Раздел II. Алгоритмы обработки информации

УДК 004.896 DOI 10.18522/2311-3103-2022-4-113-121

О.Б. Лебедев, А.А. Жиглатый

БИОИНСПИРИРОВАННЫЙ АЛГОРИТМ РЕШЕНИЯ ИНВАРИАНТНЫХ ГРАФОВЫХ ЗАДАЧ*

Предлагается биоинспирированный метод решения набора инвариантных комбинаторно-логических задач на графах: формирования паросочетания графа, выделения внутренне-устойчивого множества вершин, выделения клики графа. Описывается модифицированная парадигма муравьиной колонии использующая, в отличие от канонического метода, механизмы формирования решений на модели пространства поиска в виде звездного графа. Задача формирования в графе внутренне-устойчивого множества вершин может быть сформулирована, как задача разбиения. На начальном этапе на всех ребрах звездного графа H откладывается одинаковое (небольшое) количество феромона ξ/m, где $m=|E|$. Процесс поиска решений итерационный. Каждая итерация l включает три этапа. Агенты обладают памятью. На каждом шаге t в памяти агента a_k имеется количество феромона $\phi_j(t)$, отложенного на каждом ребре графа H. На первом этапе каждый агент a_k популяции конструктивным алгоритмом находит решение $U^r{}_{0k}$, рассчитывает оценку решения $\xi_k(U^r{}_{0k})$ и значение степени пригодности полученного агентом решения φ_k (количество феромона, соответствующее оценке). На втором этапе, после полного формирования всеми агентами решений на текущей итерации, феромон ω_j, накопленный в j-ой ячейке в буферном массиве КЭП$_б$, добавляется в каждую j-ю ячейку основного массива $Q_2=\{q_j|j=1,2,…,m\}$ коллективной эволюционной памяти КЭП$_о$. На третьем этапе происходит общее испарение феромона на множестве ребер E звездного графа H. Временная сложность алгоритма, полученная экспериментальным путем, совпадает с теоретическими исследованиями и для рассмотренных тестовых задач составляет $O(n^2)$.

Паросочетание; внутренне-устойчивое множество; клика; адаптация; модификация; муравьиная колония; модели пространства поиска; звездный граф.

O.B. Lebedev, A.A. Zhiglatiy

BIOINSPIRED ALGORITHM FOR SOLVING INVARIANT GRAPH PROBLEMS

A bioinspired method for solving a set of invariant combinatorial-logical problems on graphs is proposed: the formation of a graph matching, the selection of an internally stable set of vertices, and the selection of a graph clique. A modified paradigm of the ant colony is described, which uses, in contrast to the canonical method, the mechanisms for generating solutions on the search space model in the form of a star graph. The problem of forming an internally stable set of vertices in a graph can be formulated as a partitioning problem. At the initial stage, the same (small) amount of pheromone ξ/m, where $m=|E|$, is deposited on all edges of the star graph H. The process of finding solutions is iterative. Each iteration l includes three stages. Agents have memory. At each step t, the memory of the agent a_k contains the amount of pheromone $\phi_j(t)$ deposited on each edge of the graph H. At the first stage, each agent a_k of the population uses a constructive algorithm to find the solution $U^r{}_{0k}$, calculates the estimate of the solution $\xi_k(U^r{}_{0k})$ and the value of the degree of suitability of the solution obtained by the agent φ_k (the amount of phero-

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

mone corresponding to the estimate). At the second stage, after the complete formation of solutions by all agents at the current iteration, the pheromone ω_j accumulated in the j-th cell in the $CEP_б$ buffer array is added to each j-th cell of the main array $Q_2=\{q_j|j=1,2,\ldots,m\}$ of the CEP_0 collective evolutionary memory. At the third stage, the general evaporation of the pheromone occurs on the set of edges E of the star graph H. The time complexity of the algorithm, obtained experimentally, coincides with theoretical studies and for the considered test problems is $O(n^2)$.

