

Э.В. Кулиев, Д.Ю. Запорожец, Д.Ю. Терешенко

МОДИФИЦИРОВАННЫЙ АЛГОРИТМ ВОЛЧЬЕЙ СТАИ РЕШЕНИЯ КОНСТРУКТОРСКИХ ЗАДАЧ*

Рассматривается основная проблема искусственного интеллекта - разработка эффективных новых и модифицированных эвристических механизмов поиска оптимального решения. В настоящее время одним из перспективных направлений развития искусственного интеллекта служат вопросы применения методов и моделей поведения биологических систем для решения NP-полных и NP-трудных оптимизационных задач. В статье рассмотрена одна из актуальных задач конструкторского проектирования сверхбольших интегральных схем (СБИС) - задача размещения. Представлена постановка задачи размещения элементов СБИС. Задачу размещения элементов СБИС предложено решать на основе поведения биологических систем в природе, на примере алгоритма волчьей стаи. Волки являются типичными социальными животными, имеющими четкое разделение социальной работы. Представлены действия и правила поведения волчьей стаи в живой природе. На основе представленных правил и действий волков, описан модифицированный алгоритм волчьей стаи. Преимуществом разработанного модифицированного алгоритма является возможность улучшения каждой последующей стадии решения задачи размещения. В рамках работы алгоритм волчьей стаи был реализован на языке Java. В качестве тестовых схем использовались известные бенчмарки фирмы IBM. Сравнения проводились с результатами работы известных алгоритмов: Capo 8.6, Feng Shui 2.0, Dragon 2.23. На основе проведенных исследований, разработанный алгоритм волчьей стаи показал результаты выше, по сравнению с аналогами.

Конструкторское проектирование; размещение; волчья стая; биоинспирированный алгоритм; САПР.

E.V. Kuliev, D.Yu. Zaporozhets, D.Yu. Tereshenko

A MODIFIED WOLF PACK ALGORITHM FOR SOLVING DESIGN TASKS

The paper deals with the main problem of artificial intelligence - the development of effective new and modified heuristic mechanisms for finding the optimal solution. Currently, one of the promising areas for the development of artificial intelligence is the use of methods and models of the behavior in biological systems for solving NP-complete and NP-difficult optimization problems. The article considers one of the urgent problems of the design of ultra-large-scale integrated circuits (VLSI) - the problem of placement. Presented is the formulation of the problem of placement of elements of VLSI. The task of locating the VLSI elements is proposed to be solved on the basis of the behavior of biological systems in nature, using the wolf pack algorithm as an example. Wolves are typical social animals that have a clear division of social work. Presents the actions and rules of behavior of the wolf pack in wildlife. Based on the presented rules and actions of wolves, a modified algorithm of the wolf pack is described. The advantage of the developed modified algorithm is the possibility of improving each subsequent stage of solving the placement problem. As part of the work, the wolf pack algorithm was implemented in Java. Famous IBM benchmarks were used as test patterns. Comparisons were made with the results of the work of well-known algorithms: Capo 8.6, Feng Shui 2.0, Dragon 2.23. Based on the research, the algorithm developed by the wolf pack showed higher results compared to peers.

Design engineering; placement; wolf pack; bioinspired algorithm; CAD.

Введение. Системы автоматизированного проектирования (САПР) определяют в настоящее время прогресс науки и техники. Построение интеллектуальных САПР связано с разработкой математического обеспечения. Оно включает мате-

* Работа выполнена при финансовой поддержке гранта Президента Российской Федерации для государственной поддержки молодых российских ученых - кандидатов наук № МК-1480.2018.9.

математические модели, методы и алгоритмы интеллектуального проектирования. Важной составляющей интеллектуальных САПР, результаты которого играют основную составляющую в производстве, являются САПР конструкторского проектирования. В данных САПР электрические схемы являются входящей информацией, а топология схемы выходящей.

