

Раздел II. Биоинспирированный поиск

УДК 519.6

DOI 10.18522/2311-3103-2016-7-5362

Э.В. Кулиев, А.А. Лежебоков, Ю.А. Кравченко

РОЕВОЙ АЛГОРИТМ ПОИСКОВОЙ ОПТИМИЗАЦИИ НА ОСНОВЕ МОДЕЛИРОВАНИЯ ПОВЕДЕНИЯ ЛЕТУЧИХ МЫШЕЙ*

Работа посвящена решению ключевой проблемы искусственного интеллекта – разработке новых эффективных эвристических механизмов поиска. В качестве наиболее перспективных направлений развития теории поисковой оптимизации рассматриваются вопросы применения принципов и правил поведения объектов живой природы для решения NP-полных оптимизационных задач. Наибольший интерес представляют методы и алгоритмы на основе принципов мультиагентного управления и роевого интеллекта. В основу работы положен роевой алгоритм на основе моделирования поведения летучих мышей. Приведена модель поведения одной особи – эмулируется процесс движения в ограниченном пространстве поиска с целью нахождения квазиоптимальных решений. Представлены подходы к изменению управляющих параметров алгоритма – частот и амплитуды эхолокационных сигналов, вероятностных правил изменения направления движения. Разработана структурная схема алгоритма, который решает оптимизационную задачу размещения на плоскости некоторых одноагентных элементов, в качестве которых могут выступать компоненты сверхбольших интегральных схем. В работе представлен универсальный подход к кодированию решений в виде строки, позволяющий использовать разработанный алгоритм для разных исходных данных и ограничений задачи размещения. Исходная область на пространстве поиска формируется путем последовательного выполнения последовательного, случайного и итерационного алгоритмов. Выполнена программная реализация алгоритма и графического интерфейса пользователя на объектно-ориентированном языке программирования высокого уровня. Проведена серия экспериментальных исследований, целью которых являлось подтверждение оценок временной сложности и эффективности разработанного алгоритма. Сложность роевого алгоритма на основе поведения летучих мышей линейно зависит от количества особей и имеет квадратичный характер зависимости от размерности исходной задачи, то есть от количества размещаемых элементов, определяющих длину строки решения.

Размещение; адаптивные процедуры; моделирование эволюции; роевой алгоритм; генетический алгоритм; летучие мыши; программная реализация.

E.V. Kuliev, A.A. Lezhebokov, Yu.A. Kravchenko

SWARMS ALGORITHMS OF SEARCH OPTIMIZATION BASED ON MODELING THE BEHAVIOR OF BATS

The paper discusses to solve of the key artificial intelligence problem – development of new effective heuristic search mechanisms. As the most perspective directions in search optimization theory is considered principles and rules of species behavior in nature for solving NP-hard optimization problems. The most interesting are methods and algorithms based on multiagent management and swarm intelligent. The paper deals with swarm algorithm based on simulation of bats' behavior. The model of one specie's behavior is a simulation of it's movement within limited space to find a quasi optimal solution. There are represented several approaches to manage the algorithm configurations – frequency and range of echo signals, probabilistic rules for changing the direction of movement. The authors developed a flowchart of algorithm for the VLSI single-

* Работа выполнена при частичной поддержке РФФИ (грант № 16-01-00586) и Минобрнауки России в рамках проектной части госзадания №8.823.2014/К.

sized elements placement problem. An unified approach for solution encoding as a string allow to use the developed algorithm for various initial data and restrictions of the placement problem. Initial area of the search space is formed by sequential, random and iterative algorithms. There is a software on the basis of object oriented language. A set of experimental computations were carried out to confirm the algorithm time complexity and it efficiency. The complexity of the swarm algorithm based on bats' behavior depends on a number of species linearly and has a quadratic character of dependency on dimensionality of initially problem, i.e. on a number of placed elements which denotes a solutions' string length.

Placement; adaptive procedures; evolution simulation; swarm algorithm; genetic algorithm; bats; software.

Введение. Природа всегда была источником вдохновения для исследователей. На основе ее принципов и механизмов был разработан целый ряд алгоритмов, в том числе для решения различных задач оптимизации. Данные алгоритмы относятся к метаэвристическим и включают в себя эволюционные и алгоритмы роевого интеллекта [1].

При решении многих практических задач часто возникает необходимость выбора наилучшего решения по некоторому критерию из множества возможных. Математически такой выбор формализуется в виде задачи оптимизации.