Matching; internally stable set; clique; adaptation; modification; ant colony; search space models; star graph.

Введение. Задача определения паросочетаний – одна из фундаментальных задач в области математической оптимизации. Подмножество ребер $M \subseteq U$ графа $G=(V,U)$ называется паросочетанием в графе G, если каждой вершине $v_i \in V$ инцидентно не более одного ребра из M. **Максимальное паросочетание** – это такое паросочетание M в графе G, которое не содержится ни в каком другом паросочетании этого графа, то есть к нему невозможно добавить ни одно ребро, которое бы являлось несмежным ко всем рёбрам паросочетания [1]. Наибольшее паросочетание (или максимальное по размеру паросочетание) [2] – это такое паросочетание, которое содержит максимальное количество рёбер. **Совершенным паросочетанием** называется паросочетание, в котором участвуют все вершины графа.

Алгоритмы построения паросочетания применяются в задачах о назначении, планирования и размещения при проектировании топологии СБИС, покрытия, транспортных задачах, при проектировании инженерных сетей, коммуникаций, построения систем поддержки принятия решений в неопределенных условиях и т.п. Задачи такого типа относятся к переборным задачам с экспоненциальной временной сложностью. В этой связи разрабатывают различные эвристики для построения алгоритмов с полиномиальной временной сложностью. Существуют алгоритмы определения паросочетаний в графе, основанные на использовании потоков в сетях [1, 2], имитационного моделирования [3], генетического поиска [4] и других эвристиках, которые обеспечивают приемлемые результаты при решении задач малой и средней сложности. Часто эта процедура используется в итерационных структурах. Это предъявляет повышенные требования к качеству и времени решения задачи нахождения максимального паросочетания. Возникшие потребности в решении задач большой и очень большой размерности является побудительным мотивом исследований и разработок новых эффективных алгоритмов. Анализ литературы показывает, что наиболее успешными в этих условиях являются математические методы, в которых заложены принципы природных механизмов принятия решений [4–6]. К таким методам можно отнести, прежде всего, методы моделирования отжига [7], методы эволюционного моделирования и генетические алгоритмы [8], эволюционной адаптации [9], алгоритмы роевого интеллекта [10] и муравьиные алгоритмы (Ant Colony Optimization – ACO) [11, 12]. Идея муравьиного алгоритма – моделирование поведения муравьёв, связанного с их способностью быстро находить кратчайший путь от муравейника к источнику пищи. Основу поведения муравьиной колонии составляет самоорганизация, обеспечивающая достижения общих целей колонии на основе низкоуровневого взаимодействия, благодаря которому, в целом, колония представляет собой разумную многоагентную систему.

Для повышения эффективности, усиления сходимости алгоритма и способности выхода из локальных оптимумов, предложен подход к построению алгоритма паросочетания на основе модели адаптивного поведения муравьиной колонии, **отличающийся** от канонического тем, что в качестве модели пространства поиска решений используется звездный граф.

Предложенный подход обеспечивает более широкий обзор пространства решений и более высокую вероятность локализации глобального экстремума задачи.

В работе излагается методика решения задачи нахождения максимального паросочетания в графе, и родственных ей задач раскраски графа и выделения клик в графе, основанная на моделировании адаптивного поведения муравьиной колонии.

1. Постановка задачи нахождения паросочетания в графе. Пусть дан двудольный граф $G=(V,U)$ (рис. 1). $V=\{v_i|i=1,2,\ldots,n\}$. $U=\{u_j|j=1,2,\ldots,m\}$. На рис. 1 n=10, m=11.

