

21. Sabater J., Sierra C. REGRET: a reputation model for gregarious societies, *Proceedings of the 4th international workshop on deception, fraud and trust in agent societies*, 2005, pp. 61-69.
22. Babenko L.K., Tolomanenko E.A. Development of algorithms for data transmission in sensor networks based on fully homomorphic encryption using symmetric Kuznyechik algorithm, *Journal of Physics: Conference Series*, 2021, 1812, 012034. DOI: 10.1088/1742-6596/1812/1/012034.

Статью рекомендовал к опубликованию д.т.н. Г.Е. Веселов.

Бабенко Людмила Климентьевна – Южный федеральный университет; e-mail: lkbabenko@sfnedu.ru; г. Таганрог, Россия; тел.: 89054530191; кафедра безопасности информационных технологий; профессор.

Чудинов Павел Юрьевич – e-mail: chudinov@sfnedu.com; тел.: 89198722200; кафедра безопасности информационных технологий; аспирант.

Рогозов Юрий Иванович – e-mail: yrogozov@sfnedu.ru; тел.: +78634371787; кафедра системного анализа и телекоммуникаций; профессор.

Babenko Lyudmila Klimentevna – Southern Federal University; e-mail: blk@fib.tsure.ru; Taganrog, Russia; phone: +79054530191; the department of security of information technologies; professor.

Chudinov Pavel Yurevich – e-mail: chudinov@sfnedu.ru.com; phone: +79198722200; the department of information technology security; postgraduate student.

Rogozov Yury Ivanovich – e-mail: yrogozov@sfnedu.ru; phone: +78634371787; the department of system analysis and telecommunications; professor.

УДК 004.896

DOI 10.18522/2311-3103-2022-5-47-62

О.Б. Лебедев, А.А. Жиглатый

КО-ЭВОЛЮЦИОННЫЙ АЛГОРИТМ РАЗМЕЩЕНИЯ НА ОСНОВЕ ВЗАИМОДЕЙСТВИЯ СУБПОПУЛЯЦИЙ, ОТЛИЧАЮЩИХСЯ СТРАТЕГИЯМИ ПОИСКА*

Разработана новая методология и метод размещения элементов СБИС, отличающиеся тем, что решение задачи размещения основывается на использовании фиксированного порядка выбора позиций, ориентированного на эффективное решение задачи размещения, и эвристической процедуры распределения элементов по позициям, позволяющие снизить общую трудоемкость, и повысить качество решения. Процесс формирования списка позиций на коммутационном поле осуществляется с использованием механизмов волнового алгоритма. В основу выбора окончательного списка положен принцип построения маршрута с минимальной оценкой суммарной линейной длины расстояний между позициями маршрута. Для решения задачи размещения разработан поисковый алгоритм на основе модифицированного метода муравьиной колонии. Для исключения преждевременной сходимости и локализации глобального экстремума задачи разработка алгоритма производилась на основе ко-эволюционного подхода. Архитектура ко-эволюционного алгоритма размещения разработана на основе парадигмы муравьиного алгоритма. В пространстве поиска субпопуляции параллельно реализуют четыре стратегии оптимизации. В работе процесс ко-эволюции реализован на основе взаимодействия субпопуляций, отличающихся стратегиями поиска. Отличительной особенностью используемого ко-эволюционного подхода является то, что субпопуляции решений фактически являются виртуальными. Процесс ко-эволюции реализуется одной популяцией агентов Z путем последовательного формирования и слияния, виртуальных субпопуляций решений в одну популяцию. В работе решение задачи размещения направлено на повышение трассируемости посредством миними-

* Работа выполнена при финансовой поддержке гранта РФФИ № 20-07-00260 А.

зации ресурсов, требуемых для реализации соединений. Существенный вклад в минимизацию пространственной и временной сложности поисковой процедуры внесли: использование виртуальными субпопуляциями общей эволюционной памяти, общего графа поиска решений, формирование единой интерпретации решения в виде маршрута на полном ориентированном графе с бинарными ориентированными ребрами. Тестирование производилось на бенчмарках 19s, PrimGA1, PrimGA2. Результаты по сравнению с существующими алгоритмами улучшены на 7-8%. Вероятность получения глобального оптимума составила 0.96. В среднем решения отличаются от оптимального менее, чем на 1.5%. Временная сложность алгоритма при фиксированных значениях размера популяции и количества генераций составляет $O(n)$. Общая временная сложность гибридного алгоритма составляет $O(n^2)$ – $O(n^3)$.

СБИС; размещение; роевой интеллект; муравьиный алгоритм; адаптивное поведение; ко-эволюция; субпопуляция; оптимизация.

O.B. Lebedev, A.A. Zhiglatiy

CO-EVOLUTIONARY PLACEMENT ALGORITHM BASED ON INTERACTION SUBPOPULATIONS DIFFERING IN SEARCH STRATEGIES

A new methodology and method for placing VLSI elements has been developed, which differ in that the solution of the placement problem is based on the use of a fixed order of position selection, focused on the effective solution of the placement problem, and a heuristic procedure for distributing elements by positions, which reduces the overall complexity and improves the quality of the solution. The process of forming a list of positions on the switching field is carried out using the mechanisms of the wave algorithm. The choice of the final list is based on the principle of constructing a route with a minimum estimate of the total linear length of distances between route positions. To solve the placement problem, a search algorithm based on the modified ant colony method has been developed. To exclude premature convergence and localization of the global extremum of the problem, the development of the algorithm was carried out on the basis of the co-evolutionary approach. The architecture of the co-evolutionary placement algorithm is developed on the basis of the ant colony algorithm paradigm. In the search space, sub-populations implement four optimization strategies in parallel. In the work, the coevolution process is implemented on the basis of the interaction of subpopulations that differ in search strategies. A distinctive feature of the co-evolutionary approach used is that subpopulations of solutions are actually virtual. The process of co-evolution is implemented by one population of agents Z by sequential formation and merging of virtual subpopulations of solutions into one population. In this paper, the solution of the placement problem is aimed at improving traceability by minimizing the resources required to implement connections. A significant contribution to minimizing the spatial and temporal complexity of the search procedure was made by: the use by virtual sub-populations of a common evolutionary memory, a common solution search graph, the formation of a single interpretation of the solution in the form of a route on a complete directed graph with binary directed edges. Testing was carried out on benchmarks 19s, PrimGA1, PrimGA2. The results compared to existing algorithms are improved by 7-8%. The probability of obtaining a global optimum was 0.96. On average, solutions differ from the optimal by less than 1.5%. The time complexity of the algorithm for fixed values of the population size and the number of generations is $O(n)$. The total time complexity of the hybrid algorithm is $O(n^2)$ – $O(n^3)$.

VLSI; placement; swarm intelligence; ant colony algorithm; adaptive behavior; co-evolution; subpopulation; optimization.

Введение. Одной из важнейших задач конструкторского проектирования СБИС является задача размещения элементов на кристалле. Именно размещение во многом определяет качество последующей трассировки. В существующих алгоритмах [1, 2], с одной стороны, связь между этими задачами недостаточно глубока, с другой стороны, получаемые решения, с точки зрения их оптимальности, как правило, неудовлетворительны. Необходимость повышения качества проектирования требует поиска новых путей и подходов к решению задач размещения.