Широко известные методы математического программирования для решения задач оптимизации представляют собой детерминированную итерационную процедуру пошагового улучшения одного текущего решения. Эффективность таких алгоритмов основывается на полном использовании удобных с точки зрения оптимизации свойств целевой функции [1–3]. Для многих практических задач такие свойства либо не выполняются, либо неизвестны заранее, поэтому применение данных методов нецелесообразно. Для решения таких задач в настоящее время используются недетерминированные (стохастические), работающие одновременно с большим количеством текущих решений (многоагентные) алгоритмы, являющиеся более эффективными и универсальными. Например, генетический алгоритм, метод роя частиц, муравьиный алгоритм и т.д.

Многоагентные алгоритмы, основанные на использовании популяции, работают с набором потенциальных решений [2]. Каждое решение постепенно улучшается и оценивается, таким образом, каждое потенциальное решение влияет на то, как будут улучшены другие решения. Большинство популяционных методов заимствовало эту концепцию из биологии: процесс поиска наилучшего решения «копирует» некоторый природный процесс либо поведение определенных видов животных, причем учитываются их видовые особенности. Класс сложных систем, именуемых как алгоритмы стайного типа, также часто употребляется термин «бионические алгоритмы», – богатый источник нестандартных численных методов, с помощью которых можно решать сложные задачи, когда известно недостаточно информации об оптимизируемой функции [2–4,5, 7–11].

Один из особенно популярных и часто используемых бионических методов известен как стайный алгоритм оптимизации (Particle Swarm Optimization, PSO), также называемый методом роя частиц (МРЧ) [6]. Данный метод был предложен для решения задач безусловной оптимизации с вещественными переменными Кеннеди и Эберхартом в 1995 году. Идея стайного алгоритма была почерпнута из социального поведения стада копытных, стаи птиц, косяка рыб и т.д., то есть процесс поиска оптимального решения данным методом имитирует поиск пищи группой животных, насекомых или птиц одного вида без уточнения самого вида.

Следует также отметить, что большинство алгоритмов стайного типа изначально разработано для решения задач безусловной оптимизации с вещественными переменными, и все они имеют некоторые параметры, которые необходимо подбирать при решении той или иной задач (наиболее значимый в этом понимании «параметр» – размер популяции потенциальных решений) [3].

Целью работы является исследование эффективности одного из алгоритмов роевого интеллекта – алгоритма летучих мышей (Bat Algorithm).

Модель поведения летучих мышей. Алгоритм летучих мышей (Bat Algorithm, BA) – метаэвристический алгоритм, разработанный Янгом в 2010 году, и имитирующий свойство эхолокации летучих мышей [2–3].

Большинство видов летучих мышей обладает удивительно совершенными средствами эхолокации, которая используется ими для обнаружения добычи и препятствий, а также для обеспечения возможности разместиться в темноте на насесте. Параметры лоцирующего звукового импульса летучих мышей различных видов меняются в широких пределах, отражая их различные охотничьи стратегии. Большинство мышей используют короткие частотно-модулированные в пределах примерно одной октавы сигналы. В то же время некоторые виды не используют частотную модуляцию своих звуковых импульсов.

Алгоритм летучих мышей подчиняется следующим правилам:

1) все летучие мыши используют эхолокацию, чтобы анализировать расстояние, а также иметь различие между едой (добычей) и природными препятствиями;

2) летучие мыши перемещаются случайным образом со скоростью V_i в позиции x_i с фиксированной частотой f_{min} , изменяемой длиной волны λ и громкостью A_0 , чтобы найти добычу. Они могут автоматически регулировать длину волны (или частоту, т.к. частота = 1/длина волны) испускаемого импульса и скорость импульса v [0,1], зависящих от близости цели;

3) громкость изменяется от большего (положительного) A_0 к меньшему постоянному значению A_{min} [3].

Модель алгоритма летучих мышей. На рис. 1 представлена модель решения оптимизационной задачи на основе роевого алгоритма поведения летучих мышей. Решение оптимизационной задачи, в нашем случае задачи размещения, осуществляется на множестве решений X . Часть решений может оказаться недопустимыми согласно исходным условиям и ограничения решаемой задачи, на рисунке такие области обозначены как «препятствия». Каждый поисковый агент (летучая мышь) обладает следующими характеристиками: текущее положение (x_i), скорость движения (v_i), громкость сигнала (A_i) и его частоту (w_i). Громкость звукового сигнала изменяется от большего начального значения A_0 до меньшего заданного A_{min} . Громкость и частота звуковых сигналов каждой отдельной мыши изменяются по мере приближения к некоторой цели (искомому оптимуму или препятствию).

