую популярность получили алгоритмы семейств A^* и D^* [11, 12].

Алгоритм A^* является развитием алгоритма Дейкстры, в котором за счет выбора эвристической функции стоимости пути снижается вычислительная сложность. Вместе с тем, алгоритм требователен к памяти и его вычислительная сложность резко растет с ростом числа ячеек разбиения. Для снижения требовательности исходного алгоритма A^* к памяти и его вычислительной сложности разработаны различные модификации [13, 14]. Также заметим, что алгоритм Дейкстры позволяет прийти оптимальному решению с точностью до погрешности квантования, в отличие от методов A^* и D^* , чьи эвристики в некоторых случаях отклоняют поиск от оптимального решения.

Также существует семейство Bug-алгоритмов базирующихся на обходе препятствий, которые встретились на пути отрезка «прямой видимости», соединяющего начальную и конечную точки [15, 16]. При этом обход происходит по границам этих препятствий, до некоторой точки «отрыва», начиная с которой траектория уходит с указанной границы. Bug-алгоритм относится к классу локальных, поскольку каждый раз имеет дело с обходом одного препятствия, появляющегося на пути следования робота, и этот алгоритм не может предсказать заранее, какое следующее препятствие придется обходить [17].

Цель настоящей статьи заключается в разработке нового гибридного метода планирования траекторий, в котором bug-алгоритм является шагом итерационного процесса оптимизации на графе, что позволяет за конечное число шагов прийти к оптимальному решению на классе кусочно-ломаных кривых. Таким образом, предлагаемый аналитический метод планирования объединяет поиск на графах с итерационным Bug-алгоритмом, что позволяет построить специальный граф с узлами в виде характерных точек препятствий. Эти узлы соединяются кусочно-линейными отрезками. Затем решается задача поиска кратчайшего пути на построенном графе. Новизна алгоритма заключается в итерационной процедуре анализа соответствий между отрезком «прямой видимости», соединяющим текущую точку с целевой, и криволинейными полигональными границами препятствий, пересекаемых им.

В отличие от исходных Bug-алгоритмов, предлагаемый метод решает задачу глобального поиска пути на классе кусочно-линейных траекторий с полигональными препятствиями. В отличие от классического методов диаграмм прямой видимости, существенно снижает размерность графа за счет специального выбора ограниченного количества характерных точек соответствующих препятствий.

I. Постановка задачи. Рассматривается двумерная односвязная область $D \subset \mathbb{R}_2$, в которой заданы полигональные препятствия, множество которых обозначим $\{Obs\}$. Обозначим точки старта и финиша как A и B соответственно.

Под точечной границей $\partial(Obs)$ некоторого полигонального препятствия Obs, заданного как двумерное многообразие в $\mathbb{R}_2$, или просто его границей, будем понимать упорядоченное множество вершин замкнутой кусочно-ломаной – реальной границы данного препятствия. Последнюю обозначим так же, что будем оговаривать всякий раз отдельно. Рассматриваются всевозможные кусочно-линейные траектории $\{Tr\}$, целиком принадлежащие области D. Каждая такая траектория Tr определяется как совокупность ее вершин, которые обозначаются в квадратных скобках $Tr=[Вершина1, Вершина2, ..]$. Тем же названием Tr обозначим саму кусочно-ломаную как одномерное многообразие. Каждый раз далее всякий раз оговаривается, какая именно интерпретация траектории имеется в виду. Если некоторая траектория Tr в качестве граничных имеет точки A и B, то назовем ее траекторией глобального пути из A в B. Множество таких траекторий обозначим $G_{AB}(Tr) \subset \{Tr\}$.

Ставится задача разработать и обосновать метод, решающий задачу глобального поиска пути на классе кусочно-линейных траекторий с полигональными препятствиями. Данный метод должен обеспечивать снижение размерности графа по сравнению с классическим методом диаграмм прямой видимости.

II. Алгоритм планирования. Предлагаемый алгоритм является итерационным. Далее приводится подробное описание указанного алгоритма.

Вначале итерационного алгоритма проводится отрезок прямой видимости $L_0=AB$, причем случай отсутствия пересечения $L_0 \cap Obs = \emptyset$ не рассматривается как тривиальный. Находится такое препятствие с номером j , что оно пересекается с прямой L_0 и $d_{min,j} = \min_i(d_{min,i})$, где $d_{min,i}$ – расстояние от начальной точки A до наиболее близкой к ней точки пересечения M_{i1} прямой L_0 и i -го препятствия. Обозначим такую точку, соответствующую j -му препятствию, как M_{j1} . Кроме того, выделяется точка M_{j2} пересечения препятствия Obs_j с прямой L_0 , наиболее удаленная от точки A из всех точек пересечения. Далее строятся два пути Tr_1 и Tr_2 : $Tr_1 = [A, M_{j1}, \partial_1(Obs_j)]$, $Tr_2 = [A, M_{j1}, \partial_2(Obs_j)]$. Точки A и M_{j2} являются граничными точками этих траекторий. Здесь $\partial_1(Obs_j)$ и $\partial_2(Obs_j)$ есть связные подмножества границы $\partial(Obs_j)$ с непрерывной нумерацией элементов и такие, что $\partial_1(Obs_j) \cup \partial_2(Obs_j) = \partial(Obs_j)$, $\partial_1(Obs_j) \cap \partial_2(Obs_j) = \{M_{j1}, M_{j2}\}$.