Конструкторское проектирование является ключевых этапов, который определяет эффективность и качество создаваемой конструкции, и является наиболее проблемным, так как приходится обрабатывать огромные массивы данных и информации [1, 6]. В условиях современного развития информационных технологий возникает проблема перехода на новые проектные нормы. Существующие методы и алгоритмы конструкторского проектирования не справляются с решением данной задачи. Трудоемкость решения задач конструкторского проектирования в настоящее время резко возросла и разработчикам СБИС необходимы эффективные программные системы, реализующие наиболее совершенные и довольно сложные эффективные алгоритмы [2, 5].

В последнее время активно развивается научное направление, связанное с моделями поведения биологических систем в живой природе. Модели поведения биологических систем в живой природе объединяют математические методы, в которые заложены основные принципы природных механизмов принятия решений [3,4]. На основе принципов и механизмов поведения биологических систем в работе разработан модифицированный биоинспирированный алгоритм волчьей стаи.

1. Постановка задачи. Основная цель размещения - организовать элементы таким образом, чтобы площадь кристалла была минимальна. А также минимизировать общую длину межсоединений для уменьшения временных задержек, возникающих в длинных цепях, тем самым увеличив скорость обработки информации в СБИС [7, 8]. Создание необходимых и достаточных условий для трассировки.

КП организовано как регулярная структура. Ячейки (слоты) организованы как матрица строк и столбцов. Внешние ячейки ввода / вывода расположены вокруг. Области между слотами предназначены для маршрутизации (рис. 1).

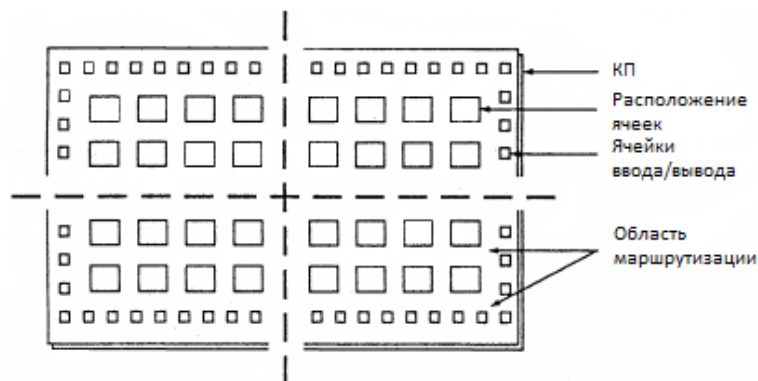


Рис. 1. Макет матрицы

При ограниченном размере кристалла, который задаётся вручную, одной из важных задач является размещение элементов таким образом, чтобы они не пересекались и не накладывались друг на друга. Обычно для оценки качества размещения используют функцию качества, в которую включают оценки суммарной длины соединений и различные штрафы, включая пересечение или наложение элементов, а также описанные выше критерии [9].

В общем виде задача размещения формулируется следующим образом: в монотажном пространстве, задаётся область, которая разбивается на некоторое множество позиций (также применяется термин посадочных мест) $p = \{p_1, p_2, \dots, p_q\}$, число которых должно равняться или быть больше числа размещаемых элементов. Каждый элемент может занимать только одно посадочное место. Расстояние между посадочными местами описывается симметричной матрицей расстояний $D = \|d_{ij}\|$. Имеющееся множество элементов $X = \{x_1, x_2, \dots, x_n\}$, которые связывается с множеством электрических цепей $E = \{e_1, e_2, \dots, e_n\}$, необходимо отобразить на множестве P таким образом, чтобы был достигнут экстремум целевой функции [10, 11].

2. Описание биоинспирированного алгоритма, основанного на поведении стаи волков. Для эффективного решения задачи размещения элементов СБИС, авторами предложено использовать модифицированный алгоритм волчьей стаи. Пространство искусственных волков можно обозначить как евклидово пространство $N \times D$, где N – число волков, а D – количество переменных. Вектор $X_i = (x_i^1, x_i^2, \dots, x_i^D)$ представляет собой положение i -го искусственного волка, а x_i^d – это d -я переменная величины X_i . $Y = f(X)$ представляет целевое значение функции X , которое можно рассматривать как концентрацию запаха жертвы, воспринимаемого искусственными волками [12–15].

