

17. *Lebedev B.K., Lebedev O.B. Lebedeva E.M.* Modernizirovannyi murav'inyi algoritm sinteza identifikirovannogo dereva gil'otinnogo razreza pri planirovani SBIS [A modernized ant algorithm for the synthesis of the identified guillotine tree tree in the course of VLSI planning], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2017, No. 7 (192), pp. 15-28.
18. *Gladkov L.A., Kureychik V.V., Kureychik V.M., Sorokoletov P.V.* Bioinspirirovannye metody v optimizatsii: monografiya [Bioinspired methods in optimization: monograph]. Moscow: Fizmatlit, 2009, 384 p.
19. *Lebedev B.K., Lebedev O.B. Lebedeva E.O.* Roeffy algoritm planirovaniya raboty mnogoprotseessornykh vychislitel'nykh sistem [Swarm algorithm for planning the work of multiprocessor computer systems], *Inzhenernyi vestnik Dona* [Engineer's Herald of the Don], 2017, No. 3.
20. *Cong J., Romesis M., and Xie M.* Optimality, Scalability and Stability Study of Partitioning and Placement Algorithms, *Proc. of the International Symposium on Physical Design*. Monterey, CA, April 2003, pp. 88-94.

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

**Лебедев Борис Константинович** – Южный федеральный университет; e-mail: lebedev.b.k@gmail.com; 347928, г. Таганрог, пер. Некрасовский, 44; тел.: 89282897933; кафедра систем автоматизированного проектирования; профессор.

**Лебедев Олег Борисович** – e-mail: lebedev.ob@mail.ru; тел.: 89085135512; кафедра систем автоматизированного проектирования; доцент.

**Лебедева Екатерина Олеговна** – e-mail: lebedev.ob@mail.ru; тел.: 89289591426; кафедра систем автоматизированного проектирования; магистрант.

**Lebedev Boris Konstantinovich** – Southern Federal University; e-mail: lebedev.b.k@gmail.com; 44, Nekrasovsky, Taganrog, 347928, Russia; phone: +79282897933; the department of computer aided design; professor.

**Lebedev Oleg Borisovich** – e-mail: lebedev.ob@mail.ru; phone: +79085135512; the department of computer aided design; associate professor.

**Lebedeva Ekaterina Olegovna** – e-mail: lebedev.ob@mail.ru; phone: +79289591426; the department of computer aided design; graduate student.

УДК 681.3.06

DOI 10.23683/2311-3103-2018-4-136-145

**Д.Ю. Запорожец, Д.В. Заруба**

### **ПАРАЛЛЕЛЬНЫЙ ПОПУЛЯЦИОННЫЙ АЛГОРИТМ\***

*Предложена архитектура параллельного поиска. Выделены три уровня распараллеливания: микроуровень – распараллеливание задач на уровне выполнения операторов генерации новых альтернативных решений; макроуровень – распараллеливание поиска на уровне генераций; метауровень – распараллеливание процесса поиска на уровне изолированного множества альтернативных решений. В качестве примера, в данной работе предложена реализация параллельного генетического алгоритма с распараллеливанием на микроуровне. Предложена обобщенная архитектура и описаны основные шаги нового параллельного генетического алгоритма с распараллеливанием на микроуровне. Основной идеей данного подхода является инициализация  $N$  потоков, в каждом из которых выполняется оператор кроссинговера и, с заданной долей вероятности, операторы мутации и инверсии. Для всех новых решений производится расчет целевой функции в том потоке, в котором данное решение было получено. Алгоритм ожидает окончания всех потоков. Далее производится*

\* Работа выполнена при финансовой поддержке гранта Президента Российской Федерации для государственной поддержки молодых российских ученых – кандидатов наук № МК-92.2017.8.

объединение всех новых сгенерированных решений в одну популяцию и выполняется оператор селекции. Процесс продолжается итерационно по достижению заданного числа итераций. Для подтверждения эффективности предложенного подхода была разработана программная реализация параллельного генетического алгоритма. В качестве прикладной задачи была выбрана одна из классических оптимизационных задач – нахождение Гамильтонова цикла в графе. Целью проведения экспериментальных исследований является скорость работы реализованного генетического алгоритма, а также сравнение его с классическим «последовательным» генетическим алгоритмом. Исследования проводились на двух различных платформах с процессорами: Intel I7 – 4 ядра, 2.7 ГГц и AMD 6000 – 2 ядра 2.5 ГГц. Проведены экспериментальные исследования, которые показали значительный прирост в скорости работы алгоритма при количестве вершин в графе более 250. Скорость работы алгоритма выросла на 100 – 200 % по сравнению с классической реализацией в зависимости от используемого процессора. Таким образом экспериментально было доказано, что при использовании одного и того же аппаратного обеспечения, предлагаемый параллельный подход позволяет сократить время работы алгоритма оптимизации в несколько раз по сравнению с его классической последовательной реализацией.