Далее пути Tr_1 и Tr_2 пути оптимизируются: для первого находится такое упорядоченное подмножество точек $Tr_1^* \subseteq Tr_1$, что для соответствующей кусочно-ломаной выполняются условия: а) она начинается с точки A; б) не пересекается ни с одним из препятствий сцены, отличным от Obs_j в) пересекая границу $\partial(Obs_j)$, не пересекает внутреннюю часть Obs_j ; г) имеет минимальную длину при условии, что последняя точка $Tr_1^*(end)$ множества Tr_1^* позволяет провести прямолинейный отрезок $Tr_1^*(end)B$, не имеющий пересечений с препятствием Obs_j . Аналогичным образом оптимизируется второй частичный путь Tr_2 с получением Tr_2^* . Заметим,

что оптимизированные частичные пути уже могут не включать точки пересечения M_{j1} или/и M_{j2} . При этом используется специальный метод прореживания кусочно-ломаной, рассмотренный ниже.

На рис. 1 приведен пример построения исходных Tr_1 , Tr_2 и оптимизированных путей Tr_1^* и Tr_2^* для сцены из трех препятствий. Здесь $Tr_1 = [A, M_{j1}, A_{j1}, A_{j2}, M_{j2}]$, $Tr_2 = [M_{j1}, A_{j4}, A_{j3}, M_{j2}]$, $Tr_1^* = [A, A_{j1}, A_{j2}]$, $Tr_2^* = [A, A_{j4}]$.

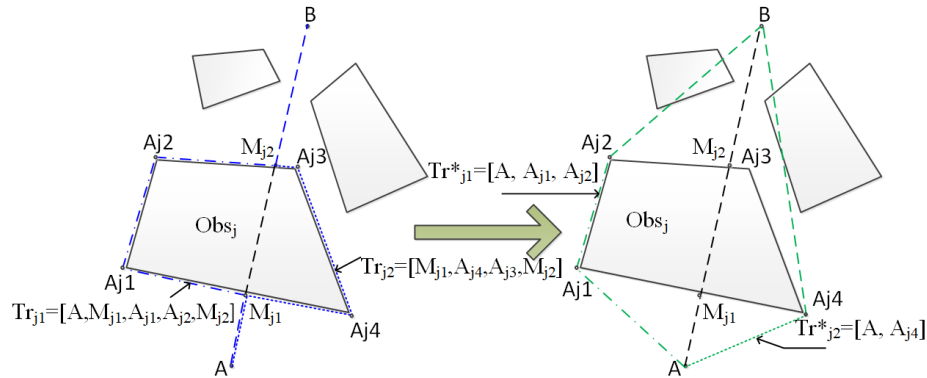


Рис. 1. Построение частичных путей на 1-й итерации

На рис. 2 показано построение частичных оптимизированных путей на двух итерациях алгоритма планирования.

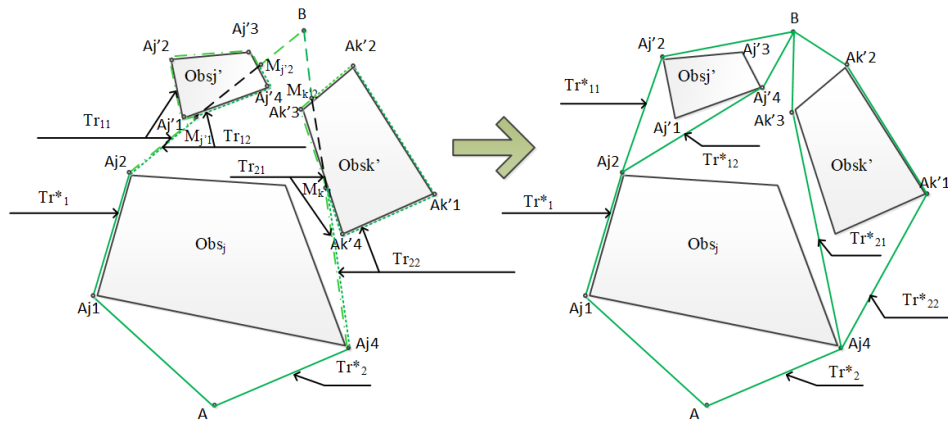


Рис. 2. Построение частичных путей на 2-й итерации

После первой итерации возникает ветвление траекторий, соответствующее двум найденным частичным путям Tr_1^* и Tr_2^* .

Введем понятие глобального пути, соответствующего конечной точке P некоторого частичного оптимизированного пути, достигнутого на u -й итерации. Оно обозначает объединение всех частичных оптимизированных путей до этой точки и отрезка PB . Каждый такой путь обозначим как $GTr_p^{(u)}$. Полную длину всех частичных оптимизированных путей до точки P обозначим как $L(P)$, тогда длина $GTr_p^{(u)}$ есть $|GTr_p^{(u)}| = L(P) + |PB|$.

Будем считать, что соответствующие глобальные пути после первой итерации $u = 1$ есть $GTr_1^{(1)} = [Tr_1^*, B]$ и $GTr_2^{(1)} = [Tr_2^*, B]$. Если отрезок $L_1 = Tr_1^*(end)B$ не пересекается ни с одним из препятствий, то $GTr_1^{(1)}$ – путь, который мы отнесем во множество $GTr^{(1)}$ отобранных глобальных путей на 1-й итерации. Точно так же поступим с $GTr_2^{(1)}$, если только отрезок $L_1 = Tr_2^*(end)B$ не пересекает ни одно из препятствий сцены. При этом, если множество $GTr^{(1)}$ имеет два элемента, то минимальный по длине путь из двух и дает искомое решение. Иначе, если мощность множества не более 1, то выбирается тот из глобальных путей, который дает пересечение с $\{Obs\}$. Предположим, что $|GTr^{(1)}| = 0$ и рассмотрим ветвление обоих глобальных путей.