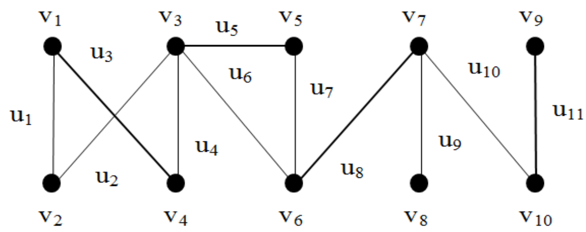

Рис. 1. Пример графа

Паросочетание P такого графа определяется как множество попарно несмежных рёбер, то есть рёбер, не имеющих общих вершин [1]. Например: для графа G (рис.1) паросочетание $P=\{u_3, u_5, u_8, u_{11}\}$. Будем говорить, что вершина графа G принадлежит паросочетанию, понимая под этим, что она инцидентна некоторому ребру паросочетания.

Построим граф $G_r=(U^r,V^r)$ – реберный для графа G. Вершины U^r графа G_r – соответствуют рёбрам U графа G. Пара вершин u^r_i, $u^r_j)$ в графе G_r связаны ребром в том и только в том случае, если в графе G пара рёбер (u_i, u_j) смежны, т.е. инцидентны одной вершине [1].

Для примера, представленного на рис. 1, реберный граф G_r графа G имеет вид, представленный на рис. 2.

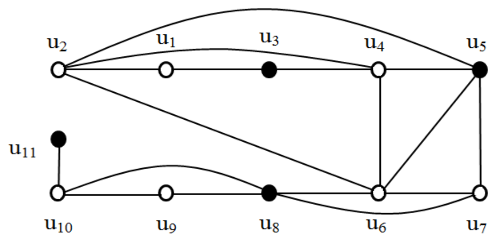

Рис. 2. Реберный граф G_r графа G

Множество $U^r_0\subset U^r$ вершин графа $G_r=(U^r,V^r)$ называется внутренне устойчивым, если любые две вершины $u^r_i\in U^r_0$ и $u^r_j\in U^r_0$ не являются смежными [2]. Максимальное число вершин во внутренне устойчивом множестве графа U^r называется числом внутренней устойчивости и обозначается как $\alpha(G_r)$. Иногда число внутренней устойчивости называют также числом независимости графа G_r [3].

Подмножество вершин $U^r_0=\{u_3, u_5, u_8, u_{11}\}$ (см. рис. 2) графа $G_r=(U^r,V)$ является внутренне устойчивым, т.к. любые две вершины подмножества U^r_0 не смежны. Таким образом, паросочетанию в графе G соответствует внутренне-устойчивое подмножество реберного графа G_r графа G. Максимальному по мощности паросочетанию в графе G соответствует предельное внутренне-устойчивое подмножество (содержащее наибольшее число вершин) реберного графа G_r.

Рассмотрим задачу о *клике*. Кликой графа G называется максимальное по включению множество $V_0 \subset V$ вершин графа G, любые две из которых являются смежными. Другими словами, клика графа есть подмножество его вершин, такое, что между каждой парой вершин этого подмножества существует ребро.

Максимальная клика – это клика, которая не может быть расширена путём включения дополнительных смежных вершин, то есть нет клики большего размера, включающей все вершины данной клики. Наибольшая клика – это клика максимального размера для данного графа [4].

Дополнение графа (обратный граф) – граф Go, имеющий то же множество вершин, что и заданный граф G, но в котором две несовпадающие вершины смежны тогда и только тогда, когда они не смежны в G [2].

Нетрудно видеть, что при переходе от графа G к его дополнению G_o каждая клика в G переходит в независимое множество в G_o. Отсюда следует, что задача выделения клики в графе G сводится к задаче выделения независимого множества вершин в графе G_o.

Таким образом, в основе процедур построения максимального паросочетания и выделения в графе клик лежит одна общая процедура формирования в графе $G(V, U)$ внутренне-устойчивого множества вершин. Задачу нахождения максимальных независимых множеств иногда называют «упаковкой вершин».