*Генетические алгоритмы; параллельный поиск; популяционные алгоритмы; задача коммивояжера; гамильтонов цикл.*

**D.Ju. Zaporozhets, D.V. Zaruba**

### PARALLEL POPULATION ALGORITHM

*The architecture of parallel search is proposed herein. The following three levels of parallelization are distinguished: microlevel, that is the parallelization of tasks at the level of execution of operators generating new alternative solutions; macrolevel, that is the parallelization of the search at the generation level; meta-level, that is the parallelization of the search process at the level of an isolated set of alternative solutions. As an example, herein we propose the implementation of a parallel genetic algorithm with parallelization at the microlevel. A generalized architecture is proposed and the main steps of a new parallel genetic algorithm with parallelization at the micro level are described. The basic idea of this approach is the initialization of N threads, in each of which the crossing-over operator is executed and, with a certain probability, mutation and inversion operators. For all new solutions, the fitness is calculated in the flow in which the decision was received. The algorithm waits for all threads to end. Then, all new generated solutions are combined into one population and the selection operator is executed. The process continues iteratively after reaching the specified number of iterations. To confirm the effectiveness of the proposed approach, a software implementation of the parallel genetic algorithm was developed. As an applied problem one of the classical optimization problems was chosen - the Hamiltonian cycle in the graph. The purpose of conducting experimental studies is the speed of the implemented genetic algorithm, as well as its comparison with the classical "sequential" genetic algorithm. The research was conducted on two different platforms with processors: Intel I7 – 4 cores, 2.7 GHz and AMD 6000 – 2 2.5 GHz cores. Experimental studies have been performed that showed a significant increase in the speed of the algorithm with a number of vertices in the graph of more than 250. The speed of the algorithm increased by 100–200 % compared with the classical implementation, depending on the processor used. Thus, it has been experimentally proved that with the use of the same hardware, the proposed parallel approach makes it possible to shorten the running time of the optimization algorithm by several times compared to its classical sequential implementation.*

*Genetic algorithms; parallel search; population algorithms; traveling salesman problem; hamiltonian cycle.*

**Введение.** Для повышения скорости решения NP-трудных задач оптимизации в середине 80-х гг. стали эффективно исследоваться стохастические поисковые методы оптимизации, основанные на природных явлениях. Данный класс алгоритмов, названных популяционными, предполагают обработку множества альтернативных решений задачи оптимизации и представляют собой альтернативу классическим поисковым алгоритмам, которые работают только с одним решением поставленной оптимизационной задачи [1].

Все популяционные алгоритмы относятся к классу эвристических алгоритмов, то есть алгоритмов, для которых теоретически не доказана сходимость к глобальному оптимуму, но эмпирически установлено, что вероятность получения оптимального или квазиоптимального решения высока [2–4].

**Обобщенная схема построения популяционного алгоритма.** На рис. 1 представлена обобщенная схема работы популяционных алгоритмов.

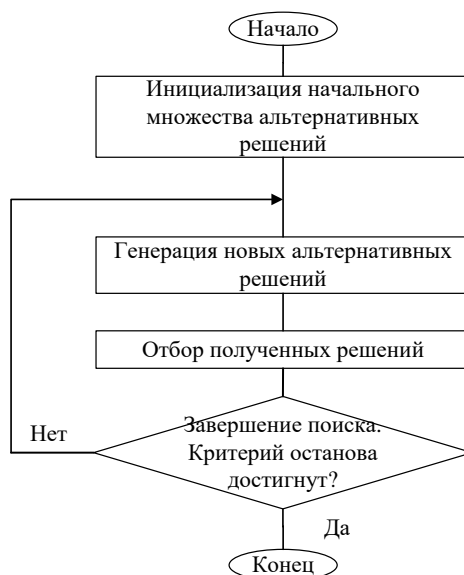


Рис. 1. Обобщенная структура популяционного алгоритма

В общем случае все популяционные алгоритмы включают в себя следующие этапы [4–6].