Пусть $Obs_{j'}$ есть препятствие, с которым пересекается отрезок L_1 . Заметим, что по построению L_1 не может пересечься с Obs_j , поэтому $j' \neq j$. Теперь повторим рассуждения, которые мы применили ранее к точке A и отрезку L_0 , но теперь – в отношении точки $Tr_1^*(end)$ и отрезка L_1 . Находим частичные пути $Tr_{11} = [Tr_1^*(end), M_{j'1}, \partial_1(Obs_{j'})]$ и $Tr_{12} = [Tr_1^*(end), M_{j'1}, \partial_2(Obs_{j'})]$, где $Tr_1^*(end)$ и $M_{j'2}$ есть крайние, ближайшая и дальняя точки пересечения прямолинейного отрезка L_1 с препятствием $Obs_{j'}$; $\partial_1(Obs_{j'})$ и $\partial_2(Obs_{j'})$ есть соответствующие непрерывные по нумерации своих элементов связанные подмножества точечной границы $\partial(Obs_{j'})$, удовлетворяющие условиям: $\partial_1(Obs_{j'}) \cup \partial_2(Obs_{j'}) = \partial(Obs_{j'})$, $\partial_1(Obs_{j'}) \cap \partial_2(Obs_{j'}) = \{M_{j'1}, M_{j'2}\}$. Далее строится оптимизированный частичный путь Tr_{11}^* , удовлетворяющий условиям, аналогичным указанным в пунктах а) – г) для Tr_1^* . Здесь и далее обозначение «end», фигурирующее в круглых скобках при некоторых, в общем случае, частичных, траекториях типа Tr_{11} , Tr_1^* , Tr_{12} , и т.п., означает число узлов соответствующей траектории.

Теперь можно объединить Tr_1^* и Tr_{11}^* с получением первого глобального пути на второй итерации:

$$GTr_1^{(2)} = [Tr_1^*(1: end - 1), Tr_{11}^*, B]. \quad (1)$$

Точно так же оптимизируем второй частичный путь с получением множества точек Tr_{12}^* , после чего можно построить начало второго глобального пути на второй итерации:

$$GTr_2^{(2)} = [Tr_1^*(1: end - 1), Tr_{12}^*, B]. \quad (2)$$

Далее берем частичный путь Tr_2^* , полученный на первой итерации, и находим для него частичные пути, отвечающие ветвлению в конечной точке $Tr_2^*(end)$ первой итерации. Вновь исключаем тривиальный случай, и находим ближайшее к этой точке препятствие по направлению отрезка $Tr_2^*(end)B$ – пусть $Obs_{k'}$ с точками пересечения $M_{k'1}, M_{k'2}$, тогда получим частичные огибающие траектории $Tr_{21} = [Tr_2^*(end), M_{k'1}, \partial_1(Obs_{k'})]$ и $Tr_{22} = [Tr_2^*(end), M_{k'1}, \partial_2(Obs_{k'})]$.

Соответствующие глобальные пути, порожденные рассматриваемой процедурой ветвления в точке $Tr_2^*(end)$, есть:

$$GTr_3^{(2)} = [Tr_2^*(1: end - 1), Tr_{21}^*], \quad GTr_4^{(2)} = [Tr_2^*(1: end - 1), Tr_{22}^*]. \quad (3)$$

Очевидно, что мощность множества отобранных глобальных путей на второй итерации может быть не более четырех.

На рис. 2 приведены построенные частичные, исходные и оптимизированные, пути для рассматриваемого примера на 2-й итерации: Tr_{11} , Tr_{12} , Tr_{21} , Tr_{22} и Tr_{11}^* , Tr_{12}^* , Tr_{21}^* , Tr_{22}^* .

Рассмотренная процедура ветвления приводит к постоянному зарождению новых путей, причем ветвление в данной точке какого-либо частичного пути на текущей u -й итерации прекращается, если эта точка совпала с целевой B . Такой путь считается глобальным завершённым.

Для структурного описания и анализа получающегося итерационного процесса может быть построен ориентированный граф, в узлах которого располагаются концевые точки каждого частичного оптимизированного пути; его ребра обозначают оптимизированные кусочно-линейные пути, соединяющие соответствующие узлы, причем длины таких путей будем считать весами обозначаящих их ребер. Например, на рис. 3 приведен такой граф для рассмотренного выше примера; здесь, в частности, A'_{11} есть точка A_{j2} , а также выполнены следующие соответствия (см. рис. 1): $A'_{12} = A_{j4}$, $A'_{111} = A_{j'2}$, $A'_{112} = A_{j'4}$, $A'_{121} = A_{k'3}$, $A'_{122} = A_{k'2}$.