По мере приближения агента к локальному (или глобальному) оптимуму громкость издаваемого сигнала уменьшается, а его интенсивность увеличивается. Приближение определяется по текущему изменению значения целевой функции. После каждой итерации для каждой мыши определяется ее новое расположение, это осуществляется в области локального поиска в окрестности текущего месторасположения. Для повышения эффективности алгоритма в случае многомерного поиска (сложной операции кодирования/декодирования) имеется возможность выбирать случайные координаты, подлежащие изменению. Все модификации данного алгоритма, связаны с вариантами изменения управляющих параметров (скорости, громкости, интенсивности) [12–15].

Количество мышей задается оператором, а их начальное расположение и значения управляющих параметров задаются с помощью процедуры генерации начального расположения. Данная процедура реализуется путем комбинирования текущего роевого алгоритма и методов случайного или итерационного поиска [13, 16, 18] – как стартовых точек отправления летучих мышей.

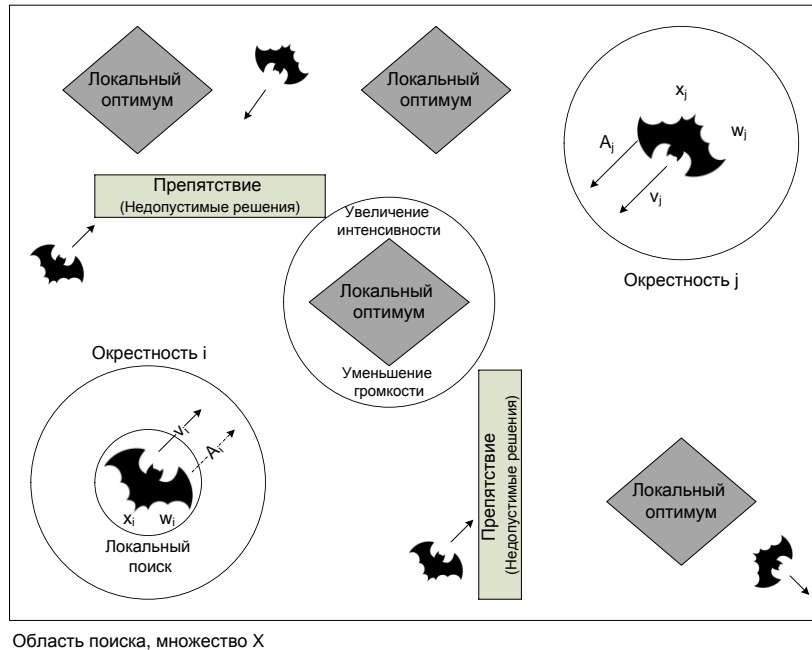


Рис. 1. Модель роевого алгоритма на основе поведения летучих мышей

Алгоритм летучих мышей. Структура разработанного роевого алгоритма представлена на рис. 2. Как уже было отмечено, в основу алгоритма положен роевой алгоритм на основе моделирования поведения летучих мышей. Данный алгоритм скомбинирован с классическими методами случайного и итерационного поиска – для генерации начального положения всех поисковых агентов (блок №2), генерации окрестностей внутри области поиска.

В блоке №3 выполняется генерация начальных случайных значений о скорости движения, громкости и интенсивности сигнала для каждой мыши (агента). Алгоритм работает итерационно, счетчик пройденных итераций проверяется в блоке №12. На каждой итерации каждая особь эволюционирует. В блоке №4 выполняется проверка, все ли мыши были рассмотрены для применения правил эволюции (изменения управляющих параметров).

В блоках №5, 6, 7 выполняется эволюция поведения отдельной мыши согласно ее текущему расположению и значению целевой функции.

Интенсивность: $w_i = w_{\min} + (w_{\max} - w_{\min}) * \beta$,

Скорость: $v_i^{(t+1)} = v_i^{(t)} + (x_i^{(t)} - x_i) * w_i$.

Расположение: $x_i^{(t+1)} = x_i^t + v_i^{(t+1)}$.

Для определения новых положений летучих мышей, каждая из них осуществляет локальный поиск в окрестности своего текущего положения путем случайного блуждания:

$$x_i^{\text{new}} = x_i^{\text{old}} + e * A(t).$$

Эволюция параметров, характеризующих испускаемые звуковые сигналы, осуществляется по формуле:

$$A_i^{(t+1)} = \alpha * A_i^{(t)}.$$

Значение весовых коэффициентов α и β вводит оператор.