2. Механизмы выделения в графе независимого подмножества вершин на основе моделирования адаптивного поведения муравьиной колонии. Пусть дан граф $G(V,U)$, где V – множество вершин, $|V| = n$, U – множество ребер, $|U|=m$. Построим граф $G_r=(U^r,V^r)$ – реберный для графа G. Множество вершин $U^r=\{u^r_j|i=1,2,\dots,m\}$ графа G_r соответствует множеству ребер U графа G. Множество ребер $V^r=\{v^r_i|i=1,2,\dots,n\}$ графа G_r соответствует множеству вершин V графа G.

Сформулируем задачу формирования в графе $G_r=(U^r,V^r)$ внутренне – устойчивого множества вершин $U^r_0 \subset U^r$ как задачу разбиения.

Необходимо разбить множество U^r графа G_r на два непустых и непересекающихся подмножества U^r_0 и U^r_2, таких, что любые две вершины $u^r_i \in U^r_0$ и $u^r_j \in U^r_0$ не являются смежными, $U^r_0 \cup U^r_2 = U^r$, $U^r_0 \cap U^r_2=\varnothing$. Пусть $|U^r_0|=n_0$, $|U^r_2|=n_2$, $n_0 + n_2=m$. Критерий оптимизации – число вершин $F=n_0$ в подмножестве U^r_0. Цель оптимизации – максимизация критерия F.

Для поиска решения задачи формируется звездный граф поиска решений $H(U^r \cup S_0, E)$, определяемый корневой вершиной S_0, который состоит из всех ребер E, связывающих корневую вершину S_0 с множеством всех вершин U^r (рис. 3). Корневая вершина S_0 связывается со всеми вершинами множества U^r.

В процессе решения участвует коллектив агентов. Каждый из агентов a_k формирует свое решение – свое подмножество $U^r_{0k} \subset U^r$ вершин звездного графа H, такое, что любые две вершины $u^r_i \in U^r_{0k}$ и $u^r_j \in U^r_{0k}$ в графе $G_r=(U^r,V^r)$ не являются смежными. k – номер агента.

Обозначим множество ребер звездного графа H, связывающих вершины сформированного внутри устойчивого подмножества U^r_{0k} с корневой вершиной S_0 как E_{0k}, $E_{0k} \subset E$.

Моделирование поведения агентов в процессе формирования каждым из них своего подмножества U^r_{0k} связано с распределением и учетом количества феромона на множестве E_{0k} ребер графа H.

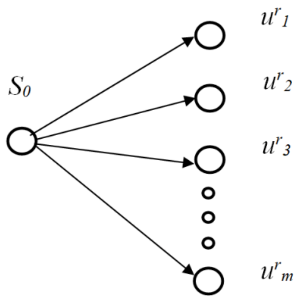

Рис. 3. Звездный граф H

На начальном этапе на всех ребрах звездного графа H откладывается одинаковое (небольшое) количество феромона ξ/m, где $m=|E|$.

Процесс поиска решений итерационный. Каждая итерация l включает три этапа. На первом этапе каждый агент a_k популяции конструктивным алгоритмом находит решение $U^r{}_{0k}$, рассчитывает оценку решения $\xi_k(U^r{}_{0k})$ и значение степени пригодности полученного агентом решения φ_k (количество феромона, соответствующее оценке).

Первый этап осуществляется следующим образом.

Агенты обладают памятью. На каждом шаге t в памяти агента a_k имеется:

- количество феромона $\phi_j(t)$, отложенного на каждом ребре графа H;
- список вершин, уже включенных в формируемое множество – $U^r{}_{0k}(t\text{-}1)$ и список оставшихся (свободных) вершин $U^r{}_{ck}(t\text{-}1)$. $U^r{}_{ck}(t\text{-}1)=U^r\backslash U^r{}_{0k}(t\text{-}1)$, $U^r{}_{0k}(t\text{-}1)\cup U^r{}_{ck}(t\text{-}1)=U^r$.

Формирование решения (множества $U^r{}_{0k}$) осуществляется последовательно (пошагово). На каждом шаге t агент a_k применяет вероятностное правило выбора следующей вершины $u^r{}_j$ для включения ее формируемое множество $U^r{}_{0k}(t)$.