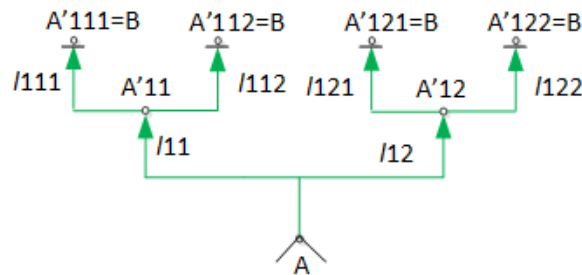


Рис. 3. Граф для случая без замыканий

Численный анализ разработанного алгоритма показал, что возможны случаи, когда одна или более пар разных узлов такого графа совпадают на некоторой итерации. В этом случае итерационный процесс приводит к пересечению двух или более разных путей, в некоторой одной характерной точке какого-либо препятствия. Например, на рис. 4 рассмотрен более сложный итерационный процесс, мощность множества всех его глобальных путей, без учета динамического отсеивания (см. далее), составляет 10. При этом узел A'_{12211} совпадает с A'_{1111} , аналогично, A'_{1121} - с A'_{122} . При каждом таком совпадении, по сути, происходит поглощение либо исключение определенного подмножества глобальных путей на текущей итерации по следующему правилу. Пусть узел $\bar{A}'$ совпал с узлом $\hat{A}'$ на u -й итерации. Для определенности будем считать, что узел $\hat{A}'$ получен на некоторой предшествующей итерации $\hat{u} < u$. Пусть $GTr_{\bar{A}'}^{(u)}$ и $GTr_{\hat{A}'}^{(\hat{u})}$ есть глобальные пути этих двух узлов, полученные на u -й и $\hat{u}$ -й итерациях, соответственно. Без учета замыкающих прямых отрезков $\bar{A}'B$ и $\hat{A}'B$ длины этих путей есть $L(\bar{A}')$ и $L(\hat{A}')$, соответственно. Если выполнено равенство $\hat{A}' = \bar{A}'$ и условие

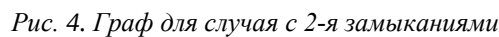
$$L(\bar{A}') \leq L(\hat{A}'), \quad (4)$$

тогда необходимо заменить все глобальные пути, порожденные точкой $\hat{A}'$ и полученные на всех итерациях от $\hat{u} + 1$ до u , на новые по правилу:

$$GTr_{\hat{A}'}^{(u)} \rightarrow (GTr_{\hat{A}'}^{(u)} \setminus \hat{A}'B) \cup [GTr_{\hat{A}'}^{(u)} \setminus (GTr_{\hat{A}'}^{(\hat{u})} \setminus \hat{A}'B)], \quad (5)$$

означающему, что одинаковые части $GTr_{\hat{A}'}^{(\hat{u})} \setminus \hat{A}'B$ всех таких путей до точки $\hat{A}'$ заменяются на кусочно-ломаную $GTr_{\hat{A}'}^{(u)} \setminus \hat{A}'B$, порожденную узлом $\bar{A}'$. Если же указанное условие не выполняется, т.е. $L(\bar{A}') > L(\hat{A}')$, то, наоборот, глобальный путь,

Граф для любого итерационного процесса, приводящего к совпадению двух или более пар (троек, четверок и т.д.) узлов, имеет дополнительные циклы, порожденные указанными перекрестными связями вершин, порождаемых разными родительскими вершинами. Для рассмотренного на рис. 4 случая граф имеет два таких дополнительных цикла.



Заметим, что нет необходимости доводить всякий раз все ветви графа до логического завершения – достижения целевой точки. Действительно, рассмотрим способ отсеивания глобальных путей, значительно увеличивающий быстродействие алгоритма. Как только процедура ветвления на некотором своем u -м шаге получает набор из одного или нескольких завершенных путей, тогда среди них вы-

бирается путь с минимальной длиной и фиксируется также эта длина как $L_{curr}(u)$. Тогда на следующем $u+1$ -м шаге производится сравнение приведенной длины каждого l -го глобального пути с величиной $L_{curr}(u)$ с отбрасыванием всех тех глобальных путей, которые получили превышение по длине над $L_{curr}(u)$. А именно, для всех $l = 1, 2, \dots, l_{max}(u+1)$ проверяются условия:

$$L(P_l) + |P_l B| = |GTr_n^{(u+1)}| \geq L_{curr}(u), \quad (6)$$

где P_l есть предпоследняя точка глобального пути $GTr_l^{(u+1)}$, полученного на u -й итерации. Напомним, что конечной точкой для любого глобального пути всегда является целевая. Здесь $l_{max}(u+1)$ есть число всех глобальных путей, полученных на $u+1$ -й итерации и не отсеянных ранее.

При выполнении для некоторого l неравенства (6) далее l -й глобальный путь не должен производить ветвлений, поскольку каждый из производных путей, полученных экстраполяцией ветвления l -го пути вплоть до их завершения, будет иметь длину, не меньшую чем левая часть неравенства (6).

Предположим, что вследствие итерационного процесса найден путь $GTr_{n_{opt}}^{(u_{max})}$ с минимальной длиной на множестве всех глобальных путей:

$$|GTr_{n_{opt}}^{(u_{max})}| = \min_{1 \leq l \leq l_{max}(u_{max})} |GTr_{n(l, u_{max})}^{(u_{max})}|, \quad n_{opt} = n(l_{opt}, u_{max}). \quad (7)$$

Рассмотренную выше последовательность действий, приводящую к построению $GTr_{n_{opt}}^{(u_{max})}$, назовем алгоритм построения глобального пути первого приближения, или первым этапом алгоритма планирования.