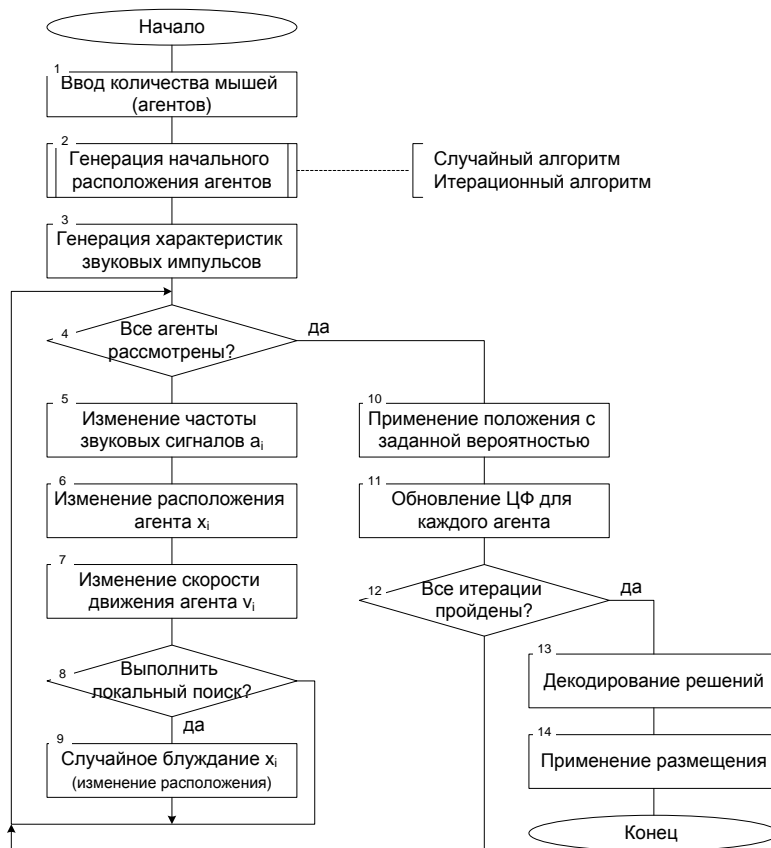


Рис. 2. Алгоритм размещения на основе поведения летучих мышей

В блоке №8 выполняется генерация случайной вероятности и в случае успешной проверки переход к блоку №9, в котором осуществляется генерация нескольких локальных решений (их количество задается оператором) в пределах текущей окрестности.

Как только проанализированы все мыши, для каждой из них применяется новое вычисленное расположение (на основе результатов локального поиска) с заданной вероятностью (блок №10). После чего для тех мышей, которые изменили свое расположение, происходит вычисление целевой функции (блок №11).

Как только алгоритм пройдет заданное число итераций (блок №12) происходит декодирование решений (из числовой последовательности в координаты дискретного рабочего поля) в блоке №13. В блоке №14 найденное лучшее решение будет применено для решения исходной оптимизационной задачи размещения (рис. 3).

Разработанный алгоритм обладает рядом преимуществ по сравнению с другими представителями роевых алгоритмов [17–19]. Во-первых, это возможность учитывать области решений недопустимые по тем или иным признакам. Во-вторых, алгоритм позволяет регулировать скорость и интенсивность поиска каждым агентом, что позволяет за небольшое количество итераций (приемлемое время) находить локально-оптимальные решения.

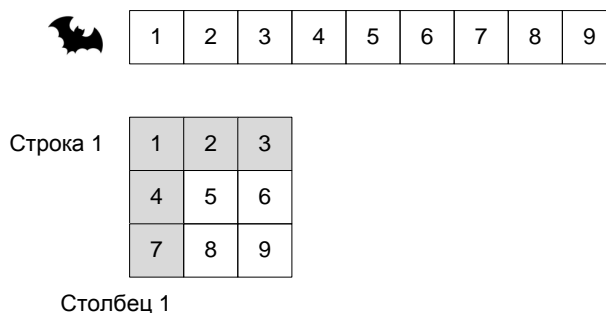


Рис. 3. Декодирование решения для задачи размещения

Экспериментальные исследования. При проведении анализа и исследования характеристик разработанного алгоритма были использованы схемы с различным количеством элементов. Целью проведения экспериментов – установление временной сложности алгоритма (ВСА) [20].