Анализ разработанных алгоритмов размещения (детерминированных, случайного поиска, последовательных, итерационных) показывает, что для создания эффективного алгоритма размещения, отвечающего современным требованиям, необходимы новые технологии и подходы [3]. В работе предлагаются новые технологии, принципы и механизмы решения задачи размещения, основанные на моделировании процессов, инспирированных природными системами.

В гибридных алгоритмах преимущества одного алгоритма могут компенсировать недостатки другого. Интеграция метаэвристик популяционных алгоритмов обеспечивает более широкий обзор пространства поиска и более высокую вероятность локализации глобального экстремума задач. Суть ко-эволюционного подхода состоит в том, что для решения задачи используется набор субпопуляций, которые эволюционируют параллельно. Предполагается, что субпопуляции отличаются стратегиями поиска, но каждая из субпопуляций решает одну и ту же задачу. Эпизодически ко-эволюционирующие системы обмениваются опытом. Такой подход позволяет частично решить проблему преждевременной сходимости, обеспечивает выход из локальных оптимумов и повышает скорость получения результата [4].

Общая структура муравьиного алгоритма (МА) реализует итерационную процедуру поиска лучшего решения, в которую включается конструктивный алгоритм. Канонический МА решает задачу нахождения кратчайшего маршрута в полном графе. Маршрут является кодом решения некоторой задачи на полном графе поиска решений (ГПР) [5].

Для повышения эффективности, усиления сходимости алгоритма и способности выхода из локальных оптимумов предложен подход к ко-гибридизации [6] алгоритма размещения на основе модели адаптивного поведения муравьиной колонии.

Одновременно в пространстве поиска решения задачи эволюционируют четыре субпопуляции решений. Субпопуляции включают маршруты, формируемые в соответствии со стратегиями *S1-S4* соответственно. Стратегии базируются на парадигме МА, но отличаются конструктивными алгоритмами (КА) размещения элементов, используемыми в составе МА.

1. Постановка задачи размещения. Проблема размещения может быть сформулирована следующим образом. Дано множество прямоугольных элементов (модулей) $A = \{a_i | i = 1, \dots, n_a\}$ с расположенными на них терминалами (выводами), где n_a – количество модулей. Задано множество цепей, связывающих терминалы модулей. Задано коммутационное поле (КП), на котором могут размещаться элементы. Вводится прямоугольная система координат XOY , у которой оси OX и OY совпадают соответственно с нижней и боковой сторонами КП. Заданы координаты размещения множества позиций $P = \{p_j | j = 1, \dots, n_p\}$ на КП. Необходимо разместить элементы на КП с оптимизацией некоторых критериев качества. Основными показателями, в соответствии с которыми классифицируются существующие алгоритмы это постановка задачи размещения и метод поиска решений. Характеристиками, определяющими суть постановки задачи размещения, являются:

- ◆ модель представления размещаемых (конструктивных) элементов как геометрических объектов;
- ◆ модель монтажного пространства (пространство позиций);
- ◆ модель представления электрической цепи;
- ◆ характер целевой функции для оценки размещения [7].

Для каждой позиции заданы координаты. Расстояние между позициями обеспечивает размещения элементов в любой комбинации без наложения друг на друга и с учетом конструктивных ограничений (требования электромагнитной и тепловой совместимости) [2, 3]. На рис. 1 приведена модель КП. Чаще всего размещаемые элементы рассматриваются как точки (вершины). Пространство позиций представляется в виде регулярной структуры (рис. 2).

Пусть дано множество элементов $A=\{a_i|i=1, \dots, n_a\}$ и множество позиций $P=\{p_i|i=1, \dots, n_p\}$ – на КП. Для размещения всех элементов необходимо выполнение условия $n_p \geq n_a$.

Решение задачи размещения представляется в виде биективного отображения $f_R=A \rightarrow P$, каждому элементу множества A соответствует один единственный элемент множества P и наоборот $f_R=P \rightarrow A$. Номер позиции p_j в которую помещен элемент a_i определяется соотношением $p_j=f_R(a_i)$. $a_i=f_R^{-1}(p_j)$.

Произвольное размещение элементов в позициях представляет собой перестановку $M=\langle m(1), m(2), \dots, m(t), \dots, m(c) \rangle$, где $m(t)$ задает номер модуля, который назначен в позицию p_t . В зависимости от выбранного критерия для оценки результатов размещения вводится целевая функция $\zeta(M)$.

Таким образом, задача размещения состоит в отыскании оптимального значения функции ζ на множестве перестановок M .

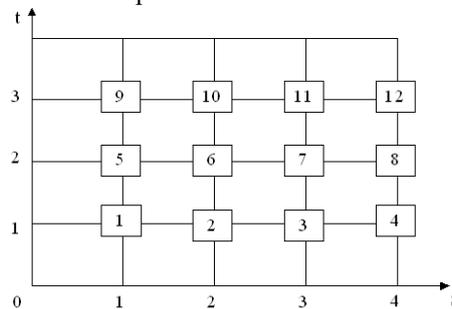


Рис. 1. Модель плоскости для размещения

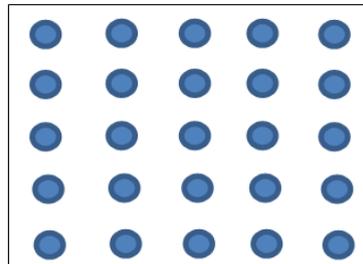


Рис. 2. Расположение позиций на коммутационном поле

Главными задачами размещения является минимизация общей площади кристалла, создание благоприятных условий для последующей трассировки (проблема трассируемости) и минимизации оценки общей длины соединений. Минимизация общей длины соединений необходима для уменьшения временных задержек возникающих, в длинных цепях и в связи с этим увеличения скорости обработки информации в СБИС [8, 9].

Задачи размещения элементов относятся к классу NP -трудных задач комбинаторной оптимизации. Основными известными критериями при размещении [1–3] являются: минимальная суммарная длина связей, минимальная длина самой длинной связи, минимум числа возможных пересечений, минимум числа изгибов соединений, минимальная площадь кристалла. Все эти критерии косвенно способствуют решению основной задачи: максимизировать число реализованных соединений при последующей трассировке [8–11].

В качестве модели схемы используется граф $B=(X,R)$ или гиперграф $H=(X,E)$, где $X=\{x_i|i=1,\dots,n_x\}$ – множество вершин, моделирующих элементы, а $R=\{r_j|j=1,\dots,n_r\}$ – множество ребер. Вершина x_i связана с вершиной x_j ребром r_{ij} , если соответствующие элементы связаны соединением. $E=\{c_j|c_j\subset X, j=1,\dots,n_c\}$ – множество гиперребер, моделирующих цепи, связывающие элементы. c_j – множество вершин графа H , связываемых цепью c_j . Граф адекватно моделирует двухтерминальные соединения, а гиперграф – многотерминальные.

Расстояние между двумя элементами a_i и a_j с координатами (x_i, y_i) и (x_j, y_j) определяется по формуле: $r_{ij} = |x_i - x_j| + |y_i - y_j|$.

В качестве оценки l_j длины цепи t_j , моделируемой гиперребром c_j , используются: длина минимального связывающего дерева, построенного на множестве вершин $c_j\subset X$; длина звездного графа, ребра которого инцидентны вершинам $c_j\subset X$, а корневая вершина помещена в центре «тяжести» множества вершин c_j ; длина полупериметра прямоугольника, описывающего множество вершин c_j ; суммарная длина ребер полного графа, построенного на множестве c_j [3].