1. Инициализация начального множества альтернативных решений (популяции). В заданной области поиска создается заданное количество альтернативных решений, которые в разной степени приближены к искомому решению. Для получения начальной популяции могут быть применены детерминированные или случайные алгоритмы. При формировании начальной популяции решениями покрывается по возможности вся область поиска (поиск в ширину). В случае если окрестность глобального экстремума известна, агентами заполняется окрестность и выполняется поиск вблизи глобального экстремума оптимизируемой функции. Такой подход может существенно сократить время решения задачи. Отметим, что при решении реальных задач информация о глобальном экстремуме отсутствует, поэтому целесообразно равномерно распределить агентов по всей области поиска [4].

2. Генерация новых альтернативных решений. С помощью некоторого набора операторов, генерируются новые альтернативные решения таким образом, что каждое новое решение итерационно приближается к оптимуму. Данный набор операторов – математических преобразований, основанных на вероятностном подходе, является особенностью каждого популяционного алгоритма. Например, для бактериального алгоритма – это моделирование движения бактерии *eColi* [7], для генетических алгоритмов (ГА) – скрещивание пар альтернативных решений, названное в парадигме ГА кроссинговером, и мутация для резкого изменения одной или нескольких частей альтернативного решения [2, 4]. Для муравьиных алгоритмов – моделирование поведения колонии муравьев в поисках пищи [8].

3. Отбор полученных решений. Этап генерации новых альтернативных решений в зависимости от значений настроечных параметров увеличивает размер популяции в несколько раз. При этом в популяции в общем случае могут оказаться решения с низкой оценкой целевой функции или даже нелегитимные решения. Так же быстрый рост популяции приводит к резкому ухудшению сходимости алгоритма. Для решения данных проблем вводится этап отбора. На данном шаге используются различные эвристики, с помощью которых осуществляется выборка альтернативных решений. Наглядным примером, является «жадная» эвристика, которая выбирает только лучшие, с точки зрения заданного критерия, решения. На сегодняшний день известно большое множество операторов отбора, как универсальных, так и специализированных, применяемых при решении конкретных прикладных задач [2].

4. Завершение поиска. На данном этапе проверяется выполнение критерия останова алгоритма. Реализация заключается в выборе лучшего из найденных решений, которое принимается за приближенное решение задачи. Известны следующие широко применяемые критерии останова: количество итерации, время работы. Также достаточно часто используют в качестве критерия останова достижение популяцией состояния стагнации алгоритма – такое состояние при котором лучшее достигнутое квазиоптимальное решение неизменно на протяжении заданного числа итераций или времени.

Одним из основных преимуществ популяционных алгоритмов является их модульная структура. Такой подход позволяет быстро получить большое число новых вариантов алгоритма за счет разработки новых и модификации имеющихся правил инициализации, генерации новых агентов и отбора [3, 5].

Для всех популяционных алгоритмов характерны следующие свойства агентов [9–12]:

- ◆ автономность – агенты в пространстве перемещаются независимо друг от друга;
- ◆ стохастичность – процесс создания новых агентов является псевдонаправленным;
- ◆ коммуникабельность – агенты способны обмениваться имеющейся информацией, выявленной в ходе поискового процесса;
- ◆ ограниченность представления – каждый из агентов популяции несет информацию об исследуемой части поискового пространства;
- ◆ децентрализация – отсутствие агентов высшего уровня, управляющих процессом поиска в целом.

Популяционные оптимизационные алгоритмы по сравнению с классическими методами оптимизации имеют ряд преимуществ особенно при решении неформализованных задач и задач высокой размерности. При таких условиях популяционные алгоритмы обеспечивают высокую вероятность нахождения оптимального или квазиоптимального решения за полиномиальное время. На практике приближенное решение часто является достаточным.

Тем не менее при больших объемах входных данных наблюдается значительный рост времени работы популяционных решений. Это связано с тем, что временная сложность данных алгоритмов может варьироваться от  $O(n^2)$  до  $O(n^3)$ .

**Архитектура параллельного поиска.** Для повышения скорости работы популяционных алгоритмов в данной работе предлагается подход к распараллеливанию задачи или ее различных частей [12, 13]. В качестве примера, предлагается модифицированный параллельный генетический алгоритм, основанный на простом генетическом алгоритме (ПГА) для решения классической оптимизационной задачи – задачи коммивояжера.

В общем случае при построении параллельного подхода применительно к популяционным алгоритмам можно выделить 3 уровня распараллеливания [14]:

1. Микроуровень – распараллеливание задач на уровне выполнения операторов генерации новых альтернативных решений.
2. Макроуровень - распараллеливание поиска на уровне генераций (итераций).
3. Метауровень – распараллеливание процесса поиска на уровне изолированного множества альтернативных решений (популяций). В парадигме генетических алгоритмов данный подход успешно реализован на примере оператора миграции [15].