Для определения ВСА алгоритма был проведён ряд экспериментов, в которых изменялось количество элементов: от 100000 до 1000000, с шагом 100000.

После проведения ряда экспериментов были получены результаты времени работы алгоритма в среднем случае, отражённые соответственно в табл. 1 и на рис. 4.

Таблица 1

Сравнение временной сложности алгоритмов

Алгоритм	Количество элементов								
	100000	200000	300000	400000	500000	600000	700000	800000	900000
Генетический (с)	11,05	21,57	33,78	54,01	72,15	94,01	125,6	169,5	185,7
Муравьиный (с)	21,4	52,12	71,82	110,1	180,5	205,9	298,8	359,6	404,9
Летучие мыши (с)	35,58	84,45	157,56	248,78	351,7	409,9	558,9	694,8	857,7

График зависимости времени работы от количества элементов подтверждает сделанные предположения о временной сложности разработанного алгоритма основанного на поведении летучих мышей, которая имеет квадратичный характер и может быть выражена формулой $O(\alpha \cdot n^2)$.

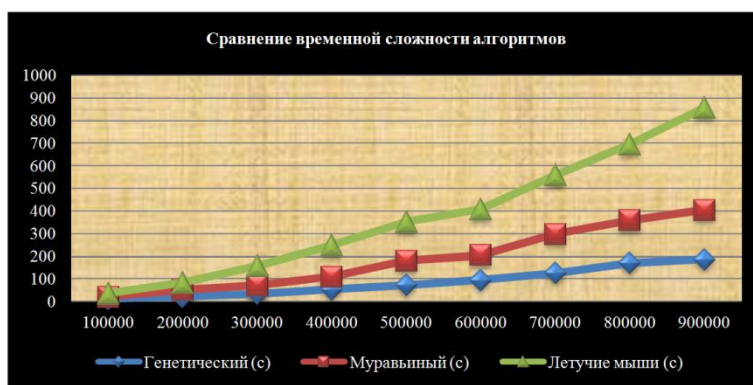


Рис. 4. График зависимости времени работы от количества элементов

Для определения эффективности разработанного алгоритма были проведены исследования качества решения на нескольких тестовых примерах. Под эффективностью алгоритма будем понимать качество решения, получаемого с его помощью (табл. 2). На рис. 5 приведена гистограмма сравнения качества решения, основываясь на данных из табл. 2.

Таблица 2

Сравнение эффективности алгоритмов

Алгоритмы (мс)	Число элементов схемы				
	100000	250000	500000	750000	1000000
Генетический	87,7	128,4	142	171,4	215,7
Муравьиный	75,1	110,1	133	154,9	203,8
Летучие мыши	66,5	98,3	101,7	139,1	177,4

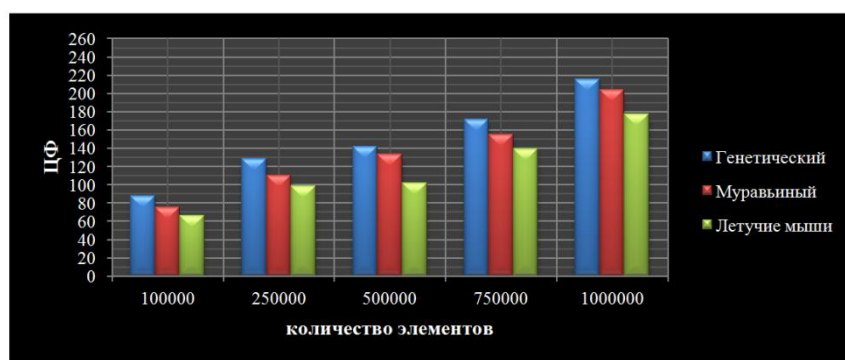


Рис. 5. Гистограмма сравнения качества решения

Из результатов эксперимента видно, что наиболее эффективным является разработанный алгоритм, основанный на поведении летучих мышей. Он в среднем на 12 % эффективнее генетического алгоритма и на 8 % эффективнее муравьиного алгоритма.