Волки являются типичными социальными животными, имеющими четкое разделение социальной работы. Волки в АВС можно разделить на три категории: волк-вожак, волки-разведчики и дикие волки. Все хищное поведение волчьей стаи сводится к трем интеллектуальным поведением - разведке, охоте и окружению. Также существуют еще два правила: "победитель получает все" для вожака и правило обновления "выживает сильнейший" для остальной стаи. Далее приведено подробное описание этих действий и правил.

1) *Правило генерации "победитель получает все" для вожака* [12–15]:

Вожак – волк с наилучшей целевой функцией. Таким образом, позицию вожака можно рассматривать как позицию жертвы. Во время каждой итерации значение функции вожака будет сравниваться со значением функции другого волка; если значение Y_{lead} не самое лучшее, то вожак будет заменен. Тогда лучший волк становится новым вожаком. Вожаку не нужно выполнять три интеллектуальных поведения, он сразу переходит в следующую итерацию, пока другой, лучший волк его не заменит.

2) *Разведка* [12–15]:

Не считая вожака, несколько элитных волков действуют как волки-разведчики. У волков-разведчиков значения функций лучшие, чем у диких волков. Y_i и Y_{lead} представляют собой целевые значения функций волка-разведчика и вожака соответственно. Если $Y_i > Y_{lead}$, это означает, что значение вожака не лучшее и вожак будет заменен волком-разведчиком, т.е. $Y_i = Y_{lead}$. Если $Y_i < Y_{lead}$, волки-разведчики будут действовать следующим образом.

Сначала i -ый волк-разведчик пытается сделать шаг в сторону h различными путями и запомнить значения функции в каждом направлении. После шага в i -ом направлении состояние i -го разведчика формулируется с помощью уравнения (7):

$$x_{i,d}^p = x_{i,d} + \sin\left(2\pi * \frac{p}{h}\right) * step_a.$$

Затем волк-разведчик выбирает направление, при котором значение целевой функции наилучшее и обновляет позицию X_i . После этого поведение разведчика повторяется, пока $Y_i > Y_{lead}$ или не достигнуто максимальное количество повторений T_{max} . Следует отметить, что h для каждого волка различно и является целым числом, выбранным из $[h_{min}, h_{max}]$. Оно направлено на стимулирование различных стратегий охоты у разных волков. Переменная $step_a$ – это длина шага.

3) *Охота (погона, преследование)* [12–15]:

Вожак воеет и зазывает оставшихся диких волков собраться вокруг добычи. Здесь, позиция вожака рассматривается как возможная позиция добычи, чтобы волки собрались к вожаку. Переменная $step_b$ – длина шага; g_d^k – положение вожака в i -ом переменном пространстве на $-ой$ итерации, k – число итераций. Позиция дикого волка обновляется согласно следующему уравнению:

$$x_{i,d}^{k+1} = x_{i,d}^k + step_b * \frac{g_d^k - x_{i,d}^k}{|g_d^k - x_{i,d}^k|}$$

Если $Y_i > Y_{lead}$, волк становится вожаком и $Y_i = Y_{lead}$; в таком случае новый вожак принимает охоту на себя. Если $Y_i < Y_{lead}$, волки продолжают группироваться вокруг вожака с большой скоростью, пока $L(i, lead) < L_{near}$, а затем волк примет осаждающее поведение. $L(i, lead)$ представляет собой расстояние между вожаком и i -ым волком; L_{near} – это расстояние окружения, которое можно рассматривать в качестве условия суждения. Оно определяет, изменяет ли волк поведение с погони на окружение. L_{near} формулируется уравнением:

$$L_{near} = \frac{1}{D \cdot w} * \sum_{d=1}^D |max_d - min_d|,$$

где w – коэффициент определения расстояния, а $[min_d, max_d]$ – диапазон значения переменной d .