В данной работе предложена реализация параллельного генетического алгоритма с распараллеливанием на микроуровне. Архитектура параллельного поиска представлена на рис. 2.

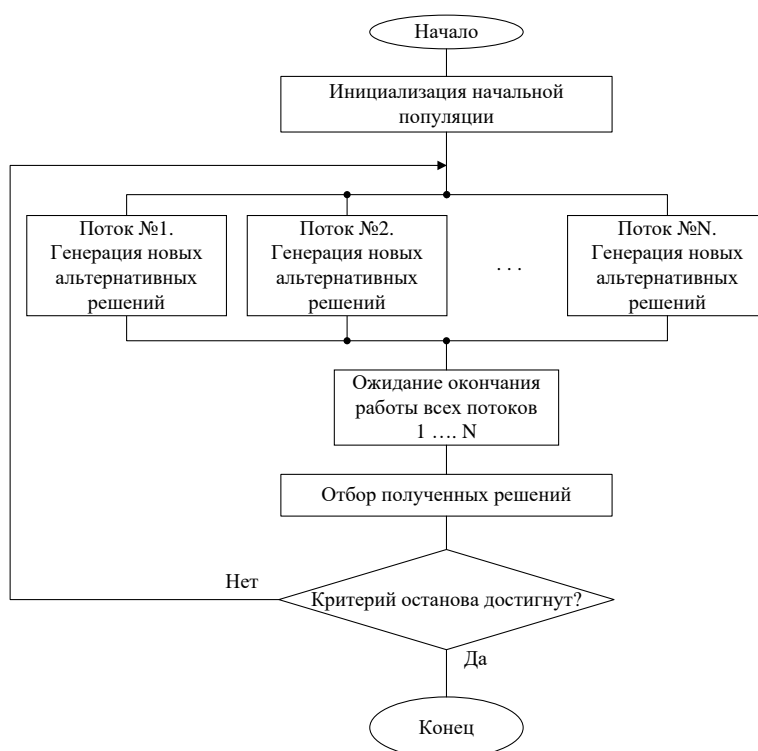


Рис. 2. Архитектура параллельного ГА с распараллеливанием на микроуровне

Алгоритм начинает работу с генерации новой популяции альтернативных решений. Далее на каждой итерации инициализируется  $N$  потоков, в каждом из которых выполняется оператор кроссинговера и с заданной долей вероятности операторы мутации и инверсии. Для всех новых решений производится расчет целевой функции (ЦФ) в том потоке, в котором данное решение было получено. Алгоритм ожидает окончания всех потоков. Далее производится объединение всех новых сгенерированных решений в одну популяцию и выполняется оператор селекции. Процесс продолжается итерационно по достижению заданного числа итераций. В качестве механизма генерации новых альтернативных решений используется циклический оператор кроссинговера, одноточечный оператор мутации и оператор инверсии с возможностью выбора точек начала и конца блока инверсии [16].

Для увеличения скорости сходимости алгоритма в качестве одного из начальных решений используется решение, полученное с помощью «жадного» алгоритма.

**Экспериментальные исследования.** Для подтверждения эффективности предложенного подхода была разработана программная реализация параллельного генетического алгоритма. В качестве прикладной задачи была выбрана одна из классических оптимизационных задач – нахождение Гамильтонова цикла в графе, также известная как задача коммивояжера [16–18]. Целью проведения экспериментальных исследований является скорость работы реализованного генетического алгоритма, а также сравнение его с классическим ПГА [2]. Исследования проводились на двух различных платформах с процессорами: Intel I7 – 4 ядра, 2.7 ГГц и AMD 6000+ – 2 ядра, 2.5 ГГц. Данные экспериментальных исследований сведены в табл. 1. Результаты сравнения времени работы параллельного и простого генетических алгоритмов представлены на рис. 3 и 4. На рис. 5 представлена гистограмма прироста скорости работы параллельного генетического алгоритма относительно простого ГА.