Первый критерий оптимизации имеет вид: $\xi_j = \sum_j(l_j)$. Цель оптимизации минимизация ξ_j .

На качество решения задачи трассировки при проектировании топологии СБИС значительное влияние оказывают начальные условия, сформированные в результате решения предшествующих трассировке задач конструкторского проектирования – покрытия, разбиения, планирования кристалла, размещения. Оценкой *трассируемости* является соотношение между ресурсами коммутационной среды (КС) и ресурсами, требуемыми для реализации соединений. В работе решение задачи размещения направлено на повышение *трассируемости* посредством минимизации ресурсов требуемых для реализации соединений [1–3, 12].

Формирование критерия размещения, учитывающим ресурсы КС, производится на *дискретном рабочем поле* (ДРП), представляющем собой ограниченную замкнутой линией прямоугольную фигуру в виде решетки $D=(V,R)$. Решетка разбита на квадратные ячейки, с ребрами единичной длины, где $V=\{v_i|i=1,\dots,n_v\}$ – множество узлов решетки, а $R=\{r_j|j=1,\dots,n_r\}$ – множество ребер решетки. Размеры ячеек и их количество определяется площадью поля, и размерами (площадью) позиций, необходимой для размещения в них элементов. Позиции p_i располагаются внутри ячеек решетки. Координаты (x_i, y_i) ячеек и размещенных в них позиций совпадают. Между позициями P и координатами Q их расположения на ДРП установлено биективное соответствие $f_q: P \rightarrow Q$, где $Q=\{(x_i, y_i)|i=1, \dots, n_m\}$. $f_q(p_i)=(x_i, y_i)$.

В работе модель ДРП (рис. 3,а) представляется в виде матрицы $D=||d_{ij}||_{n \times m}$. Элемент d_{ij} матрицы соответствует ячейке ДРП. Значением элемента d_{ij} матрицы может быть номер элемента схемы или номер позиции, помещенных в ячейку, или расстояние до стартовой вершины.

На рис. 3 значением элемента матрицы является номер элемента схемы, размещенного в соответствующей ячейке.

Любые две смежные ячейки ДРП могут рассматриваться как единая область, ограниченная контуром. На (рис. 3,б,с,д) представлены варианты областей, модель ДРП в виде матрицы. Контур является границей объединенной области ДРП. Ребра контуров используются в качестве индикаторов распределения трассировочных ресурсов КС.

С помощью множества $\Gamma=\{\gamma_j|j=1,\dots,n_\gamma\}$ для каждого ребра r_i решетки задается пропускная способность γ_j – число соединений, которые могут его пересечь. Значения γ_j определяются размерами ребра и параметрами соединений: толщина, допустимое расстояние между трассами и т.п.

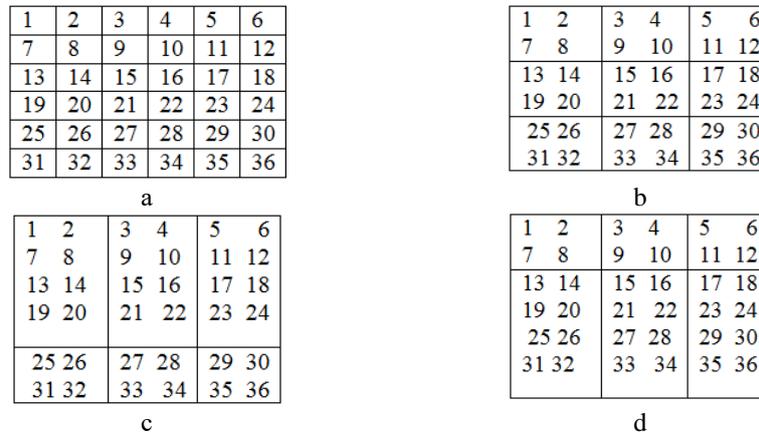


Рис. 3. Разбиение коммутационного поля на прямоугольные области

Границу некоторой объединенной области ω_k коммутационного поля образует цикл L_k , составленный из ребер сетки. Пропускная способность ψ_k границы L_k определяется, как сумма пропускных способностей ребер решетки, входящих в состав L_k , $\psi_k = \sum \gamma_{ij} / (\forall j | r_j \in L_k)$.

Пусть φ_k – число цепей, связывающих элементы, размещенные внутри области ω_k , ограниченной L_k , с элементами, расположенными вне этой области. В качестве характеристики границы используется параметр $\delta_k = (\psi_k - \varphi_k) / \psi_k$. Чем большее значение имеет δ_k , тем легче осуществить прокладку связей через границу L_k и выше трассируемость КС.

Пусть задано некоторое размещение элементов в ячейках ДРП и некоторое множество областей Ω , для которых определено множество границ $L = \{L_k | k=1, \dots, n_k\}$. Найдем среди характеристик границ наименьшую δ_{min} , т.е. $\forall k [(\psi_k - \varphi_k) / \psi_k \geq \delta_{min}]$.

Параметр $\zeta_2 = \delta_{min}$ используется в качестве критерия оптимизации. Цель оптимизации – максимизация δ_{min} .

Известно, что при одной и той же площади области, ограниченной некоторым контуром (а следовательно, при одном и том же числе позиций в области), наименьшую длину имеет контур квадратной формы. Следовательно, такая граница будет наиболее «чувствительна» к пересекающим ее цепям, т.к. для одного и того же множества элементов будет иметь наименьшую оценку ψ_k , а следовательно, и γ_k . Поэтому в работе используется система контуров, имеющих квадратную форму или по возможности близкую к ней.

Формирование множества контуров (границ) производится следующим образом [3]. Пусть имеется ДРП размером $X \times Y$, $X \leq Y$. Единицей измерения служит длина одного ребра опорной сетки (рис. 1). Выбираем контур с размерами $x \times x$, $x \leq X$. Сначала он помещается в левый угол ДРП, а затем путем сканирования (последовательного сдвига на один шаг вправо или вниз) формируется набор контуров. Число таких контуров определяется так: $n_k = (X - x + 1) \cdot (Y - x + 1)$.

2. Разработка ко-эволюционного алгоритма размещения на основе моделей адаптивного поведения муравьиной колонии. Основу любого последовательного алгоритма размещения составляют две процедуры: процедура выбора очередной позиции (ПВП) для назначения в неё выбранного элемента и процедура выбора очередного элемента (ПВЭ). В зависимости от способов, лежащих в основе этих процедур, в алгоритмах размещения используется четыре подхода.

1. Порядок выбора элементов (ПВЭ) и позиций (ПВП) задается априори.

2. Порядок выбора позиций фиксированный – задается априори. Порядок выбора элементов на основе эвристик.

3. Порядок выбора элементов – априори, позиций на основе эвристик.

4. Порядок выбора элементов и позиций на основе эвристик.

Наименьшей трудоемкостью обладают алгоритмы 1 класса, наибольшей – 4 класса. Анализ существующих методов и алгоритмов размещения [1–7] показал, что в их основе в явном или неявном виде лежит эвристическая процедура упорядочивания исходного списка позиций и элементов, последовательно распределяемых по позициям.

В работе реализован второй подход. Априори формируется поиск $S_w = \langle s_{wt}/t=1, \dots, n_s \rangle$ с фиксированным порядком выбора позиций, ориентированным на эффективное решение задачи размещения. Поиск решения задачи размещения выполняется процедурой последовательного распределения элементов по позициям в заданном порядке [12–15].