Второй этап подразумевает корректировку описанного выше алгоритма. Покажем, что отсеиваемые по критериям (4) и (6) глобальные пути могут быть в общем случае оптимизированы по длине, и поэтому получение результирующего пути алгоритма сопряжено с учетом не только оптимального по длине пути $GTr_{n_{opt}}^{(u_{max})}$ на множестве $GTr_{n(l, u_{max})}^{(u_{max})}$, но и, в общем случае, всех путей этого множества.

Действительно, при обходе некоторого препятствия П1 по принципу построения виртуальной прямой видимости может возникнуть ситуация, когда в некоторой окрестности этого препятствия могут быть другие, которые увеличивают длину участка кусочно-ломаной, идущей до характерной точки препятствия П1. Подобная ситуация показана на рис. 3. Препятствие П2 не допускает проложить локальный маршрут AA_3 , и для обхода П2 первоначально строится более длинный локальный путь AA_2A_3 . Вместе с тем, наличие препятствия П2 позволяет решить локальную задачу его обхода при движении от точки A к A_3 , с тем, чтобы сократить длину этого участка пути. В результате получатся два пути $AA_5A_6A_3$ и AA_7A_3 длины которых меньше, чем у пути AA_2A_3 . Заметим, что препятствий, подобных П2 по влиянию на возможность быстрого достижения точки A_5 , может быть более одного. В этом случае целесообразно применить последовательность действий первого этапа алгоритма, рассмотренную выше, ко всем тройкам узлов построенной на этом этапе кусочно-ломаной $Tr(1:N) = [A_1, A_2, \dots, A_N]$ с числом узлов N .

Суть второго этапа планирования заключается в следующем. Для каждой тройки узлов $Tr_k(1:3) \equiv [A_k, A_{k+1}, A_{k+2}]$, $k=1: N-2$, в цикле по k применяется рассмотренный выше алгоритм первого этапа, причем точка A_k считается начальной, а A_{k+2} – конечной. Если длина $|Tr'_k|$ полученной кусочно-ломаной $Tr'_k(1:N_k) \equiv [A_k, A'_{k1}, A'_{k2}, \dots, A'_{kN_k-1}, A_{k+2}]$, $N_k \geq 2$, меньше длины $|Tr_k|$, то производится замена участка $[A_k, A_{k+1}, A_{k+2}]$ на $Tr'_k(1:N_k)$. При этом производится инкрементация $k:=k+2$ и действия в цикле повторяются для новой тройки с начальной точ-

III. результаты численного исследования. Результаты численного моделирования представлены на рис. 6 и 7. При моделировании использован программный комплекс, разработанный ранее для исследования различных алгоритмов планирования и управления движением [18, 19].

На рис. 6 и 7 представлены результаты расчета целевых траекторий для двух сцен с помощью двух алгоритмов: рассматриваемого и Дейкстры. Каждая сцена здесь представляет собой сложный лабиринт с препятствиями, каждое из которых есть связанное объединение некоторой совокупности ячеек прямоугольной равномерной сетки. Для первой сцены длина траектории, построенной по рассматриваемому алгоритму и показанной сплошной кривой, составляет $L_A = 1.535e+03$ условных единиц длины, при времени расчета $T_A = 5,3$ с, а для траектории, построенной по алгоритму Дейкстры и показанной пунктирной кривой, – $L_D = 1.564741996484947e+03$ при времени расчета $T_D = 14,4$ с. Во втором случае $L_A = 1.446e+03$ при времени расчета $T_A = 7,5$ с, $L_D = 1.467e+03$ при времени расчета $T_D = 8,7$ с.

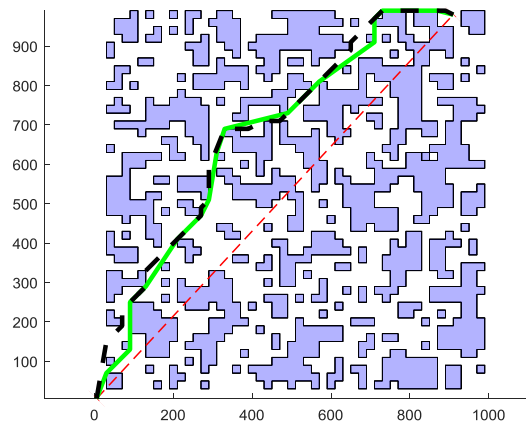


Рис. 6. Результаты моделирования на случайно сгенерированной сцене 1

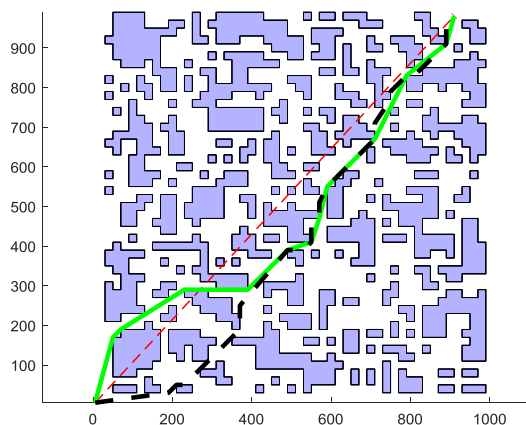


Рис. 7. Результаты моделирования на случайно сгенерированной сцене 2

На рис. 6 и 7 сплошной линией показаны результаты моделирования разработанного в данной статье алгоритма, а пунктирной линией – результаты моделирования алгоритма Дейкстры.