Таблица 1

Данные экспериментальных исследований

AMD 6000+, 2 ядра, 2.5 ГГц				Core I7, 4 ядра, 2.7ГГц			
Параллельный ГА		Простой ГА		Параллельный ГА		Простой ГА	
Время работы, мс	Значение ЦФ, у.е.	Время работы, мс	Значение ЦФ, у.е.	Время работы, мс	Значение ЦФ, у.е.	Время работы, мс	Значение ЦФ, у.е.
3436	11274	566	11987	1412	10703	428	11113
3370	26378	1229	25877	1598	26975	844	26490
5766	88347	5743	91281	2559	89157	3461	91041
13662	206367	17841	202756	5702	205833	12492	206268
46885	446725	82719	430594	17075	441331	48029	446612
190723	927196	412912	926242	57052	936829	189073	932780
357058	1182746	727041	1186145	91067	1192587	298450	1187945
1772381	2427189	3601055	2452883	377670	2448037	1230888	2457078

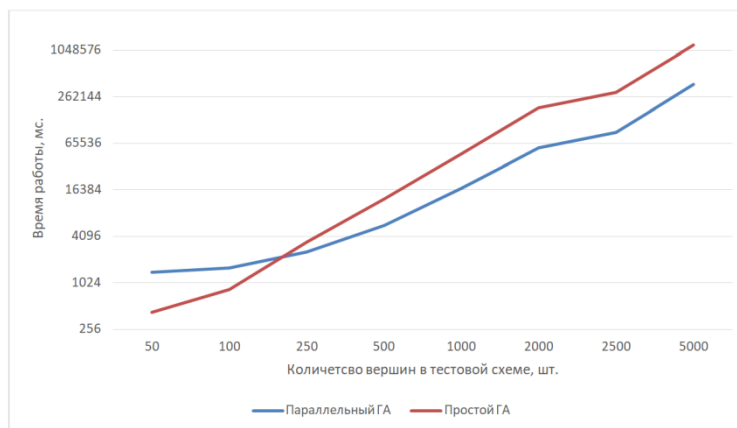


Рис. 3. Сравнение времени работы параллельного и простого ГА на процессоре Core I7, 4 ядра, 2.7ГГц

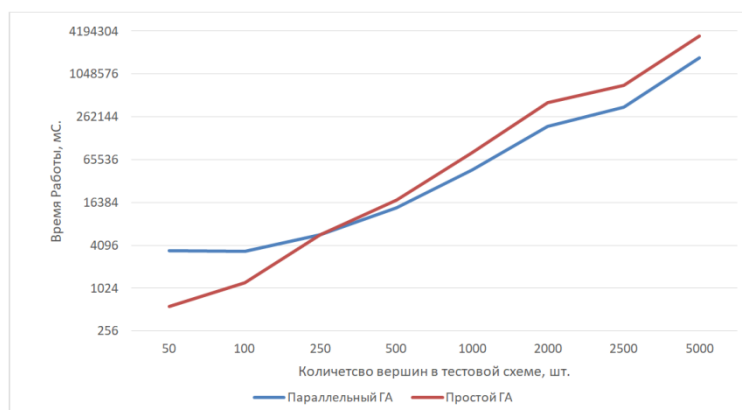


Рис. 4. Сравнение времени работы параллельного и простого ГА на процессоре AMD 6000+, 2 ядра, 2.5ГГц

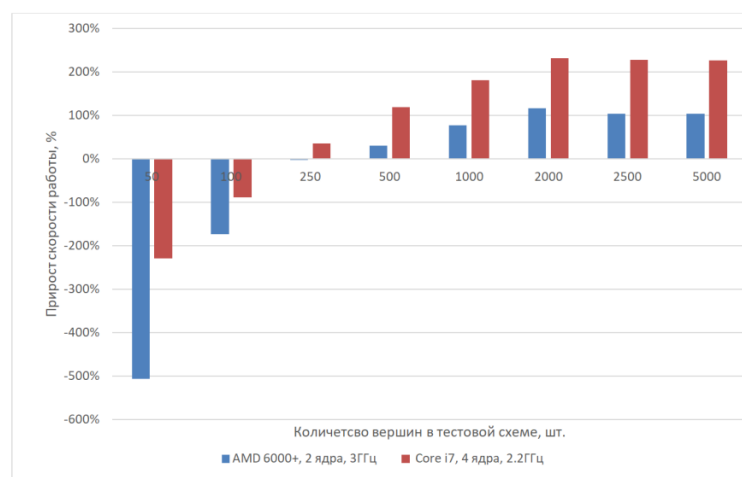


Рис. 5. Прирост скорости параллельного ГА относительно простого ГА

Из рис. 5 видно, что распараллеливание на микроуровне дает прирост в скорости работы для графов с количеством вершин 250 и более. На основании этого можно сделать вывод, что данный подход может быть эффективно применен для оптимизационных задач большой размерности [19–20].