Решение задачи размещения представляется в виде двух списков:

- ◆ предварительно построенного и зафиксированного;

- ◆ *общего для популяции*, упорядоченного списка позиций $S_w = \langle s_{wt}/t=1, \dots, n_s \rangle$, где s_{wt} – соответствует номеру p_i позиции на ДРП, $p_i = f(s_{wt})$, и построенного вектора $M = \langle m_t/t=1, \dots, n_m \rangle$, где значением элемента m_t – в векторе M является номер a_t модуля, $a_t = f_m(m_t)$, размещенного в позиции s_{wt} . Если число размещаемых элементов n_m на d меньше числа позиций n_p , то в конце вектора M добавляется d элементов m_t с нулевым значением.

Пусть $S_w = \langle 3, 7, 5, 4 \rangle$, $M = \langle 11, 9, 8, 1 \rangle$: это значит, что элемент a_{11} в позиции p_3 , a_9 в позиции p_7 , a_8 в позиции p_5 , a_1 в позиции p_4 .

Для решения задачи размещения разработан поисковый алгоритм на основе модифицированного метода муравьиной колонии (МК) [9, 16].

Для исключения преждевременной сходимости и локализации глобального экстремума задача разработка алгоритма производилась на основе ко-эволюционного подхода [9, 12].

Одновременно в пространстве поиска решения задачи размещения эволюционируют четыре субпопуляции решений, каждая из которых имеет свою стратегию оптимизации, базирующуюся на методе МК.

Ядром муравьиного алгоритма является конструктивный алгоритм, с помощью которого агенты на каждой итерации находят решение задачи [9, 12].

Предварительно для заданного порядка расположения позиций на ДРП формируются и фиксируются четыре упорядоченных списка позиций: S_1 – S_4 . За каждой субпопуляцией закрепляется свой список позиций. Стратегии каждой субпопуляции агентов отличаются конструктивными алгоритмами размещения, входящими в состав муравьиного алгоритма.

Основная идея конструктивного алгоритма размещения элементов по позициям заключается в следующем. Агент последовательно по порядковому номеру t , на каждом шаге, начиная с первого, выбирает в векторе (списке S_w) позицию s_{wt} , в которую помещает модуль, выбираемый на основе эвристических правил. Конструктивные алгоритмы, используемые субпопуляциями, отличаются порядками просмотра позиций на ДРП, задаваемым упорядоченными списками позиций: S_1 – S_4 .

На рис. 4 приведен пример расположения позиций на ДРП.

Процесс формирования списка позиций на КП осуществляется с использованием механизмов волнового алгоритма. Множество всех ячеек ДРП разбивается на подмножества: «проходимые» (свободные), т.е. при поиске пути их можно проходить, «непроходимые» (препятствия), путь через эту ячейку запрещён. Фиксируется стартовая ячейка (источник). Алгоритм предназначен для поиска кратчайшего от стартовой ячейки маршрута, включающем заданное число ячеек.

В качестве стартовых выбираются ячейки ДРП, расположенные по краям решетки: в верхнем и нижнем углах левого края; в верхнем и нижнем углах правого края. В стартовой ячейке присваивается значение весовой оценки, равное единице.

От стартовой ячейки порождается шаг в соседние ячейки. Соседними ячейками считаются только 4 ячейки по вертикали и горизонтали. Элементы, в которые пришла волна, образуют фронт волны. Каждый элемент первого фронта волны является источником вторичной волны. Элементы второго фронта волны генерируют волну третьего фронта и т.д. Волна, в свою очередь, не может проходить через ячейки, помеченные как «пройденные» или «непроходимые».

В процессе распространения волны ячейкам решетки присваиваются весовые оценки. При выполнении условий проходимости и непринадлежности к ранее помеченным в пути ячейкам, в атрибут ячейки записывается число, равное количеству шагов от стартовой ячейки, на первом шаге это будет 1. Каждая ячейка, помеченная числом шагов от стартовой ячейки, становится стартовой и из неё порождаются очередные шаги в соседние ячейки. Волна движется, пока не останется непомяченных ячеек.

Далее, формируется по убыванию весовых оценок список S номеров позиций, от конечной помеченной ячейки ДРП, до стартовой ячейки ДРП, размещенных в этих ячейках.

В основу выбора окончательного списка положен принцип построения маршрута с минимальной оценкой суммарной линейной длины расстояний между позицией p_i и остальными позициями маршрута. В каждом списке позиции располагаются в порядке увеличения расстояния от стартовой вершины до позиции. В работе в качестве стартовых ячеек используются позиции, расположенные в левом верхнем углу ДРП, правом верхнем углу ДРП, левом нижнем углу ДРП, правом нижнем углу ДРП.

Варианты стратегий $S1-S4$ задаются соответственно с помощью списков: S_1-S_4 .

Пример. На рис. 4,а представлена модель ДРП в виде матрицы размером 5×5 , на которой отражены номера ячеек. В табл. 1 представлены координаты ячеек.



Рис. 4. Распространение волны. Стартовая вершина №1

Таблица 1

Я	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
x	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
y	5	5	5	5	5	4	4	4	4	4	3	3	3	3	3	2	2	2	2	2	1	1	1	1	1

На рис. 4,б представлена модель ДРП, в которой отражены весовые оценки ячеек после распространения волны от ячейки №1, расположенной в левом верхнем углу матрицы.

На основе матрицы (рис. 4,б) с весовыми оценками ω_i всех ячеек ДРП, полученных после распространения волны, формируется упорядоченный по возрастанию значений ω_i список $\Omega = \{\omega_i | i=1, \dots, n_\omega\}$, где $\omega_i \geq \omega_{i-1}$ и соответствующий список позиций S_ω . Список Ω имеет вид, представленный в табл. 2.

Таблица 2

Упорядоченный по возрастания значений список Ω и список позиций S_w

Ω	1	2	2	3	3	3	4	4	4	4	5	5	5	5	5	6	6	6	6	7	7	7	8	8	9
S_j	1	6	2	7	11	3	12	8	16	4	13	17	9	21	5	18	14	22	10	19	23	15	24	20	25

В свою очередь элементы списка Ω , имеющие одинаковое значение, упорядочиваются по возрастанию суммарного расстояния между ячейкой с весом w_i и всеми ячейками предыдущей волны (рис. 4,с). Дополнительно упорядоченный список Ω^* имеет вид, представленный в табл. 3.

Таблица 3

Дополнительно упорядоченный список Ω^*

Ω^*	1	2_1	2_2	3_1	3_2	3_3	4_1	4_2	4_3	4_4	5_1	5_2	5_3	5_4	5_5	6_1	6_2	6_3	6_4	7_1	7_2	7_3	8_1	8_2	9_1
S_j	1	6	2	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Сформированный список используется субпопуляцией $S1$.

На рис. 5 и в табл. 4 и 5 представлены результаты распространения волны и формирования списка S_2 позиций в случае расположения стартовой ячейки в нижнем левом углу.