Проведенные численные исследования на различных сценах, при случайном генерируемых картах и положениях начальной и целевой точек показали, что разработанный аналитический алгоритм выигрывает у алгоритма Дейкстры, как по длине найденного пути (в среднем, на 4%), так и по времени расчета (в среднем, на 10%).

Численный анализ показал наличие следующей тенденции: с ростом сложности препятствий рассматриваемый алгоритм выигрывает по быстродействию у алгоритмов клеточной декомпозиции на один-два порядка и более, потому что достаточно точный учет формы этих препятствий требует соответствующего детального клеточного разбиения, а это резко снижает быстродействие алгоритмов Дейкстры, A* и других аналогичных алгоритмов.

С другой стороны, окрестности препятствий прямоугольного типа, препятствий, полученных объединением прямоугольных ячеек, или приближенных к ним, могут быть аппроксимированы меньшим количеством ячеек сетки, чем для препятствий со сложной полигональной границей при одной и той же точности приближения.

Заметим, что разработанный алгоритм может быть эффективно использован и при планировании перемещений группы роботов в средах с источниками помеховых возмущений со сложными по форме геометрическими областями действия; для этого дополнительно может быть использован аппарат характеристических вероятностных функций и/или функций риска с соответствующими методами решения задач поиска целевой траектории с наибольшей вероятностью успешного прохождения [20, 21].

Заключение. В работе представлена итерационная процедура, в которой использован bug-алгоритм локального планирования. В ходе разработки указанной процедуры как шага специального итерационного процесса оптимизации на графе, отражающем связь характерных точек полигональных препятствий кусочно-ломаными траекториями, bug-алгоритм обобщен до глобального алгоритма планирования.

Особенностью алгоритма является итерационный процесс анализа соответствий между текущим отрезком «прямой видимости», соединяющим текущую точку с целевой, и криволинейными полигональными границами препятствий, пересекаемых им.

В отличие от исходных bug-алгоритмов, предлагаемый метод решает задачу глобального поиска пути на классе заданных траекторий с полигональными препятствиями; а в отличие от классических методов диаграмм прямой видимости, существенно снижает размерность графа за счет специального выбора ограниченного количества характерных точек соответствующих препятствий.

Показано, что для прокладывания пути на сценах лабиринтного типа со сложными препятствиями, являющимися связным объединением некоторых ячеек прямоугольного сеточного разбиения, рассматриваемый алгоритм на 10% превосходит оптимальный алгоритм Дейкстры. В том случае, когда препятствия будут иметь более сложную полигональную форму, разработанный алгоритм будет еще более превосходить алгоритм Дейкстры по вычислительной эффективности.

Превосходство методов клеточной декомпозиции над разработанным методом прогнозируется только в тех случаях, когда число препятствий сцены начнет сильно превосходить число свободных ячеек клеточного разбиения.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Han-Pang Huang, Shu-Yun Chung*. Dynamic visibility graph for path planning // IEEE-RSJ International Conference on Intelligent Robots and Systems. Sendai, Japan, 2004. – Vol. 3. – P. 2813-2818.
2. *Janet J.A., Luo R.C., Kay M.G.* The essential visibility graph: An approach to global motion planning for autonomous mobile robots // IEEE International Conference on Robotics and Automation. Nagoya, Japan, 1995. – Vol. 2. – P. 1958-1963.
3. *Habib M.K., Asama H.* Efficient method to generate collision free paths for an autonomous mobile robot based on new free space structuring approach // IEEE-RSJ International Conference on Intelligent Robots and Systems. Osaka, Japan, 1991. – Vol. 2. – P. 563-567.
4. *Wallgrun J.O.* Voronoi graph matching for robot localization and mapping // Transactions on computational science. – Springer. 2010. – P. 76-108.
5. *LaValle S.M., Kuffner J.* Randomized kinodynamic planning // Int. Journal of Robotics Research. – 2001. – Vol. 20 (5). – P. 378-400.
6. *Пишихов В.Х., Медведев М.Ю., Бросалин Д.О., Васильева М.А., Гуренко Б.В., Хамдан Низар*. Исследование методов планирования движения в двумерных картографированных средах // Известия ЮФУ. Технические науки. – 2022. – № 3 (227). – С. 170-192.
7. *Пишихов В.Х., Медведев М.Ю., Костюков В.А., Хуссейн Ф., Кадим А.* Алгоритмы планирования траекторий в двумерной среде с препятствиями // Информатика и автоматизация. – 2022. – Т. 21 (3). – С. 459-492.
8. *Сухарев А.Г.* Об оптимальных стратегиях поиска экстремума // Журнал вычислительной математики и математической физики. – 1971. – Т. 11 (4). – С. 910-924.
9. *Hornung A., Wurm K.M., Bennet M., Stachniss C., Burgard W.* OctoMap: An efficient probabilistic 3D mapping framework based on octrees // Autonomous Robots. – 2013. – Vol. 34 (3). – P. 189-206.
10. *Janson L., Ichter B., Pavone M.* Deterministic sampling-based motion planning: Optimality, complexity, and performance // International Journal of Robotics Research. – 2018. – Vol. 37 (1). – P. 46-61.
11. *Hart P.E., Nilsson N.J., Raphael B.A.* Formal Basis for the Heuristic Determination of Minimum Cost Paths // IEEE Transactions on Systems Science and Cybernetics. – 1968. – Vol. 2. – P. 100-107.
12. *Stentz A.* Optimal and efficient path planning for partially known environments // In Intelligent Unmanned Ground Vehicles. – Springer, Boston, MA, USA, 1997. – P. 203-220.
13. *Koenig S., Likhachev M., Furcy D.* Lifelong Planning A* // Artificial Intelligence. – 2004. – Vol. 155 (1-2). – P. 93-146.
14. *Казаков К.А., Семенов В.А.* Обзор современных методов планирования движения // Тр. ИСП РАН. – 2016. – Т. 28 (4). – С. 241-294.
15. *Ng J., Braunl T.* Performance comparison of bug navigation algorithms // Journal of Intelligent and Robotic Systems. – 2007. – Vol. 50 (1). – P. 73-84.
16. *Lumelsky V., Stepanov A.* Dynamic path planning for a mobile automaton with limited information on the environment // IEEE Transactions on Automatic Control. – 1986. – Vol. 31 (11). – P. 1058-1063.
17. *Yufka A., Parlaktuna O.* Performance comparison of the BUG's algorithms for mobile robots // Proc. of International Symposium on Innovations in Intelligent Systems and Applications. Trabzon, Turkey, 2009. – P. 416-421.
18. *Костюков В.А., Медведев М.Ю., Пишихов В.Х.* Планирование движения наземных роботов в среде с препятствиями: алгоритмы построения траекторий в группе при заданном шаблоне // Мехатроника, автоматизация, управление. – 2023. – Т. 24. № 1. – С. 33-45.
19. *Костюков В.А., Медведев М.Ю., Пишихов В.Х.* Алгоритмы планирования сглаженных индивидуальных траекторий движения наземных роботов // Мехатроника, автоматизация, управление. – 2022. – Т. 23 (11). – С. 585-595.
20. *Kostjukov V., Pshikhov V., Medvedev M.* Optimization of mobile robot movement on a plane with finite number of repeller sources // SPIIRAS Proceedings. – 2020. – Vol. 19 (1). – P. 43-78.
21. *Zabarankin M., Uryasev S., Pardalos P.* Optimal Risk Path Algorithms // Cooperative Control and Optimizaton. Dordrecht: Kluwer Acad., 2002. – P. 271-303.