**Вывод.** В данной работе был рассмотрен новый подход к распараллеливанию популяционных алгоритмов. Дана общая структура популяционного подхода, а также описаны различные уровни распараллеливания. В качестве примера рассмотрен простой генетический алгоритм и его параллельная модификация. Реализован параллельный генетический алгоритм с распараллеливанием на микроуровне для решения задачи коммивояжера (построения Гамильтонова цикла в графе). Проведены экспериментальные исследования, которые показали значительный прирост в скорости работы алгоритма при количестве вершин в графе более 250. Скорость работы алгоритма выросла на 100–200 % по сравнению с классической реализацией в зависимости от используемого процессора.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Курейчик В.М., Писаренко В.И., Кравченко Ю.А. Технология многоаспектного аналитического исследования как метод машинного обучения // Открытое образование. – 2008. – № 2. – С. 11-17.
2. Kurejchik V.V., Kurejchik V.M. On genetic-based control // Автоматика и телемеханика. – 2001. – № 10. – С. 174-187.
3. Курейчик В.М., Курейчик В.В., Родзин С.И., Гладков Л.А. Основы теории эволюционных вычислений. – Ростов-на-Дону: ЮФУ, 2010. – 222 с.
4. Курейчик В.В., Курейчик В.М., Зинченко Л.А. и др. Бионические информационные системы и их практические применения. – М.: Физматлит, 2011. – 288 с.
5. Родзин С.И., Курейчик В.В. Состояние, проблемы и перспективы развития биоэвристик // Программные системы и вычислительные методы. – 2016. – № 2. – С. 158-172.
6. Родзин С.И., Курейчик В.В. Теоретические вопросы и современные проблемы развития когнитивных биоинспирированных алгоритмов оптимизации // Кибернетика и программирование. – 2017. – № 3. – С. 51-79.
7. Zaporozhets D., Zaruba D. Bacterial foraging optimization for VLSI fragments placement // Advances in Intelligent Systems and Computing. – 2018. – 679. – P. 341-348.
8. Kureichik V.M., Lebedev B.K., Lebedev O.B. Channel routing based on ant colony adaptive behavior model // Journal of Computer and Systems Sciences International. – 2015. – Vol. 54 (2). – P. 278-293.
9. Kureichik V.V., Kravchenko Y.A. Bioinspired algorithm applied to solve the travelling salesman problem // World Applied Sciences Journal. – 2013. – Vol. 22, No. 12. – P. 1789-1797.
10. Курейчик В.В., Заруба Д.В., Запорожец Д.Ю. Алгоритм параметрической оптимизации на основе модели поведения роя светлячков // Известия ЮФУ. Технические науки. – 2015. – № 6 (167). – С. 6-15.
11. Курейчик В.В., Заруба Д.В., Запорожец Д.Ю. Иерархический подход при размещении компонентов СБИС // Известия ЮФУ. Технические науки. – 2014. – № 7 (156). – С. 75-84.
12. Кулиев Э.В., Курейчик В.В., Курсытис И.О. Принятие решений в задаче размещения компонентов СБИС на основе модели поведения стаи волков // Международная конференция по мягким вычислениям и измерениям. – 2018. – Т. 1. – С. 712-715.
13. Кулиев Э.В., Кравченко Ю.А., Логинов О.А., Запорожец Д.Ю. Метод интеллектуального принятия эффективных решений на основе биоинспирированного подхода // Известия Кабардино-Балкарского научного центра РАН. – 2017. – № 6-2 (80). – С. 162-169.
14. Knysh D.S., Kureichik V.M. Genetic algorithm for routing switching units // Russian Microelectronics. – 2011. – Vol. 40 (7). – P. 486-490.
15. Курейчик В.М., Курейчик В.В., Родзин С.И. Модели параллелизма эволюционных вычислений // Вестник Ростовского государственного университета путей сообщения. – 2011. – № 3 (43). – С. 93-97.
16. Knysh D.S., Kureichik V.M. Parallel genetic algorithms: A survey and problem state of the art // Journal of Computer and Systems Sciences International. – 2010. – Vol. 49 (4). – P. 579-589.
17. Kasprzyk J., Kureichik V.M., Malioukov S.P., Kureichik V.V., Malioukov A.S. Experimental investigation of algorithms developed // Studies in Computational Intelligence. – 2009. – Vol. 212. – P. 211-223+227-236.
18. Gladkov L.A., Kureichik V.V., Kravchenko Y.A. Evolutionary algorithm for extremal subsets comprehension in graphs // World Applied Sciences Journal. – 2013. – Vol. 27, No. 9. – P. 1212-1217.
19. Polkovnikova N.A., Kureichik V.M. Hybrid Expert System Development Using Computer-Aided Software Engineering Tools // Communications in Computer and Information Science. – 466 CCIS. – 2014. – P. 433-445.
20. Gudilov V., Kureichik V. Evolutional synthesis with incomplete information // Life Science Journal, 11 (10 SPEC. ISSUE), art. no. 68. – 2014. – P. 359-363.