4) *Окружение добычи* [12–15]:

После движения большими шагами по направлению к вожаку, дикие волки близки к добыче. Затем они изменяют свое поведение с погони на окружение добычи. Положение i -го дикого волка обновляется в соответствии со следующим уравнением:

$$x_{i,d}^{k+1} = x_{i,d}^k + \lambda \cdot step_c * |g_d^k - x_{i,d}^k|,$$

где λ – случайное число, лежащее в интервале $[-1,1]$; $step_c$ – это длина шага при окружении добычи. Если $Y_i^{new} > Y_i^{old}$ после того, как волк начал окружать добычу, то положение X_i обновляется; в противном случае его не следует изменять. Взаимосвязь $step_a$, $step_b$ и $step_c$ реализована следующим образом:

$$step_a = \frac{step_b}{2} = 2 * step_c = S,$$

где S – это коэффициент шага и представляет собой сложную степень искусственных волков, охотящихся за добычей в пространстве решений.

5) *Правило обновления волчьей стаи "выживает сильнейший"*:

Это правило имитирует естественный отбор Дарвина. По мере того, как добыча распределяется от сильных к слабым особям, это приведет к гибели некоторых слабых волков. Алгоритм будет регенерировать волков случайным образом при удалении R волков, которые недостаточно хороши. R представляет собой целое число и случайным образом выбирается в интервале $[N/2\beta, N/\beta]$. β – пропорциональный коэффициент обновления популяции.

3. Модифицированный алгоритм волчьей стаи. Представим разработанный авторами модифицированный алгоритм стаи волков в виде словесного описания по шагам (рис. 2):

1. Инициализация популяции серых волков X_i ($i = 1, 2, \dots, n$).
2. Инициализация параметров a , A и C .
3. Оценка пригодность каждого агента поиска:
 - 3.1. Инициализация первого лучшего решения как X_α , второго лучшего решения как X_β , третьего лучшего решения как X_δ .
4. Максимальное количество итераций задается в начальных данных.

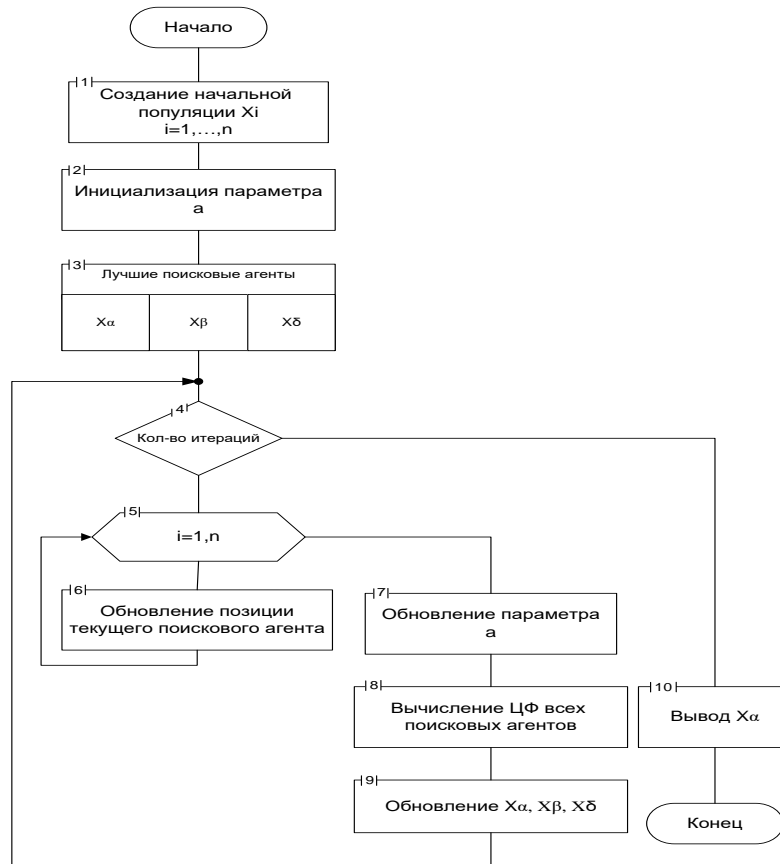


Рис. 2. Модифицированный алгоритм волчьей стаи

5. Счетчик.
6. Обновление позиции поисковых агентов.
7. Обновление параметров a , A и C .
8. Оценка функции пригодности всех поисковых агентов.
9. Обновление лучших агентов X_α , X_β , X_δ .