REFERENCES

1. Han-Pang Huang, Shu-Yun Chung. Dynamic visibility graph for path planning, *IEEE-RSJ International Conference on Intelligent Robots and Systems. Sendai, Japan, 2004*, Vol. 3, pp. 2813-2818.
2. Janet J.A., Luo R.C., Kay M.G. The essential visibility graph: An approach to global motion planning for autonomous mobile robots, *IEEE International Conference on Robotics and Automation. Nagoya, Japan, 1995*, Vol. 2, pp. 1958-1963.
3. Habib M.K., Asama H. Efficient method to generate collision free paths for an autonomous mobile robot based on new free space structuring approach, *IEEE-RSJ International Conference on Intelligent Robots and Systems. Osaka, Japan, 1991*, Vol. 2, pp. 563-567.
4. Wallgrun J.O. Voronoi graph matching for robot localization and mapping, *Transactions on computational science*. Springer. 2010, pp. 76-108.
5. LaValle S.M., Kuffner J. Randomized kinodynamic planning, *Int. Journal of Robotics Research*, 2001, Vol. 20 (5), pp. 378-400.
6. Pshikhopov V.Kh., Medvedev M.Yu., Brosalin D.O., Vasil'eva M.A., Gurenko B.V., Khamdan Nizar. Issledovanie metodov planirovaniya dvizheniya v dvumernykh kartografirovannykh sredakh [Study of path planning methods in two-dimensional mapped environments], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2022, No. 3 (227), pp. 170-192.
7. Pshikhopov V.Kh., Medvedev M.Yu., Kostyukov V.A., Khusseyn F., Kadim A. Algoritmy planirovaniya traektoriy v dvumernoy srede s prepyatstviyami [Algorithms for trajectory planning in a two-dimensional environment with obstacles], *Informatika i avtomatizatsiya* [Informatics and Automation], 2022, Vol. 21 (3), pp. 459-492.
8. Sukharev A.G. Ob optimal'nykh strategiyakh poiska ekstremuma [About optimal strategies for finding the extremum], *Zhurnal vychislitel'noy matematiki i matematicheskoy fiziki* [Journal of Computational Mathematics and Mathematical Physics], 1971, Vol. 11 (4), pp. 910-924.
9. Hornung A., Wurm K.M., Bennet M., Stachniss C., Burgard W. OctoMap: An efficient probabilistic 3D mapping framework based on octrees, *Autonomous Robots*, 2013, Vol. 34 (3), pp. 189-206.
10. Janson L., Ichter B., Pavone M. Deterministic sampling-based motion planning: Optimality, complexity, and performance, *International Journal of Robotics Research*, 2018, Vol. 37 (1), pp. 46-61.
11. Hart P.E., Nilsson N.J., Raphael B.A. Formal Basis for the Heuristic Determination of Minimum Cost Paths, *IEEE Transactions on Systems Science and Cybernetics*, 1968, Vol. 2, pp. 100-107.
12. Stentz A. Optimal and efficient path planning for partially known environments, *In Intelligent Unmanned Ground Vehicles*. Springer, Boston, MA, USA, 1997, pp. 203-220.
13. Koenig S., Likhachev M., Furcy D. Lifelong Planning A*, *Artificial Intelligence*, 2004, Vol. 155 (1-2), pp. 93-146.
14. Kazakov K.A., Semenov V.A. Obzor sovremennykh metodov planirovaniya dvizheniya [Overview of modern traffic planning methods], *Tr. ISP RAN* [Proceedings of ISP RAS], 2016, Vol. 28 (4), pp. 241-294.
15. Ng J., Brauml T. Performance comparison of bug navigation algorithms, *Journal of Intelligent and Robotic Systems*, 2007, Vol. 50 (1), pp. 73-84.
16. Lumelsky V., Stepanov A. Dynamic path planning for a mobile automaton with limited information on the environment, *IEEE Transactions on Automatic Control*, 1986, Vol. 31 (11), pp. 1058-1063.
17. Yufka A., Parlaktuna O. Performance comparison of the BUG's algorithms for mobile robots, *Proc. of International Symposium on Innovations in Intelligent Systems and Applications. Trabzon, Turkey, 2009*, pp. 416-421.
18. Kostyukov V.A., Medvedev M.Yu., Pshikhopov V.Kh. Planirovanie dvizheniya nazemnykh robotov v srede s prepyatstviyami: algoritmy postroeniya traektoriy v gruppe pri zadannom shablone [Planning the movement of ground robots in an environment with obstacles: algorithms for constructing trajectories in a group with a given formation], *Mekhatronika, avtomatizatsiya, upravlenie* [Mechatronics, Automation, Control], 2023, Vol. 24, No. 1, pp. 33-45.