## REFERENCES

1. Kureychik V.M., Pisarenko V.I., Kravchenko Yu.A. Tekhnologiya mnogoaspektного analiticheskogo issledovaniya kak metod mashinnogo obucheniya [Technology multidimensional analytical studies as a method of machine learning], *Otkrytoe obrazovanie* [Open education], 2008, No. 2, pp. 11-17.
2. Kureychik V.V., Kureychik V.M. On genetic-based control, *Avtomatika i telemekhanika* [Automation and remote control], 2001, No. 10, pp. 174-187.
3. Kureychik V.M., Kureychik V.V., Rodzin S.I., Gladkov L.A. Osnovy teorii evolyutsionnykh vychisleniy [Fundamentals of the theory of evolutionary computation]. Rostov-on-Don: YuFU, 2010, 222 p.
4. Kureychik V.V., Kureychik V.M., Zinchenko L.A. i dr. Bionicheskie informatsionnye sistemy i ikh prakticheskie primeneniya [Bionic information systems and their practical applications]. Moscow: Fizmatlit, 2011, 288 p.
5. Rodzin S.I., Kureychik V.V. Sostoyanie, problemy i perspektivy razvitiya bioevristik [State, problems and prospects of development of bio-heuristics], *Programmnyye sistemy i vychislitel'nye metody* [Software systems and computational methods], 2016, No. 2, pp. 158-172.
6. Rodzin S.I., Kureychik V.V. Teoreticheskie voprosy i sovremennyye problemy razvitiya kognitivnykh bioinspirirovannykh algoritmov optimizatsii [Theoretical questions and contemporary problems of the development of cognitive bio-inspired algorithms for optimization], *Kibernetika i programmirovaniye* [Cybernetics and programming], 2017, No. 3, pp. 51-79.
7. Zaporozhets D., Zaruba D. Bacterial foraging optimization for VLSI fragments placement, *Advances in Intelligent Systems and Computing*, 2018, 679, pp. 341-348.
8. Kureichik V.M., Lebedev B.K., Lebedev O.B. Channel routing based on ant colony adaptive behavior model, *Journal of Computer and Systems Sciences International*, 2015, Vol. 54 (2), pp. 278-293.
9. Kureichik V.V., Kravchenko Y.A. Bioinspired algorithm applied to solve the travelling salesman problem, *World Applied Sciences Journal*, 2013, Vol. 22, No. 12, pp. 1789-1797.
10. Kureychik V.V., Zaruba D.V., Zaporozhets D.Yu. Algoritm parametricheskoy optimizatsii na osnove modeli povedeniya roya svetlyachkov [The algorithm of parametric optimization based on the behavior patterns of a swarm of fireflies], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2015, No. 6 (167), pp. 6-15.
11. Kureychik V.V., Zaruba D.V., Zaporozhets D.Yu. Ierarkhicheskiy podkhod pri razmeshchenii komponentov SBIS [A hierarchical approach to the placement of VLSI components], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2014, No. 7 (156), pp. 75-84.
12. Kuliev E.V., Kureychik V.V., Kursitya I.O. Prinyatie resheniy v zadache razmeshcheniya komponentov SBIS na osnove modeli povedeniya stai volkov [Decision-making in the problem of placing VLSI components based on the behavior model of a pack of wolves], *Mezhdunarodnaya konferentsiya po myagkim vychisleniyam i izmereniyam* [International conference on soft computing and measurements], 2018, Vol. 1, pp. 712-715.
13. Kuliev E.V., Kravchenko Yu.A., Loginov O.A., Zaporozhets D.Yu. Metod intellektual'nogo prinyatiya effektivnykh resheniy na osnove bioinspirirovannogo podkhoda [Method of intellectual decision-making based on bioinspired approach], *Izvestiya Kabardino-Balkarskogo nauchnogo tsentra RAN* [Izvestiya of Kabardino-Balkar scientific center of RAS], 2017, No. 6-2 (80), pp. 162-169.
14. Knysh D.S., Kureichik V.M. Genetic algorithm for routing switching units, *Russian Microelectronics*, 2011, Vol. 40 (7), pp. 486-490.
15. Kureychik V.M., Kureychik V.V., Rodzin S.I. Modeli parallelizma evolyutsionnykh vychisleniy. [Models of parallelism of evolutionary calculations], *Vestnik Rostovskogo gosudarstvennogo universiteta putey soobshcheniya* [Bulletin of Rostov state University of railway engineering], 2011, No. 3 (43), pp. 93-97.
16. Knysh D.S., Kureichik V.M. Parallel genetic algorithms: A survey and problem state of the art, *Journal of Computer and Systems Sciences International*, 2010, Vol. 49 (4), pp. 579-589.
17. Kacprzyk J., Kureichik V.M., Malioukov S.P., Kureichik V.V., Malioukov A.S. Experimental investigation of algorithms developed, *Studies in Computational Intelligence*, 2009, Vol. 212, pp. 211-223+227-236.