На первом шаге в формируемое множество $U^r{}_{0k}(t)$, где t=1, включается одна их вершин $u^r{}_j\in U^r$ графа H, смежная корневой вершине S_0.

На конечном шаге $t=n_0$ агентом a_k будет сформировано множество $U^r{}_{0k}(n_0)=U^r{}_{0k}$, $|U^r{}_{0k}(n_0)|=n_0$ такое, что любые две вершины $u^r{}_i\in U^r{}_{0k}$ и $u^r{}_j\in U^r{}_{0k}$ в графе $G_r=(U^r,V^r)$ не являются смежными. Шаг $t=n_0$ является конечным, если после его выполнения ни одна из вершин множества $U^r{}_{ck}(t)$ не может быть включена в $U^r{}_{0k}(t)$.

На шаге t формируется множество $U^r{}_{нk}(t)\subset U^r{}_{ck}(t\text{-}1)$ вершин, не смежных вершинам множества $U^r{}_{0k}(t\text{-}1)$ (рис. 4).

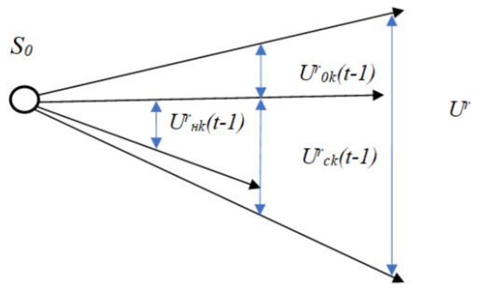

Рис. 4. Формирование множества $U^r{}_{нk}(t\text{-}1)$ вершин, не смежных с вершинами множества $U^r{}_{0k}(t\text{-}1)$

Вероятность включения в формируемое агентом a_k на базе звездного графа H множество $U^r{}_{0k}(t)$ вершины $u^r{}_j \in U^r{}_{нk}(t)$, не имеющей в графе $G_r=(U^r,V^r)$ связей с вершинами множества $U^r{}_{0k}(t-1)$, пропорциональна суммарному количеству феромона на ребре, связывающему вершину $u^r{}_j \in U^r{}_{нk}(t)$, $U^r{}_{нk}(t) \subset U^r$ с корневой вершиной S_0 в графе H.

Формируется множество ребер $E_{0k}(t-1) \subset E$ звездного графа H, связывающих вершины сформированного подмножества $U^r{}_{нk}(t-1)$ с корневой вершиной S_0.

Агент просматривает все ребра $(S_0,\ u^r{}_j) \in E_{0k}(t-1)$, $u^r{}_j \in U^r{}_{нk}(t-1)$. Для каждого ребра $e_j \in E_{0k}(t-1)$ рассчитываются два параметра:

$\phi_j(t)$ – суммарное количество феромона на ребре e_j (в массиве коллективной эволюционной памяти КЭП$_o$);

$s_j(t)$ – число связей на графе $G_r=(U^r,V^r)$ между $u^r{}_j$ и $U^r{}_{ck}(t-1)$.

По формуле (1) – при мультипликативной свертке, либо по формуле (2) – при аддитивной свертке определяется потенциальная степень предпочтения F_{ik} включения $u^r{}_j$

$$F_{jk}=(\phi_j(t))/(\,s_j(t)+1)^{\beta} \qquad (1)$$

$$F_{jk}=(\alpha\phi_j(t))+(\beta\,/(s_{ik}+1)), \qquad (2)$$

где $\alpha,\ \beta$ – управляющие параметры, которые подбираются экспериментально.

Вероятность $P_{ik}(t)$ включения вершины $u^r{}_j \in U^r{}_{нk}(t)$ в формируемое $U^r{}_{0k}(t)$, определяется следующим соотношением:

$$P_{ik}(t)=F_{ik}/\sum_i F_{ik}\ . \qquad (3)$$

Агент с вероятностью $P_{ik}(t)$ выбирает одну из вершин, которая включается в подмножество $U^r{}_{0k}(t)$ и исключается из подмножества $U^r{}_{ck}(t-1)$.