Нумерация ячеек на ДРП

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

а

Оценки позиций

5	6	7	8	9
4	5	6	7	8
3	4	5	6	7
2	3	4	5	6
1	2	3	4	5

б

Идентификация позиций

5_4	6_3	7_2	8_1	9_1
4_3	5_2	6_1	7_1	8_2
3_2	4_1	5_1	6_2	7_3
2_1	3_1	4_2	5_3	6_4
1	2_2	3_3	4_4	5_5

с

Рис. 5. Распространение волны. Стартовая вершина №2

Таблица 4

Результаты распространения волны

Ω	1	2	2	3	3	3	4	4	4	4	5	5	5	5	5	6	6	6	6	7	7	7	8	8	9
S_2	21	16	22	17	11	23	12	18	6	24	13	7	19	1	25	8	14	2	20	9	3	15	4	10	5

Таблица 5

Формирования списка S_2 позиций

Ω^*	1	2_1	2_2	3_1	3_2	3_3	4_1	4_2	4_3	4_4	5_1	5_2	5_3	5_4	5_5	6_1	6_2	6_3	6_4	7_1	7_2	7_3	8_1	8_2	9_1
S_2	21	16	22	17	11	23	12	18	6	24	13	7	19	1	25	8	14	2	20	9	3	15	4	10	5

Аналогичным образом осуществляется формирование списков позиций S_3 и S_4 в случае расположения стартовой ячейки в правом верхнем и нижнем углах ДРП.

Таблица 6

Формирование списков позиций S_3

Ω	1	2	2	3	3	3	4	4	4	4	5	5	5	5	5	6	6	6	6	7	7	7	8	8	9
S_3	5	10	4	9	15	3	14	8	20	2	13	19	7	25	1	18	12	24	6	17	23	11	22	16	21

Таблица 7

Формирование списков позиций S_4

Ω	1	2	2	3	3	3	4	4	4	4	5	5	5	5	5	6	6	6	6	7	7	7	8	8	9
S_4	25	20	24	19	15	23	14	18	10	22	13	9	17	5	21	8	12	4	16	7	3	11	2	6	1

Поиск агентами решений (размещения элементов в позиции фиксированного списка) осуществляется на полном графе поиска решений $G=(V,U)$, $U=U1 \cup U2$. Вершины множества $V=\{v_i | i=1, \dots, n_v\}$ соответствуют модулям a_i , размещаемым в позициях: $f_v=V \rightarrow A$, $a_i=f_v(v_i)$. В состав $U1$ входят ребра, исходящие из вершин множества V , а в состав $U2$ входят ребра, входящие в вершины множества V . Каждая пара вершин $(v_i, v_j) \in V$ связана парой ориентированных ребер (u_{ij}, u_{ji}) . Ребро u_{ij} , исходит из v_i , и входит в v_j ; ребро u_{ji} , исходит из v_j , и входит в v_i .

Работа конструктивного алгоритма размещения элементов в позиции КП заключается в следующем.

В состав разрабатываемого конструктивным алгоритмом вектора M_{kw} , (k – номер агента, w – номер списка) входят все вершины, принадлежащие множеству V графа G . Формирование вектора M_{kw} производится на графе G пошагово.

На каждом шаге t конструктивного алгоритма, начиная с первого, выбирается на основе эвристических правил вершина v_i (элемент $a_j \in A$, который помещается в позицию t формируемого вектора M_{kw} т.е. $m_t=v_i$). После выполнения $t-1$ шагов при построении вектора агентом z_k формируется множество еще не размещенных вершин $V2_k(t-1)$, таких, что каждая из вершин $v_j \in V2_k(t-1)$ может быть добавлена на шаге t в формируемый вектор $M_{kw}(t)$, и множество уже размещенных вершин $V1_k(t-1) \in V$, $V1_k(t-1) \cup V2_k(t-1) = V$.

Для каждой $v_j \in V2_k(t-1)$ рассчитывается параметр $\psi_{kj}(t)$ – суммарный уровень феромона, накопленный на множестве ориентированных ребер графа G , входящих в v_j и исходящих из вершин множества $V1_k(t-1)$ (т.е. связывающих v_j с $V1_k(t-1)$).

Значение параметра $\psi_{kj}(t)$ рассматривается в качестве степени $\delta_{kj}(t)$ привлекательности вершины $v_j \in V2_k(t-1)$ на шаге t , для включения в формируемый маршрут $M_{kw}(t)$. Вероятность P_{kj} включения вершины $v_j \in V2_k(t-1)$ в формируемый маршрут $M_{kw}(t)$ определяется следующим соотношением:

$$P_{kj} = \delta_{kj}(t) / \sum_j (\delta_{kj}(t)), \quad (\forall j/v_j \in V2_k(t-1)).$$

Агент z_k с вероятностью P_{kj} выбирает одну из вершин $v_j \in V2_k(t-1)$, которая включается в вектор M_{kw} .

После распределения всех элементов по позициям рассчитывается оценка размещения ζ_{kw} задаваемого маршрутом M_{kw} .

Коллективная эволюционная память (КЭП) муравьиной колонии представляет число ψ , показывающее полезность фрагмента при построении решений на предыдущих итерациях алгоритма. Следует отметить, что решением задачи размещения является вектор $M_k = \langle m_i | i=1, \dots, n_m \rangle$, где m_i – номер элемента, размещаемого в позиции p_i . Для фиксации взаимного расположения элементов m_i на графе G вектор M_k трансформируется в маршрут по правилу – пара соседних в векторе M_k вершин m_i и m_{i+1} связана ребром $u_{i+1} = (m_i, m_{i+1})$ направленным от m_i к m_{i+1} . Таким образом, вектор $M_k = \langle m_1, m_2, \dots, m_i, m_{i+1}, \dots, m_k \rangle$ трансформируется в маршрут $M_k^* = (m_1, u_{12}, m_2, \dots, m_i, u_{i+1}, m_{i+1}, \dots, m_{k-1}, u_{k-1,k}, m_k)$, отражающий решение на графе $G=(V,U)$, где все ребра в M_k^* ориентированные: каждое ребро u_{ij+1} маршрута M_k^* исходит из вершины m_i и входит в вершину m_{i+1} , где m_i – номер вершины графа G , выбранной агентом z_i , на шаге t ($i=t$) при построении маршрута, m_{i+1} – номер вершины графа G , выбранной агентом z_k , на шаге $t+1$. Маршрут M_k^* является адекватным представлением решения задачи размещения на графе G .

В качестве коллективной эволюционной памяти (КЭП) алгоритмом используется матрица $A = \|\delta_{ij}\|_{m \times m}$, в которой для каждого ребра графа G хранятся две интегральные оценки степени пригодности δ_{ij} и δ_{ji} , в зависимости от ориентации ребра графа графе $G=(V,U)$. Каждая пара вершин $(v_i, v_j) \in V$ связана парой ориентированных ребер (u_{ij}, u_{ji}) . Ребро u_{ij} , исходит из v_i , и входит в v_j ; ребро u_{ji} , исходит из v_j , и входит в v_i .

Итерационный процесс поиска решений осуществляется мультипопуляцией, включающей четыре виртуальных субпопуляции решений эволюционирующих параллельно.

Отличительной особенностью используемого ко-эволюционного подхода является то, что четыре субпопуляции решений фактически являются виртуальными и эволюционируют на базе одной популяции агентов $Z=\{z_k|k=1,\dots,n_k\}$. Процесс ко-эволюции реализуется одной популяцией агентов Z путем последовательного формирования и слияния виртуальных субпопуляций решений $S1-S4$ в одну популяцию.

Все стратегии используют одно и то же множество позиций, но каждая стратегия $S1$ – индивидуальный список позиций S_i .