18. Gladkov L.A., Kureichik V.V., Kravchenko Y.A. Evolutionary algorithm for extremal subsets comprehension in graphs, *World Applied Sciences Journal*, 2013, Vol. 27, No. 9, pp. 1212-1217.
19. Polkovnikova N.A., Kureichik V.M. Hybrid Expert System Development Using Computer-Aided Software Engineering Tools, *Communications in Computer and Information Science*, 466 CCIS, 2014, pp. 433-445.
20. Gudilov V., Kureichik V. Evolutional synthesis with incomplete information, *Life Science Journal*, 11 (10 SPEC. ISSUE), art. no. 68, 2014, pp. 359-363.

Статью рекомендовал к опубликованию д.т.н., профессор Ю.А. Гатчин.

**Запорожец Дмитрий Юрьевич** – Южный федеральный университет; e-mail: elpilasgsm@gmail.com; 347928, г. Таганрог, пер. Некрасовский, 44; тел.: 88634371651; кафедра систем автоматизированного проектирования; доцент.

**Заруба Дарья Викторовна** – e-mail: daria.zaruba@gmail.com; кафедра систем автоматизированного проектирования; ассистент.

**Zaporoghetz Dmitri Yurievich** – Southern Federal University; e-mail: elpilasgsm@gmail.com; 44, Nekrasovskiy, Taganrog, 347928, Russia; phone: +78634371651; the department of computer aided design; associate professor.

**Zaruba Darya Viktorovna** – e-mail: daria.zaruba@gmail.com; the department of computer aided design; assistant.

УДК 004.032.26

DOI 10.23683/2311-3103-2018-4-145-153

**М.В. Сычугов**

### **МОДЕЛИРОВАНИЕ ПРОГНОЗА МЕТРИКИ УДЕРЖАНИЯ ПОЛЬЗОВАТЕЛЕЙ НА ОСНОВЕ НЕЙРОННЫХ СЕТЕЙ**

*Данная статья посвящена решению проблемы моделирования прогноза метрики удержания в области мобильной разработки, где возникла острая необходимость своевременное иметь данные в какой момент времени пользователи уходят из приложения для принятия мер по увеличению времени жизни и уровня монетизационных показателей. Так как в большинстве приложений значительный отток пользователей случается в начальном интервале времени пользования, то своевременное упреждение узких мест в приложении является полезным как в коммерческом плане, так и для исследования пользовательского поведения, которое в свою очередь позволит разрабатывать более рентабельный продукт. Цель работы состоит в разработке модели прогнозирования метрик достаточно близких к реальным показателям. Предложенная модель на основе нейронной сети, обученной методом обратного распространения ошибки, предоставляет возможность прогнозировать различные временные периоды по отношению к числу оставшихся пользователей в приложении. Метод обратного распространения ошибки выбран из-за его достаточной эффективности и высоко-адаптируемой архитектуры с произвольным числом слоёв, входов и выходов. Научная новизна представлена комбинированием нейронных сетей с эвристиками на основе ключевых категорий данных. Эти данные представляют собой сегментацию мобильных пользователей на различные категории, которые позволяют уловить их аспекты поведения для более адаптируемого прогноза удержания разных целевых групп. Результаты прогноза приведены для разных временных интервалов, характерных для использования мобильного приложения.*

*Нейронные сети; метод обратного распространения ошибки; удержание мобильных пользователей; системы прогноза; временные ряды.*