Количество φ_k откладываемого агентом a_k феромона на ребрах $E_{0k} \subset E$ звездного графа H, связывающих вершины $u^r{}_j \in U^r{}_{0k}$ сформированного внутри устойчивого подмножества $U^r{}_{0k}$ с вершиной S_0, пропорционально числу вершин n_0 сформированного подмножества $U^r{}_{0k}$: $\varphi_k=\mu n_0$, где μ – коэффициент пропорциональности. Чем больше n_0, тем больше феромона φ_k будет отложено агентом a_k на каждом ребре $(S_0,\ u^r{}_j) \in E_{0k}$ звездного графа H, построенного на вершинах множества $U^r{}_{0k}$. Следовательно, с большей вероятностью агенты будут выбирать такие вершины при синтезе собственного внутренне-устойчивого подмножества.

В работе используется циклический (ant-cycle) метод муравьиных систем. В этом случае рассчитанное агентом количество феромона φ_k предварительно откладывается в буферном массиве $Q_1=\{q_j|j=1,2,\ldots,m\}$ коллективной эволюционной памяти (КЭП$_б$) в ячейках q_j, соответствующих ребрам множества $E_{0k} \subset E$.

На втором этапе, после полного формирования всеми агентами решений на текущей итерации, феромон ω_j, накопленный в j-ой ячейке в буферном массиве КЭП$_б$, добавляется в каждую j-ю ячейку основного массива $Q_2=\{q_j|j=1,2,\ldots,m\}$ коллективной эволюционной памяти КЭП$_o$.

$$\phi_j(t)=\phi_j(t)+\omega_j. \qquad (4)$$

Для избежания преждевременной сходимости используется отрицательная обратная связь в виде испарения феромона. После отложения феромона в основной массив коллективной эволюционной памяти КЭП$_o$. на третьем этапе происходит общее испарение феромона на множестве ребер E звездного графа H в соответствии с формулой (5).

$$\phi_j(t)=\phi_j(t)\cdot(1-\rho)), \qquad (5)$$

где $\phi_j(t)$ – уровень феромона на ребре $e_j \in E_{0k}$, ρ – коэффициент обновления.

После выполнения всех действий на итерации находится агент с лучшим решением, которое запоминается. Далее осуществляется переход на следующую итерацию. Временная сложность этого алгоритма зависит от времени жизни колонии l (число итераций), количества вершин графа n и числа муравьев m, и определяется как $O(l*n^2*m)$.

После формирования в реберном графе $G_r=(U^r,V)$ графа $G(X, U)$ внутренне – устойчивого множества вершин $U^r{}_0 \subset U^r$ осуществляется переход от графа G_r к исходному графу G для построения паросочетания или выделения в графе клики.

Для преодоления локального барьера используются подходы, основанные на сочетании различных видов эволюции. В частности, эффективность показала композитная архитектура бионического поиска, заключающаяся в последовательной работе муравьиного и эволюционного алгоритма нахождения максимального паросочетания, рассмотренного в работе [11–14].

3. Экспериментальные исследования. Алгоритм нахождения паросочетания в графе реализован в виде программы **ПАР.**

Исследование программы **ПАР** проводились на примерах с известным оптимумом $F_{опт}$ по методологии BEKU [13–20].

Контрольные примеры для исследования содержали до 1000 вершин.

Для оценки качества используется безразмерная величина $\xi=F_{опт}/F$, где F – оценка решения полученного, полученного с помощью разработанного алгоритма. Зависимости показателя качества ξ от числа итераций и размера популяции представлены на рис. 4 и 5.