Взаимодействие между ко-эволюционирующими популяциями решений, решающими параллельно одну и ту же задачу оптимизации, реализуется в форме сотрудничества.

В течении заданного числа итераций производится независимое развитие четырех виртуальных субпопуляций решений, используя общую структуру коллективной эволюционной памяти (КЭП) в виде полного графа поиска решений $G=(V,U)$. Периодически по накопленным в процессе слияния результатов работы виртуальных субпопуляций производится модификация общей КЭП. На основе интегральных результатов виртуальных субпопуляций после фиксированного числа итераций модифицируются веса множества ребер полного графа G .

Моделирование поведения агентов в задаче размещения связано с распределением феромона на ребрах графа поиска решений, феромон откладывается в матрицу КЭП Δ только после полного формирования популяцией агентов всех субпопуляций решений и расчета оценок на текущей итерации. Для накопления феромона, откладываемого агентами субпопуляций за одну итерацию, используется вспомогательная накопительная матрица $KЭП-\Delta 1=||\delta I_{ij}||_{m \times m}$. Другими словами на текущей итерации агенты каждой раз после построения очередного решения откладывают феромон в накопительную $KЭП-\Delta 1$, а после завершения итерации количество феромона, накопленное агентами всех популяций в $KЭП-\Delta 1$, откладывается в $KЭП \Delta$.

Каждая итерация l ко-эволюционного алгоритма включает начальный этап, основную стадию и два заключительных этапа итерации.

На начальном этапе муравьиного алгоритма размещения:

- ◆ фиксируются входные данные;
- ◆ задаются начальные значения управляющих параметров;
- ◆ строится модель ДРП;
- ◆ формируются 4 списка позиций на ДРП.

Следом за начальным этапом выполняются процедуры основной стадии.

На основной стадии каждый агент $z_k \in Z$ поочередно для каждого списка S_w выполняет следующие действия:

- ◆ приводится в начальное состояние память агента;
 - ◆ обнуляются вспомогательные матрицы;
 - ◆ конструктивным алгоритмом на графе поиска решений G строится маршрут M_{kw} для списка S_w ;
 - ◆ M_{kw} трансформируется в M_{kw}^* ;
 - ◆ рассчитывается оценка ζ_{kw} решения – M_{kw}^* ;
 - ◆ рассчитывается степень пригодности $\psi_{kw} = \alpha \zeta_{kw}$ элементов маршрута M_{kw}^* ,
- где α – коэффициент пропорциональности;

♦ определяется множество ребер U_{kw} графа G , входящих в состав маршрута M_{kw}^* ;

♦ значения элементов накопительной матрицы ΔI КЭП1, соответствующих ребрам U_{kw} графа G , входящим в состав маршрута M_{kw}^* , увеличиваются на величину ψ_{kw} .

На первом заключительном этапе итерации интегральные оценки δI_{ij} , накопленные элементами промежуточной матрицы ΔI КЭП1, добавляются к соответствующим добавляются к соответствующим оценка элементов основной матрицы $KЭП - \Delta$.

1. Задаются начальные значения управляющих параметров:

N_s – число субпопуляций и стратегий;

N_k – число агентов;

N_l – число итераций;

N – число позиций.

2. Строится модель ДРП – $D = \|d_{ij}\|_{n \times m}$.

3. Задается расположение позиций в ячейках d_{ij} ДРП.

Задаются координаты (x_i, y_i) расположения позиций p_i на ДРП.

Устанавливается биективное соответствие $f_q: P \rightarrow Q$, где $Q = \{(x_i, y_i) | i=1, \dots, n_m\}$.
 $f_q(p) = (x_i, y_i)$.

4. Для выбранного расположения позиций на КП на основе механизмов волнового алгоритма формируются четыре списка позиций $S_1 - S_4$.

5. Формируется полный граф поиска решений $G = (V, U)$.

Формируются матрицы коллективной эволюционной памяти: основная $\Delta = \|\delta_{ij}\|_{m \times m}$ и накопительная $\Delta I = \|\delta I_{ij}\|_{m \times m}$.

6. Элементам основной матрицы КЭП $\Delta = \|\delta_{ij}\|$ присваивается начальное значение \emptyset .

(Основная стадия)

7. $l=1$ (номер итерации).

8. Обнуляется накопительная матрица КЭП $\Delta I = \|\delta I_{ij}\|_{m \times m}$, ($\forall \delta I_{ij} \in \Delta I$) [$\delta I_{ij} = 0$].

9. $k=1$ (номер агента).

10. $w=1$ (номер списка).

11. Приводится в начальное состояние память агента z_k : $V1_{kw}(0) = \emptyset$; $V2_{kw}(0) = V$.

12. Конструктивным алгоритмом на графе поиска решений $G = (V, U)$ для списка S_w агентом $z_k \in Z$ строится маршрут M_{kw} .

13. M_{kw} трансформируется в M_{kw}^* .

14. Рассчитывается оценка ζ_{kw} решения M_{kw}^* .

15. Рассчитывается степень пригодности ψ_{ij} элементов маршрута M_{kw}^* , $\psi_{kw} = \alpha \zeta_{kw}$, где α – коэффициент пропорциональности.

16. Значения элементов промежуточной матрицы ΔI КЭП1, соответствующих ребрам графа G , входящим в состав маршрута M_{kw}^* , увеличиваются на величину ψ_{kw} . ($\forall u_{ij} \in M_{kw}^*$) [$\delta I_{ij} = \delta I_{ij} + \psi_{kw}$].

17. Если $w < N_s$, то $w = w + 1$ и переход к 12, иначе переход к 18.

18. Если $k < N_k$, то $k = k + 1$ и переход к 10, иначе переход к 15.

(Первый заключительный этап итерации)

19. Интегральные оценки δI_{ij} , накопленные элементами промежуточной матрицы ΔI КЭП1, добавляются к соответствующим оценкам δ_{ij} элементов основной матрицы $KЭП - \Delta$. ($\forall u_{ij} \in U$) [$\delta_{ij} = \delta_{ij} + \delta I_{ij}$].

(Второй заключительный этап)

20. Выполняется процедура испарения феромона на всех ребрах U графа G , в соответствии с формулой: ($\forall u_{ij} \in U$) [$\delta_{ij}(u_{ij}) = \rho \delta_{ij}(u_{ij})$], где δ_{ij} количество феромона на ребре u_{ij} графа G , ρ – коэффициент обновления.

21. Если $l < N_b$, то $l = l + 1$ и переход к 8, иначе переход к 21.
22. Конец работы алгоритма.

3. Экспериментальные исследования. Ко-эволюционный алгоритм размещения реализован в виде программы **CoP**. Программа CoP написана на языке C++ в среде Microsoft Visual Studio 2010 для ОС Windows. Эксперименты проводились по аналогии с известным методом (РЕКО) [17–20].

Для экспериментальных исследований разработанной программы размещения были применены схемы набора РЕКУ с известным оптимумом F_{opt} : Ex.1 на 30 блоков, Ex.2 – 60, Ex.3 – 90, Ex.4 – 120, Ex.5 – 150. Для сравнения были выбраны известные алгоритмы размещения: Dragon v2.20 [17], Capo v.8., mPL v.2.0, mPG v1.0 и QPlace v.5.1 [18–21].