19. Kostyukov V.A., Medvedev M.Yu., Pshikhopov V.Kh. Algoritmy planirovaniya sglazhennykh individual'nykh traektoriy dvizheniya nazemnykh robotov [Algorithms for planning smoothed individual trajectories of ground robots], *Mekhatronika, avtomatizatsiya, upravlenie* [Mechatronics, Automation, Control], 2022, Vol. 23 (11), pp. 585-595.
20. Kostjukov V. Pshikhopov V., Medvedev M. Optimization of mobile robot movement on a plane with finite number of repeller sources, *SPIIRAS Proceedings*, 2020, Vol. 19 (1), pp. 43-78.
21. Zabaranin M., Uryasev S., Pardalos P. Optimal Risk Path Algorithms, *Cooperative Control and Optimizaton*. Dordrecht: Kluwer Acad., 2002, pp. 271-303.

Статью рекомендовал к опубликованию д.т.н., профессор В.В. Курейчик.

Костюков Владимир Александрович – НКБ «РиСУ»; e-mail: wkost-einheit@yandex.ru; г. Таганрог, Россия; тел.: 88634371694; к.т.н.; с.н.с.

Медведев Михаил Юрьевич – НИИ робототехники и процессов управления Южного федерального университета; e-mail: medvmihal@sfedu.ru; г. Таганрог, Россия; тел.: 88634371694; д.т.н.; в.н.с.

Пшихопов Вячеслав Хасанович – e-mail: pshichop@rambler.ru; тел.: 88634371694; д.т.н.; профессор; директор.

Kostyukov Vladimir Aleksandrovich – Joint-Stock Company "Robotics and Control Systems"; e-mail: wkost-einheit@yandex.ru; Taganrog, Russia; phone: 88634371694; can. of eng. sc.; senior researcher.

Medvedev Mikhail Yurjevich – R&D Institute of Robotics and Control Systems; e-mail: medvmihal@sfedu.ru; Taganrog, Russia; phone: 88634371694; dr. of eng. sc.; leading researcher.

Pshikhopov Viacheslav Khasanovich – e-mail: pshichop@rambler.ru; phone: 88634371694; dr. of eng. sc.; professor; director.

УДК 621.3.049.771.14

DOI 10.18522/2311-3103-2023-5-48-57

Д.Б. Шокарев, Р.Ж. Чочаев, А.Н. Щелоков, С.В. Гаврилов

РАЗРАБОТКА АЛГОРИТМА ДЕТАЛЬНОГО РАЗМЕЩЕНИЯ НА ПЛИС

Иерархические программируемые логические интегральные схемы (ПЛИС) состоят из множества логических блоков, объединенных в группы. Для успешной трассировки необходимо оптимальное размещение элементов в пределах групп с учётом особенностей архитектуры локальных связей. Классические алгоритмы не способны обеспечить учёт различных особенностей архитектуры. Решение данной проблемы возможно только путем разработки специализированных алгоритмов. В данной работе представлен алгоритм детального размещения, в котором для вычисления оптимальных позиций элементов в группе была разработана новая метрика, позволяющая оценить количество доступных локальных связей между элементами в группах логических блоков с учётом особенностей архитектуры связей между ними. Алгоритм детального размещения состоит из нескольких этапов. На первом этапе группа логических элементов представляется в виде ориентированного графа. На втором этапе определяется порядок размещения логических элементов в группе с помощью алгоритма поиска в ширину. На финальном этапе для каждого элемента, согласно полученному порядку, определяется оптимальное размещение в группе с учётом разработанной метрики. Если среди свободных позиций для размещения в группе нет оптимальной, то проверяются занятые позиции. Текущий элемент назначается на занятую позицию, а для замененного элемента выполняется поиск новой. Такая замена может проводиться многократно, увеличивая вероятность нахождения оптимальной конфигурации. Предложенный алгоритм был реализован и протестирован на наборах тестовых схем. На основе результатов тестирования выполнено сравнение представленного алгоритма с алгоритмом последовательного размещения. Сравнение алгоритмов показало, что приме-