Заключение. Предложен новый роевой алгоритм для решения оптимизационных задач этапа конструкторского проектирования, в основу которого положена модель поведения летучих мышей при движении в ограниченном пространстве. Приведена модель поведения особи – движение в области поиска с целью нахождения локально-оптимальных участков. Представлены новые подходы к изменению управляющих параметров алгоритма – частоты, амплитуды, направления движения, на основе весовых коэффициентов. Разработана структурная схема алгоритма основанного на поведении летучих мышей. Исходная область поиска формируется за счет последовательного, случайного и итерационного алгоритмов. Разработана программная среда на языке C++. Проведен вычислительный эксперимент. Проведенные серии тестов и экспериментов позволили уточнить теоретические оценки временной сложности алгоритмов и их поведение для схем различной структуры. В лучшем случае временная сложность алгоритмов $\approx O(n \log n)$, в худшем случае – $O(n^2)$.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Норенков И.П., Арутюнян Н.М. Эволюционные методы в задачах выбора проектных решений // Научно-техническое издание МГТУ им. Н.Э. Баумана «Наука и образование». – 2007. – № 9.

2. Карпенко А.П. *Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой: учебное пособие.* – М.: Изд-во МГТУ им. Н.Э. Баумана, 2014. – 448 с.
3. Ахмедова Ш.А. Об эффективности «стайного» алгоритма оптимизации // Труды XLIII Краевой научной студенческой конференции по математике и компьютерным наукам. – Красноярск: СФУ, 2010. – С. 9-12.
4. Курейчик В.В., Запорожец Д.Ю. Современные проблемы при размещении элементов СБИС // Известия ЮФУ. Технические науки. – 2011. – № 7 (120). – С. 68-73.
5. Кулиев Э.В., Лежебоков А.А. О гибридном алгоритме размещения компонентов СБИС // Известия ЮФУ. Технические науки. – 2012. – № 11 (136). – С. 188-192.
6. Курейчик В.В., Бова В.В., Курейчик В.В. Комбинированный поиск при проектировании // Образовательные ресурсы и технологии. – 2014. – № 2 (5). – С. 90-94.
7. Курейчик В.В., Курейчик Вл.Вл. Бионический поиск при проектировании и управлении // Известия ЮФУ. Технические науки. – 2012. – № 11 (136). – С. 178-183.
8. Zaporozhets D.U., Zaruba D.V., Kureichik V.V. Representation of solutions in genetic VLSI placement algorithms // Proceedings of IEEE East-West Design and Test Symposium, EWDTs 2014.
9. Zaporozhets D.Yu., Zaruba D.V., Kureichik V.V. Hybrid bionic algorithms for solving problems of parametric optimization // World Applied Sciences Journal. – 2013. – No. 23 (8). – P. 1032-1036.
10. Kuliev E.V., Dukhardt A.N., Kureychik V.V., Legebokov A.A. Neighborhood research approach in swarm intelligence for solving the optimization problems // Proceedings of IEEE East-West Design and Test Symposium, EWDTs 2014.
11. Курейчик В.В., Запорожец Д.Ю. Роевой алгоритм в задачах оптимизации // Известия ЮФУ. Технические науки. – 2010. – № 7 (108). – С. 28-32.
12. Bova V.V., Lezhebokov A.A., Gladkov L.A. Problem-oriented algorithms of solutions search based on the methods of swarm intelligence // World Applied Sciences Journal. – 2013. – No. 27 (9). – P. 1201-1205.
13. Zaruba D., Zaporozhets D., Kureichik V. VLSI placement problem based on ant colony optimization algorithm // Advances in Intelligent Systems and Computing. – 2016. – No. 464. – P. 127-133.
14. Kureichik V., Kureichik V., Bova V. Placement of VLSI fragments based on a multilayered approach // Advances in Intelligent Systems and Computing. – 2016. – No. 464. – P. 181-190.
15. Kureichik V.V., Zaruba D.V. The bioinspired algorithm of electronic computing equipment schemes elements placement // Advances in Intelligent Systems and Computing. – 2015. – No. 347. – P. 51-58.
16. Zaporozhets D., Zaruba D.V., Kureichik V.V. Hierarchical approach for VLSI components placement // Advances in Intelligent Systems and Computing. – 2015. – No. 347. – P. 79-87.
17. Zaporozhets D.U., Zaruba D.V., Kureichik V.V. Representation of solutions in genetic VLSI placement algorithms // Proceedings of IEEE East-West Design and Test Symposium, EWDTs 2014.
18. Кулиев Э.В., Лежебоков А.А., Дуккардт А.Н. Подход к исследованию окрестностей в роевых алгоритмах для решения оптимизационных задач // Известия ЮФУ. Технические науки. – 2014. – № 7 (156). – С. 15-26.
19. Кулиев Э.В., Запорожец Д.Ю., Ксолов А.М. Биоинспирированный поиск при решении задачи размещения компонентов СБИС // Известия Кабардино-Балкарского научного центра РАН. – 2014. – № 6 (62). – С. 58-65.
20. Кулиев Э.В., Лежебоков А.А. Исследование характеристик гибридного алгоритма размещения // Известия ЮФУ. Технические науки. – 2013. – № 3 (140). – С. 255-261.