Для сравнительной оценки эффективности тестируемых алгоритмов, рассчитывался параметр: наилучшая длина соединений (для РЕКО) или (для G-РЕКУ и РЕКУ) к полученной длине соединений. Оценкой качества служит величина $\xi = F_{opt}/F$ – «степень качества», где F – оценка полученного решения. Ни один из алгоритмов размещения при исследовании не достиг значения коэффициента качества близкого к 1. В таблице 8. приведены полученные значения показателя степень качества ряда известных алгоритмов и алгоритма **CoP**.

Таблица 8

Полученные значения

Тест	Dragon	Capo	mPL	mPG	Qplace	CoP
Ex.1	0.71	0.73	0.81	0.71	0.73	0.95
Ex.2	0.72	0.72	0.84	0.72	0.78	0.95
Ex.3	0.81	0.8	0.83	0.81	0.82	0.96
Ex.4	0.83	0.84	0.85	0.83	0.81	0.98
Ex.5	0.82	0.81	0.86	0.82	0.84	0.97

Степень качества у разработанной программы **CoP** на 7% выше, чем у программ Dragon, Capo, mPL, mPG и Qplace. Временная сложность алгоритма – $O(n^2)$.

На основе результатов экспериментальных исследований была построена средняя зависимость степени качества от числа итераций (рис. 6) и от размера популяции (рис. 7).

Исследования показали, что число итераций, при которых популяция объемом в 100 агентов находил лучшее решение, лежит в пределах 122-132.

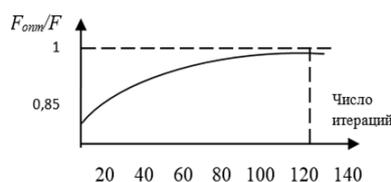


Рис. 6. Зависимость качества решений алгоритма от числа итераций

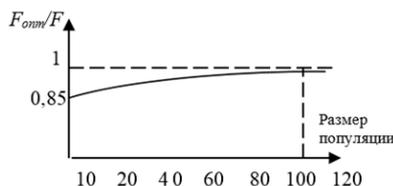


Рис. 7. Зависимость качества решений алгоритма от размера популяции

Тестирование производилось на бенчмарках 19s, PrimGA1, PrimGA2 [17, 21]. Результаты по сравнению с существующими алгоритмами улучшены на 7-8%. Вероятность получения глобального оптимума составила 0.96. В среднем решения отличаются от оптимального менее, чем на 1.5%. Временная сложность алгоритма при фиксированных значениях размера популяции и количества генераций составляет $O(n)$. Общая временная сложность гибридного алгоритма составляет $O(n^2) - O(n^3)$.

Заключение. Разработана новая методология и метод размещения элементов СБИС, отличающиеся тем, решение задачи размещения основывается на использовании фиксированного порядка выбора позиций, ориентированного на эффективное решение задачи размещения, и эвристической процедуры распределения элементов по позициям, позволяющие снизить общую трудоемкость, и повысить качество решения.

Процесс формирования списка позиций на коммутационном поле осуществляется с использованием механизмов волнового алгоритма. В основу выбора окончательного списка положен принцип построения маршрута с минимальной оценкой суммарной линейной длины расстояний между позициями маршрута. Для решения задачи размещения разработан поисковый алгоритм на основе модифицированного метода муравьиной колонии.

В работе процесс ко-эволюции реализован на основе взаимодействия субпопуляций, отличающихся стратегиями поиска. Разработана новая архитектура ко-эволюционного алгоритма размещения на основе модернизированной парадигмы муравьиного алгоритма, отличающаяся тем, что субпопуляции решений фактически являются виртуальными, а процесс ко-эволюции реализуется одной популяцией агентов путем последовательного формирования и слияния виртуальных субпопуляций решений в одну популяцию, что позволяет обеспечить более высокую вероятность локализации глобального экстремума задачи.

В работе решение задачи размещения направлено на повышение трассируемости посредством минимизации ресурсов, требуемыми для реализации соединений. Существенный вклад в минимизацию пространственной и временной сложности поисковой процедуры внесли: использование субпопуляциями общей эволюционной памяти, общего графа поиска решений, формировании единой интерпретации решения в виде маршрута на полном ориентированном графе с бинарными ориентированными ребрами. Это позволило повысить вероятность локализации глобального экстремума задачи. Для решения задачи размещения разработан поисковый алгоритм на основе модифицированного метода муравьиной колонии.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Тучин А.В., Бормонтов Е.Н., Пономарев К.Г. Введение в системы автоматизированного проектирования интегральных микросхем. – Воронеж: Издательский дом ВГУ, 2017. – 110 с.
2. Казеннов Г.Г. Основы проектирования интегральных схем и систем. – М.: Бином. Лаборатория знаний, 2010. – 295 с.
3. Лебедев Б.К., Лебедев О.Б., Лебедев В.Б. Методы, модели и алгоритмы размещения. – Ростов-на-Дону: Изд-во ЮФУ, 2015. – 150 с.
4. Лебедев Б.К., Лебедев О.Б. Гибридный биоинспирированный алгоритм размещения базовых стандартных библиотечных элементов при проектировании топологии полужаказной СБИС // Известия ЮФУ. Технические науки. – 2019. – № 3. – С. 97-110.
5. Карпенко А.П. Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой: учеб. пособие. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2016. – 448 с.
6. Воробьева Е.Ю., Карпенко А.П., Селиверстов Е.Ю. Ко-гибридизация алгоритмов роя частиц // Наука и образование. – 2012. – № 4. – С. 28-35.
7. Лебедев О.Б. Распределение ресурсов на основе гибридных моделей роевого интеллекта // Научно-технический вестник информационных технологий, механики и оптики. – 2017. – № 6. – С. 1063-1073.

8. *Roze K., Radchenko D.* Платформа Synopsys для проектирования цифровых систем – новый уровень технологий проектирования СнК // *Электроника*. – 2018. – № 2. – С. 122-141.
9. *Лебедев О.Б.* Модели адаптивного поведения муравьиной колонии в задачах проектирования. – Таганрог: Изд-во ЮФУ, 2013. – 199 с.
10. *Sur-Kolay S., Bishnu A., Das S., Nandy S.C., Bhattacharjee S.* FPGA placement using space-filling curves: Theory meets practice // *ACM Trans. Embed. Comput. Syst.* – 2009. – Vol. 9, № 2. – P. 1-23.
11. *Yang X., Choi B.-K., Sarrafzadeh M.* Routability-driven white space allocation for fixed-die standard-cell placement // *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*. – 2003. – Vol. 22 (4). – P. 410-419.
12. *Лебедев Б.К., Лебедев О.Б., Жиглатый А.А.* Размещение элементов СБИС на основе моделей роевого интеллекта // *Проблемы разработки перспективных микро- и нанoeлектронных систем: Сб. трудов под общ. ред. академика РАН А.Л. Стемповского*. – М.: ИПИМ РАН, 2020. – С. 118-126.
13. *Cong J., Romesis M., Xie M.* Optimality, Scalability and Stability Study of Partitioning and Placement Algorithms // *Proc. of the International Symposium on Physical Design*. – Monterey, CA, 2003. – P. 88-94.
14. *Sherwani N.A.* Algorithms for VLSI Physical Design Automation. – 3rd ed. – Kluwer Academic Publisher. USA, 2013. – 572 p.
15. *Mourelle M.* Swarm intelligent systems. – Berlin: Heidelberg: Springer Verlag, 2006. – 217 p.
16. *Норенков И.П.* Основы автоматизированного проектирования: учебник. – М.: Изд-во МГТУ имени Н.Э. Баумана, 2006. – 336 с.
17. *Wang M., Yang X., Sarrafzadeh M.* Dragon2000: Standard-cell Placement Tool for Large Industry Circuits // *ICCAD 2000*. – P. 260-263.
18. *Caldwell A.E., Kahng A.B., Markov I.L.* Can Recursive Bisection Alone Produce Routable Placements? // *DAC 2000*. – P. 477-482.
19. Cadence design systems, Inc. QPlace version 5.1.55, compiled on 10/25/1999. Envisia ultra-placer reference.
20. IBM-PLACE 2.0 benchmark suits. – <http://er.cs.ucla.edu/benchmarks/-ibm-place2/bookshelf/ibm-place2-all-bookshelf-nopad.tar.gz>.
21. *Adya S.N.* ISPD02 IBM-MS Mixed-size Placement Benchmarks. – <http://vlsicad.eecs.umich.edu/BK/ISPD02bench/>.