10. Завершение алгоритма и визуализация значения первого лучшего агента X_α , найденного на данный момент.

4. Экспериментальные исследования. В рамках данной работы модифицированный алгоритм был реализован на языке Java. Были проведены две серии экспериментов. Основные цели проведения вычислительных экспериментов, следующие [16, 17]: выявление общего улучшения результатов размещения в процесс поиска; нахождение лучших характеристик по времени работы и значению целевой функции.

В качестве тестовых схем использовались известные бенчмарки фирмы IBM [18]. Целью проведения экспериментов является исследования качества решений, полученных при использовании разработанного алгоритма волчьей стаи. Сравнения проводились с результатами работы известных алгоритмов: Capo 8.6 [19], Feng Shui 2.0 [19], Dragon 2.23 [20]. Исследования проводились на базе процессора: Intel I7 – 4 ядра, 2,7 ГГц. Данные экспериментальных исследований отображены на рис. 3. Результаты сравнения качества полученных решений представлены на рис. 4 и 5.

Тестовые схемы		Сапо 8.6	Feng Shui 2.0	Dragon 2.23	Алгоритм волчьей стаи						
Название	Число элементов	длина, м	длина, м	длина, м	длина, м	length increment (Сапо 8.6), m	length increment (Сапо 8.6), %	length increment (Feng Shui 2.0), m	length increment (Feng Shui 2.0), %	length increment (Dragon 2.23), m	length increment (Dragon 2.23), %
		ibm01	12752	4,97	4,87	4,42	4,15	-0,82	-19,76%	-0,72	-17,35%
ibm02	19601	15,23	14,38	13,57	12,4	-2,83	-22,82%	-1,98	-15,97%	-1,17	-9,44%
ibm03	23136	14,06	12,84	12,33	11,44	-2,62	-22,90%	-1,4	-12,24%	-0,89	-7,78%
ibm04	27507	18,13	16,69	15,41	15,5	-2,63	-16,97%	-1,19	-7,68%	0,09	0,58%
ibm05	29347	44,73	37,3	36,38	35,66	-9,07	-25,43%	-1,64	-4,60%	-0,72	-2,02%
ibm06	32498	21,96	20,27	20,38	18,5	-3,46	-18,70%	-1,77	-9,57%	-1,88	-10,16%
ibm07	45926	36,06	31,5	29,97	31,5	-4,56	-14,48%	0	0,00%	1,53	4,86%
ibm08	51309	37,89	34,14	32,2	29,01	-8,88	-30,61%	-5,13	-17,68%	-3,19	-11,00%
ibm09	53395	30,28	29,86	28,1	25,41	-4,87	-19,17%	-4,45	-17,51%	-2,69	-10,59%
ibm10	69429	61,25	57,99	57,2	53,1	-8,15	-15,35%	-4,89	-9,21%	-4,1	-7,72%
ibm11	70558	46,45	43,28	40,77	38,6	-7,85	-20,34%	-4,68	-12,12%	-2,17	-5,62%
ibm12	71076	81,55	75,91	71,03	70,41	-11,14	-15,82%	-5,5	-7,81%	-0,62	-0,88%