REFERENCES

1. Norenkov I.P., Arutyunyan N.M. Evolyutsionnye metody v zadachakh vybora proektnykh resheniy [Evolutionary techniques in the problems of design choices], *Nauchno-tekhicheskoe izdanie MGTU im. N.E. Baumana «Nauka i obrazovanie»* [Scientific and technical periodical of the Bauman MSTU N. Uh. Bauman "Science and education"], 2007, No. 9.
2. Карпенко А.П. *Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой: учебное пособие* [Modern algorithms of search engine optimization. Algorithms inspired by nature: a training manual]. Moscow: Izd-vo MGTU im. N.E. Baumana, 2014, 448 p.

3. Akhmedova Sh.A. Ob effektivnosti «staynogo» algoritma optimizatsii [The effectiveness of the "schooling" of the optimization algorithm], *Trudy XLIII Kraevoy nauchnoy studentcheskoy konferentsii po matematike i komp'yuternym naukam* [Proceedings of XLIII the Regional student scientific conference on mathematics and computer science]. Krasnoyarsk: SFU, 2010, pp. 9-12.
4. Kureychik V.V., Zaporozhets D.Yu. Sovremennye problemy pri razmeshchenii elementov SBIS [Modern placement's problems of VLSI], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2011, No. 7 (120), pp. 68-73.
5. Kuliev E.V., Lezhebokov A.A. O gibridnom algoritme razmeshcheniya komponentov SBIS [On the hybrid algorithm of component placement VLSI], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2012, No. 11 (136), pp. 188-192.
6. Kureychik V.V., Bova V.V., Kureychik V.V. Kombinirovannyi poisk pri proektirovanii [Combined search in the design], *Obrazovatel'nye resursy i tekhnologii* [Educational Resources and Technology], 2014, No. 2 (5), pp. 90-94.
7. Kureychik V.V., Kureychik V.V. Bionicheskiy poisk pri proektirovanii i upravlenii [Search inspired by natural systems, for the design and management], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2012, No. 11 (136), pp. 178-183.
8. Zaporozhets D.U., Zaruba D.V., Kureichik V.V. Representation of solutions in genetic VLSI placement algorithms, *Proceedings of IEEE East-West Design and Test Symposium, EWDTS 2014*.
9. Zaporozhets D.Yu., Zaruba D.V., Kureichik V.V. Hybrid bionic algorithms for solving problems of parametric optimization, *World Applied Sciences Journal*, 2013, No. 23 (8), pp. 1032-1036.
10. Kuliev E.V., Dukkardt A.N., Kureychik V.V., Legebokov A.A. Neighborhood research approach in swarm intelligence for solving the optimization problems, *Proceedings of IEEE East-West Design and Test Symposium, EWDTS 2014*.
11. Kureychik V.V., Zaporozhets D.Yu. Roeffvoy algoritm v zadachakh optimizatsii [Swarm algorithm in optimisation problems], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2010, No. 7 (108), pp. 28-32.
12. Bova V.V., Lezhebokov A.A., Gladkov L.A. Problem-oriented algorithms of solutions search based on the methods of swarm intelligence, *World Applied Sciences Journal*, 2013, No. 27 (9), pp. 1201-1205.
13. Zaruba D., Zaporozhets D., Kureichik V. VLSI placement problem based on ant colony optimization algorithm, *Advances in Intelligent Systems and Computing*, 2016, No. 464, pp. 127-133.
14. Kureichik V., Kureichik V., Bova V. Placement of VLSI fragments based on a multilayered approach, *Advances in Intelligent Systems and Computing*, 2016, No. 464, pp. 181-190.
15. Kureichik V.V., Zaruba D.V. The bioinspired algorithm of electronic computing equipment schemes elements placement, *Advances in Intelligent Systems and Computing*, 2015, No. 347, pp. 51-58.
16. Zaporozhets D., Zaruba D.V., Kureichik V.V. Hierarchical approach for VLSI components placement, *Advances in Intelligent Systems and Computing*, 2015, No. 347, pp. 79-87.
17. Zaporozhets D.U., Zaruba D.V., Kureichik V.V. Representation of solutions in genetic VLSI placement algorithms, *Proceedings of IEEE East-West Design and Test Symposium, EWDTS 2014*.
18. Kuliev E.V., Lezhebokov A.A., Dukkardt A.N. Podkhod k issledovaniyu okrestnostey v roevykh algoritmakh dlya resheniya optimizatsionnykh zadach [Approach to research environs in swarms algorithm for solution of optimizing problems], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2014, No. 7 (156), pp. 15-26.
19. Kuliev E.V., Zaporozhets D.Yu., Ksalov A.M. Bioinspirovannyi poisk pri reshenii zadachi razmeshcheniya komponentov SBIS [Bioinspired search when solving the problem of component placement of VLSI], *Izvestiya Kabardino-Balkarskogo nauchnogo tsentra RAN* [Izvestiya of Kabardino-Balkar scientific centre of the RAS], 2014, No. 6 (62), pp. 58-65.
20. Kuliev E.V., Lezhebokov A.A. Issledovanie kharakteristik gibridnogo algoritma razmeshcheniya [Research parameters of hybrid algorithm for placement], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2013, No. 3 (140), pp. 255-261.