REFERENCES

1. *Tuchin A.V., Bormontov E.N., Ponomarev K.G.* Vvedenie v sistemy avtomatizirovannogo proektirovaniya integral'nykh mikroskhem [Introduction to computer-aided design systems for integrated circuits]. Voronezh: Izdatel'skiy dom VGU, 2017, 110 p.
2. *Kazennov G.G.* Osnovy proektirovaniya integral'nykh skhem i system [Basics of designing integrated circuits and systems]. Moscow: Binom. Laboratoriya znaniy, 2010, 295 p.
3. *Lebedev B.K., Lebedev O.B., Lebedev V.B.* Metody, modeli i algoritmy razmeshcheniya [Methods, models and placement algorithms]. Rostov-on-Don: Izd-vo YuFU, 2015, 150 p.
4. *Lebedev B.K., Lebedev O.B.* Gibridnyy bioinspirirovanny algoritm razmeshcheniya bazovykh standartnykh biblioteknykh elementov pri proektirovanii topologii poluzakaznoy SBIS [Hybrid bioinspired algorithm for placing basic standard library elements in the design of the topology of a semi-custom VLSI], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2019, No. 3, pp. 97-110.
5. *Karpenko A.P.* Sovremennyye algoritmy poiskovoy optimizatsii. Algoritmy, vdokhnovlennyye prirodoy: ucheb. posobie [Modern search engine optimization algorithms. Algorithms inspired by nature: textbook]. Moscow: Izd-vo MGTU im. N.E. Baumana, 2016, 448 p.
6. *Vorob'eva E.Yu., Karpenko A.P., Seliverstov E.Yu.* Ko-gibridizatsiya algoritmov roya chastits [Co-hybridization of particle swarm algorithms], *Nauka i obrazovanie* [Science and Education], 2012, No. 4, pp. 28-35.
7. *Lebedev O.B.* Raspredelenie resursov na osnove gibridnykh modeley roevogo intellekta [Resource distribution based on hybrid models of swarm intelligence], *Nauchno-tekhnicheskiy vestnik informatsionnykh tekhnologiy, mekhaniki i optiki* [Scientific and technical bulletin of information technologies, mechanics and optics], 2017, No. 6, pp. 1063-1073.

8. *Roze K., Radchenko D.* Platforma Synopsys dlya proektirovaniya tsifrovyykh sistem – novyy uroven' tekhnologiy proektirovaniya SnK [Synopsys platform for designing digital systems – a new level of SoC design technologies], *Elektronika* [ELECTRONICS], 2018, No. 2, pp. 122-141.
9. *Lebedev O.B.* Modeli adaptivnogo povedeniya murav'inoy kolonii v zadachakh proektirovaniya [Models of adaptive behavior of an ant colony in design problems]. Taganrog: Izd-vo YuFU, 2013, 199 p.
10. *Sur-Kolay S., Bishnu A., Das S., Nandy S.C., Bhattacharjee S.* FPGA placement using space-filling curves: Theory meets practice, *ACM Trans. Embed. Comput. Syst.*, 2009, Vol. 9, No. 2, pp. 1-23.
11. *Yang X., Choi B.-K., Sarrafzadeh M.* Routability-driven white space allocation for fixed-die standard-cell placement, *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 2003, Vol. 22 (4), pp. 410-419.
12. *Lebedev B.K., Lebedev O.B., Zhiglaty A.A.* Razmeshchenie elementov SBIS na osnove modeley roevogo intellekta [Placement of VLSI elements based on swarm intelligence models], *Problemy razrabotki perspektivnykh mikro- i nanoelektronnykh sistem: Sb. trudov pod obshch. red. akademika RAN A.L. Stempkovskogo* [Problems of development of promising micro- and nanoelectronic systems. Collection of works under the total. ed. Academician of the Russian Academy of Sciences A.L. Stempkovsky]. Moscow: IPPM RAN, 2020, pp. 118-126.
13. *Cong J., Romesis M., Xie M.* Optimality, Scalability and Stability Study of Partitioning and Placement Algorithms, *Proc. of the International Symposium on Physical Design*. Monterey, CA, 2003, pp. 88-94.
14. *Sherwani N.A.* Algorithms for VLSI Physical Design Automation. 3d ed. – Kluwer Academic Publisher. USA, 2013. – 572 p.
15. *Mourelle M.* Swarm intelligent systems. Berlin: Heidelberg: Springer Verlag, 2006, 217 p.
16. *Norenkov I.P.* Osnovy avtomatizirovannogo proektirovaniya: uchebnik [Fundamentals of computer-aided design: textbook]. Moscow: Izd-vo MGTU imeni N.E. Bauman, 2006, 336 p.
17. *Wang M., Yang X., Sarrafzadeh M.* Dragon2000: Standard-cell Placement Tool for Large Industry Circuits, *ICCAD 2000*, pp. 260-263.
18. *Caldwell A.E., Kahng A.B., Markov I.L.* Can Recursive Bisection Alone Produce Routable Placements?, *DAC 2000*, pp. 477-482.
19. Cadence design systems, Inc. QPlace version 5.1.55, compiled on 10/25/1999. Envisia ultra-placer reference.
20. IBM-PLACE 2.0 benchmark suits. Available at: <http://er.cs.ucla.edu/benchmarks/-ibm-place2/bookshelf/ibm-place2-all-bookshelf-nopad.tar.gz>.
21. *Adya S.N.* ISPD02 IBM-MS Mixed-size Placement Benchmarks. Available at: <http://vlsicad.eecs.umich.edu/BK/ISPD02bench/>.

Статью рекомендовал к опубликованию д.т.н., профессор А.Г. Коробейников.

Лебедев Олег Борисович – Южный федеральный университет; e-mail: lebedev.ob@mail.ru; г. Таганрог, Россия; тел.: 89085135512; кафедра систем автоматизированного проектирования; доцент.

Жиглатый Артемий Александрович – e-mail: artemiy.zhiglaty@gmail.com; тел.: 89185916819; кафедра систем автоматизированного проектирования; аспирант.

Lebedev Oleg Borisovich – Southern Federal University; e-mail: lebedev.ob@mail.ru; Taganrog, Russia; phone: +79085135512; the department of computer aided design; associate professor.

Zhiglatiy Artemy Alexandrovich – e-mail: artemiy.zhiglaty@gmail.com; phone: +79185916819; the department of computer aided design; graduate student.