Рис. 3. Данные экспериментальных исследований

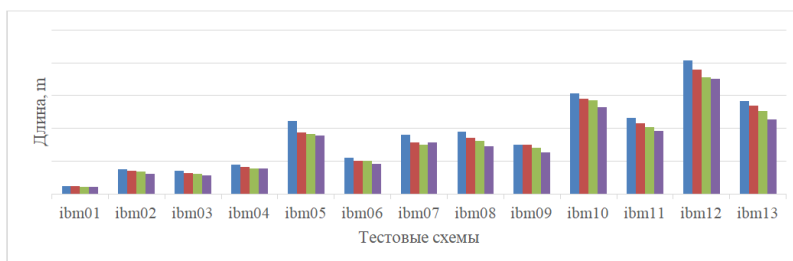


Рис. 4. График зависимости качества решений от используемого алгоритма

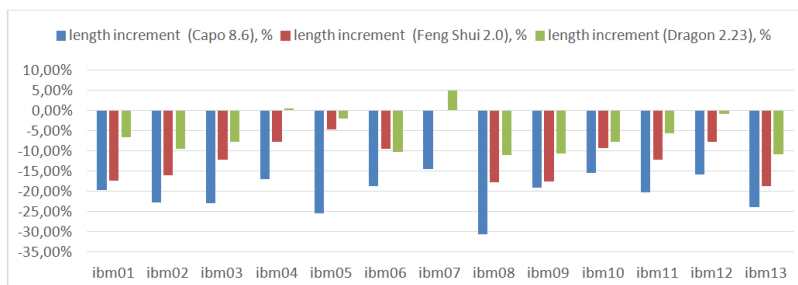


Рис. 5. Гистограмма зависимости приращеня значения ЦФ алгоритма волчьей стаи от используемого аналога

Проанализировав графики зависимости, можно сделать вывод, что качество размещений, полученных разработанным авторами модифицированным алгоритмом, в среднем на 4 % превосходит результаты размещения, полученные с использованием известных алгоритмов Capo 8.6, Feng Shui 2.0, Dragon 2.23, что говорит об эффективности предложенного подхода.

Заключение. Приведена обобщенная постановка задачи размещения элементов СБИС. Описаны действия и правила поведения волчьей стаи в живой природе. Для эффективного решения задачи размещения элементов СБИС, на основе описанных правил и действий поведения волков, разработан модифицированный алгоритм волчьей стаи, позволяющий уменьшать размерность решаемой задачи и получать наборы квазиоптимальных решений за полиномиальное время. В рамках работы модифицированный алгоритм волчьей стаи был реализован на языке Java. В качестве тестовых схем использовались известные бенчмарки фирмы IBM. Сравнения проводились с результатами работы известных алгоритмов: Capo 8.6, Feng Shui 2.0, Dragon 2.23. Проведенные серии тестов и экспериментов позволили уточнить теоретические оценки временной сложности алгоритмов размещения и их поведение для схем различной структуры. В лучшем случае временная сложность алгоритмов $\approx O(n \log n)$, в худшем случае - $O(n^3)$.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Норенков И.П.* Основы автоматизированного проектирования. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2010.
2. *Карпенко А.П.* Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой: учеб. пособие. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2014.
3. *Курейчик В.В., Запорожец Д.Ю.* Современные проблемы при размещении элементов СБИС // Известия ЮФУ. Технические науки. – 2011. – № 7 (120). – С. 68-73.
4. *Sherwani N.A.* Algorithms for VLSI Physical Design Automation. – Third Edition, Kluwer Academic Publisher, USA, 2013.
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 (обзор) // Кибернетика и программирование. – 2017. – № 3. – С. 51-79.
7. *Курейчик В.В., Запорожец Д.Ю.* Современные проблемы при размещении элементов СБИС // Известия ЮФУ. Технические науки. – 2011. – № 7 (120). – С. 68-73.
8. *Kureichik, V., Jr., Kureichik, V., Bova, V.* Placement of VLSI fragments based on a multi-layered approach // Advances in Intelligent Systems and Computing. – 2016. – Vol. 464. – P. 181-190.
9. *Кулиев Э.В., Лежебоков А.А.* О гибридном алгоритме размещения компонентов СБИС // Известия ЮФУ. Технические науки. – 2012. – № 11 (136). – С. 188-192.
10. *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.
11. *Курейчик В.В., Бова В.В., Курейчик Вл.Вл.* Биоинспирированный поиск в задачах конструкторского проектирования и оптимизации // Информационные технологии в науке, образовании и управлении / под ред. проф. Е.Л. Глорнозова. – 2015. – С. 427-432.
12. *Madadi M. Motlagh.* Optimal Control of DC motor using Grey Wolf Optimizer Algorithm - Technical Journal of Engineering and Applied Science. – 2014-4-04/373-379.
13. *Rezaei H., Bozorg-Haddad O., Chu X.* Grey wolf optimization (GWO) algorithm // Studies in Computational Intelligence. – 2018. – Vol. 720. – P. 81-91.
14. *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.