Статью рекомендовала к опубликованию д.т.н., профессор Л.С. Лисицына.

Кулиев Эльмар Валерьевич – Южный федеральный университет; e-mail: ekuliev@sfedu.ru; 347928, г. Таганрог, пер. Некрасовский, 44; тел.: 88634371651; кафедра систем автоматизированного проектирования; ассистент.

Лежебоков Андрей Анатольевич – e-mail: legebokov@gmail.com; кафедра систем автоматизированного проектирования; доцент.

Кравченко Юрий Алексеевич – e-mail: krav-jura@yandex.ru; кафедра систем автоматизированного проектирования; доцент.

Kuliev Elmar Valerievich – Southern Federal University; e-mail: ekuliev@sfedu.ru; 44, Nekrasovskiy, Taganrog, 347928, Russia; phone: +78634371651; the department of computer aided design; assistant.

Lezhebokov Andrey Anatolievich – e-mail: legebokov@gmail.com; the department of computer aided design; associate professor.

Kravchenko Yury Alekseevich – e-mail: krav-jura@yandex.ru; the department of computer aided design; associate professor.

УДК 519.712.2

DOI 10.18522/2311-3103-2016-7-6273

Л.А. Гладков, Н.В. Гладкова, М.Ю. Лаврик

РЕШЕНИЕ ЗАДАЧ ПРОИЗВОДСТВЕННОГО ПЛАНИРОВАНИЯ НА ОСНОВЕ ГИБРИДНЫХ ЭВОЛЮЦИОННЫХ МЕТОДОВ*

Рассматриваются методы решения задач планирования производственного процесса. Приведена классификация задач планирования. Перечислены основные типы задач планирования, предусматривающие составление планов на различные сроки. Подробно рассмотрена задача оперативного планирования производства. Определены задачи, которые необходимо решить в процессе планирования. Перечислены условия построения и решения задачи оперативного планирования. Приведен пример задачи построения графика выполнения производственного процесса. Показаны примеры отображения процесса решения в графическом виде. Сформулирована задача построения графика производственного процесса в терминах теории расписаний. Приведены основные критерии и ограничения, используемые при и решении задач производственного планирования. Сформулированы требования к набору входной информации, требуемой для успешного решения поставленной задачи. Выполнена математическая постановка задачи оперативного планирования производственного процесса. Сформулированы и описаны основные критерии оценки качества получаемых решений. Предложен генетический алгоритм поиска и оптимизации решений при построении графика производственного процесса. Приведено описание последовательности выполнения операций генетического алгоритма. В статье предложен новый гибридный метод решения задачи планирования, основанный на использовании генетических алгоритмов и модели нечеткого логического контроллера. Приводится описание используемых логическим контроллером правил, структуры предложенного алгоритма. Приведены примеры нечетких правил, используемых в гибридном алгоритме. Представлена структурная схема, разработанного гибридного алгоритма оперативного планирования производства. На основе предложенного алгоритма разработана программа решения задачи построения временного графика производственного процесса. Представлено краткое описание проведенных вычислительных экспериментов, подтверждающих эффективность предложенного метода.

Временной график производственного процесса; оперативное планирование; эволюционные вычисления; нечеткие генетические алгоритмы; нечеткий логический контроллер.

* Работа выполнена при поддержке РФФИ, грант № 16-01-00715.