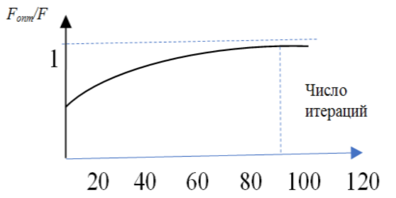

Рис. 5. Зависимость показателя ξ от числа итераций

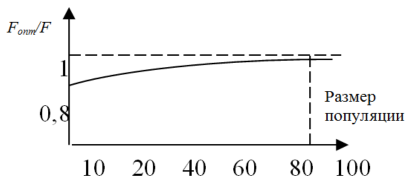

Рис. 6. Зависимости показателя ξ от размера популяции

Временная сложность алгоритма $O(n^2)$. Сравнение с известными алгоритмами показало, что при меньшем времени работы новый алгоритм дает более качественные решения.

Заключение. В работе рассматривается поисковый популяционный алгоритм. Разработана модифицированная метаэвристика по аналогии с моделями адаптивного поведения МА. Для поиска решения задачи в отличии от канонической парадигмы муравьиной колонии используется звездный граф поиска решений.

Основными достоинствами разработанного алгоритма являются:

♦ Низкая трудоемкость: временная сложность алгоритма $O(n^2)$.

♦ Реализация алгоритма отличается быстродействием и точностью вычислений.

Полученные в работе результаты можно использовать при решении задач, таких как: задачи о выделения внутренне-устойчивых множеств в графе, максимальной клике, проблема определения паросочетаний, раскраске графов и составление расписаний, задаче о назначениях.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Андерсон Д.* Дискретная математика и комбинаторика. – М.: Вильямс, 2003.
2. *Кормен К., Лейзерсон Ч., Ривест Р.* Алгоритмы построение и анализ. – М.: МЦМНО, 2000.
3. *Свами М., Тхуласираман К.* Графы, сети и алгоритмы. – М.: Мир, 1984.
4. *Лебедев Б.К., Лебедев О.Б.* Гибридный биоинспирированный алгоритм размещения базовых стандартных библиотечных элементов при проектировании топологии полузаказной 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. *Лебедев Б.К., Лебедев О.Б.* Покрытие на основе методов роевого интеллекта // Проблемы разработки перспективных микро- и наноэлектронных систем: Сб. трудов / под общ. ред. академика РАН А.Л. Стемпковского, 2016. – С. 187-193.
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. *Lebedev O.B.* Hybrid bioinspired algorithm of 1.5 dimensional bin-packing // Proceedings of the Third International Scientific Conference «Intelligent Information Technologies for Industry (IITI'18)». – 2018. – P. 254-263.
9. *Лебедев О.Б.* Модели адаптивного поведения муравьиной колонии в задачах проектирования. – Таганрог: Изд-во ЮФУ, 2013. – 199 с.
10. *Лебедев Б.К., Лебедев О.Б.* Ко-эволюционный алгоритм разбиения // Проблемы разработки перспективных микро- и наноэлектронных систем / под ред. А.Л. Стемпковского, 2020. – № 3. – С. 79-86.
11. *Лебедев Б.К., Лебедев О.Б.* Муравьиные алгоритмы разбиения, использующие представление задачи, отличные от канонического // Вестник Ростовского государственного университета путей сообщения. – 2016. – № 3 (63). – С. 42-47.
12. *Лебедев Б.К., Лебедев О.Б., Жиглатый А.А.* Размещение элементов СБИС на основе моделей роевого интеллекта // Проблемы разработки перспективных микро- и наноэлектронных систем: Сб. трудов / под общ. ред. академика РАН А.Л. Стемпковского. – М.: ИППМ 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. – P. 88-94.
14. *Sherwani N.A.* Algorithms for VLSI Physical Design Automation. – Third Edition, 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. MCNC Electronic and Information Technologies (Online). Available: www.mcnc.org.
18. hMetis [Online]. Available: http://www-users.cs.umn.edu/karypis/metis/hmet300. HB Floorplan Benchmarks [Online]. Available: http://cadlab.cs.ucla.edu/ cpmo/HBsuite.html.
19. IBM-PLACE 2.0 benchmark suits. – http://er.cs.ucla.edu/benchmarks/-ibm-place2/bookshelf/ibm-place2-all-bookshelf-nopad.tar.gz.
20. *Adya S.N.* ISPD02 IBM-MS Mixed-size Placement Benchmarks. – http://vlsicad.eecs.umich.edu/BK/ISPD02bench/.