15. Кулиев Э.В., Щеглов С.Н., Пантелюк Е.А., Кулиева Н.В. Адаптивный алгоритм стаи серых волков для решения задач проектирования // Известия ЮФУ. Технические науки. – 2017. – № 7 (192). – С. 28-38.
16. Кулиев Э.В., Лежебоков А.А. О гибридном алгоритме размещения компонентов СБИС // Известия ЮФУ. Технические науки. – 2012. – № 11 (136). – С. 188-192.
17. Кулиев Э.В., Лежебоков А.А. Исследование характеристик гибридного алгоритма размещения // Известия ЮФУ. Технические науки. – 2013. – № 3 (140). – С. 255-261.
18. IBM-PLACE 2.0 benchmark suits. – <http://er.cs.ucla.edu/benchmarks/ibm-place2/bookshelf/ibm-place2-all-bookshelf-nopad.tar.gz>.
19. Adya, S.N., Markov, I.L. Combinatorial techniques for mixed-size placement // ACM Transactions on Design Automation of Electronic Systems. – 2005. – No. 10 (1). – P. 58-90.
20. Wang M., Yang X., Sarrafzadeh M. Dragon2000: Standard-cell Placement Tool for Large Industry Circuits – ICCAD2000. – P. 260-263.

REFERENCES

1. Norenkov I.P. Osnovy avtomatizirovannogo proektirovaniya [Fundamentals of computer-aided design]. Moscow: Izd-vo MGTU im. N.E. Baumana, 2010.
2. Karpenko A.P. Sovremennye algoritmy poiskovoy optimizatsii. Algoritmy, vdokhnovlennyye prirodoy: ucheb. posobie [Modern search engine optimization algorithms. Algorithms inspired by nature: a training manual]. Moscow: Izd-vo MGTU im. N.E. Baumana, 2014.
3. Kureychik V.V., Zaporozhets D.Yu. Sovremennyye problemy pri razmeshchenii elementov SBIS [Modern problems in the placement of VLSI elements], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2011, No. 7 (120), pp. 68-73.
4. Sherwani N.A. Algorithms for VLSI Physical Design Automation. Third Edition, Kluwer Academic Publisher, USA, 2013.
5. Rodzin S.I., Kureychik V.V. Sostoyaniye, problemy i perspektivy razvitiya bioevristik [Status, problems and prospects of bio-heuristics], *Programmnyye sistemy i vychislitel'nyye metody* [Software systems and computational methods], 2016, No. 2, pp. 158-172.
6. Rodzin S.I., Kureychik V.V. Teoreticheskiye voprosy i sovremennyye problemy razvitiya kognitivnykh bioinspirirovannykh algoritmov optimizatsii (obzor) [Theoretical questions and contemporary problems of the development of cognitive bio-inspired optimization algorithms (review)], *Kibernetika i programmirovaniye* [Cybernetics and programming], 2017, No. 3, pp. 51-79.
7. Kureychik V.V., Zaporozhets D.Yu. Sovremennyye problemy pri razmeshchenii elementov SBIS [Modern problems in the placement of VLSI elements], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2011, No. 7 (120), pp. 68-73.
8. Kureichik, V., Jr., Kureichik, V., Bova, V. Placement of VLSI fragments based on a multilayered approach, *Advances in Intelligent Systems and Computing*, 2016, Vol. 464, pp. 181-190.
9. Kuliev E.V., Lezhebokov A.A. O gibridnom algoritme razmeshcheniya komponentov SBIS [On hybrid algorithm of VLSI components placement], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2012, No. 11 (136), pp. 188-192.
10. 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 on the basis of bioinspired approach], *Izvestiya Kabardino-Balkarskogo nauchnogo tsentra RAN* [Izvestiya of Kabardino-Balkar scientific center of RAS], 2017, No. 6-2 (80), pp. 162-169.
11. Kureychik V.V., Bova V.V., Kureychik V.V. Bioinspirirovannyy poisk v zadachakh konstruktorskogo proektirovaniya i optimizatsii [Bio-inspired search in problems of engineering design and optimization], *Informatsionnyye tekhnologii v nauke, obrazovanii i upravlenii* [Information technologies in science, education and management], ed. by E.L. Glorizova, 2015, pp. 427-432.
12. Madadi M. Motlagh. Optimal Control of DC motor using Grey Wolf Optimizer Algorithm - Technical Journal of Engineering and Applied Science. 2014-4-04/373-379.
13. Rezaei H., Bozorg-Haddad O., Chu X. Grey wolf optimization (GWO) algorithm, *Studies in Computational Intelligence*, 2018, Vol. 720, pp. 81-91.