REFERENCES

1. *Anderson D.* Diskretnaya matematika i kombinatorika [Discrete mathematics and combinatorics]. Moscow: Vil'yams, 2003.
2. *Kormen K., Leyzerson Ch., Rivest R.* Algoritmy postroenie i analiz [Algorithms construction and analysis]. Moscow: MTSMNO, 2000.
3. *Svami M., Tkhulasiraman K.* Grafy, seti i algoritmy [Graphs, networks and algorithms]. Moscow: Mir, 1984.
4. *Lebedev B.K., Lebedev O.B.* Gibridnyy bioinspirirovannyy algoritm razmeshcheniya bazovykh standartnykh bibliotechnykh 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.* Sovremennye algoritmy poiskovoy optimizatsii. Algoritmy, vdokhnovlennye prirodoy: ucheb. posobie [Modern search engine optimization algorithms. Algorithms inspired by nature]. Moscow: Izd-vo MGTU im. N.E. Baumana, 2016, 448 p.
6. *Lebedev B.K., Lebedev O.B.* Pokrytie na osnove metodov roevogo intellekta [Coverage based on swarm intelligence methods // *Problems of development of promising micro- and nanoelectronic systems], Problemy razrabotki perspektivnykh mikro- i nanoelektronnykh sistem: Sb. trudov* [Collection of works], under the total. ed. academician of the RAS A.L. Stempkovskogo, 2016, pp. 187-193.
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. *Lebedev O.B.* Hybrid bioinspired algorithm of 1.5 dimensional bin-packing, *Proceedings of the Third International Scientific Conference «Intelligent Information Technologies for Industry (IITI'18)»*, 2018, pp. 254-263.
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. *Lebedev B.K., Lebedev O.B.* Ko-evolyutsionnyy algoritm razbieniya [Co-evolutionary partitioning algorithm], *Problemy razrabotki perspektivnykh mikro- i nanoelektronnykh sistem* [Problems of development of promising micro- and nanoelectronic systems], ed. by A.L. Stempkovskogo, 2020, No. 3, pp. 79-86.
11. *Lebedev B.K., Lebedev O.B.* Murav'inye algoritmy razbieniya, ispol'zuyushchie predstavlenie zadachi, otlichnye ot kanonicheskogo [Ant partitioning algorithms using problem representation other than the canonical one], *Vestnik Rostovskogo gosudarstvennogo universiteta putey soobshcheniya* [Bulletin of the Rostov State University of Communications], 2016, No. 3 (63), pp. 42-47.
12. *Lebedev B.K., Lebedev O.B., Zhiglatyy 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* [Problems of development of promising micro- and nanoelectronic systems. Collection of works], under the total. ed. academician of the RAS A.L. Stempkovskogo. 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, pp. 88-94.
14. *Sherwani N.A.* Algorithms for VLSI Physical Design Automation. Third Edition, 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]. Modvow: Izd-vo MGTU imeni N.E. Baumana, 2006, 336 p.
17. MCNC Electronic and Information Technologies (Online). Available: www.mcnc.org.
18. hMetis [Online]. Available: http://www-users.cs.umn.edu/karypis/metis/hmet300. HB Floorplan Benchmarks [Online]. Available: http://cadlab.cs.ucla.edu/ cpmo/HBsuite.html.
19. IBM-PLACE 2.0 benchmark suits. Available at: http://er.cs.ucla.edu/benchmarks/-ibm-place2/bookshelf/ibm-place2-all-bookshelf-nopad.tar.gz.
20. 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.