14. 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 placement of 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.
15. Kuliev E.V., Shcheglov S.N., Pantelyuk E.A., Kulieva N.V. Adaptivnyy algoritm stai serykh volkov dlya resheniya zadach proektirovaniya [The adaptive algorithm of the pack of gray wolves for the decision of tasks of design], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2017, No. 7 (192), pp. 28-38.
16. Kuliev E.V., Lezhebokov A.A. O gibridnom algoritme razmeshcheniya komponentov SBIS [On hybrid algorithm of VLSI components placement], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2012, No. 11 (136), pp. 188-192.
17. Kuliev E.V., Lezhebokov A.A. Issledovanie kharakteristik gibridnogo algoritma razmeshcheniya [The study of the characteristics of hybrid positioning algorithm], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2013, No. 3 (140), pp. 255-261.
18. IBM-PLACE 2.0 benchmark suits. <http://er.cs.ucla.edu/benchmarks/ibm-place2/bookshelf/ibm-place2-all-bookshelf-nopad.tar.gz>.
19. Adya, S.N., Markov, I.L. Combinatorial techniques for mixed-size placement, *ACM Transactions on Design Automation of Electronic Systems*, 2005, No. 10 (1), pp. 58-90.
20. Wang M., Yang X., Sarrafzadeh M. Dragon2000: Standard-cell Placement Tool for Large Industry Circuits – ICCAD2000, pp. 260-263.

Статью рекомендовал к опубликованию д.т.н., профессор Н.И. Витиска.

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

Запорожец Дмитрий Юрьевич – e-mail: duzaporozhets@sfnu.ru; кафедра систем автоматизированного проектирования; доцент.

Терешенко Дмитрий Юрьевич – e-mail: tereshenko@sfnu.ru; кафедра систем автоматизированного проектирования; аспирант.

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

Zaporozhets Dmitriy Yurievich – e-mail: duzaporozhets@sfnu.ru; the department of computer aided design; associate professor.

Tereshenko Dmitriy Yurievich – e-mail: tereshenko@sfnu.ru; the department of computer aided design; graduate student.

УДК 519.725, 621.3.049.771.14

DOI 10.23683/2311-3103-2019-4-195-206

**А.Л. Стемповский, Д.В. Тельпухов, С.И. Гуров, Т.Д. Жукова, А.Н. Щелоков,
А.Д. Новиков**

СИНТЕЗ СФК НА ОСНОВЕ LDPC КОДА С ИСПОЛЬЗОВАНИЕМ МАЖОРИТАРНОГО ДЕКОДИРОВАНИЯ

Ионизирующее излучение приводит к возникновению кратковременных нарушений работоспособности электронной аппаратуры. Данный тип сбоя в основном рассматривался в контексте запоминающих устройств и элементов памяти. Однако, интенсивное развитие микроэлектронной промышленности приводит к росту числа сбоев в комбинационных участках, и в скором времени может привести к тому, что частота возникновения сбоев в данных участках будет сопоставима с частотой в незащищенных элементах памяти. На сегодняшний день существует множество различных методов борьбы с последствиями