

15. Pshikhopov V.Kh., Medvedev M.Yu., Gaiduk A.R., Fedorenko R.V., Krukhmalev V.A., Gurenko B.V. Position-Trajectory Control System for Unmanned Robotic Airship, *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 2014, pp. 8953-8958.
16. Pshikhopov V., Medvedev M. Multi-Loop Adaptive Control of Mobile Objects in Solving Trajectory Tracking Tasks, *Automation and Remote Control*, 2020, Vol. 81 (11), pp. 2078-2093.
17. Bogoslavskiy I., & Stachniss C. Efficient Online Segmentation for Sparse 3D Laser Scans, *ISPRS – Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, 2017, pp. 1-12. Retrieved from <https://link.springer.com/article/10.1007%2Fs41064-016-0003-y>.
18. Liu C, Xie S, Ma X, Huang Y, Sui X, Guo N, Yang F, Yang X. A Hierarchical Clustering Obstacle Detection Method Applied to RGB-D Cameras, *Electronics*, 2023, 12 (10), 2316. Available at: <https://doi.org/10.3390/electronics12102316>.
19. Kang D, Wong A, Lee B, Kim J. Real-Time Semantic Segmentation of 3D Point Cloud for Autonomous Driving, *Electronics*, 2021, 10 (16), 1960. Available at: <https://doi.org/10.3390/electronics10161960>.
20. Phalgun Chintala, Rolf Dornberger, and Thomas Hanne. Robotic Path Planning by Q Learning and a Performance Comparison with Classical Path Finding Algorithms, *International Journal of Mechanical Engineering and Robotics Research*, June 2022, Vol. 11, No. 6, pp. 373-378. DOI: 10.18178/ijmerr.11.6.373-378.

Статью рекомендовал к опубликованию д.т.н., профессор В.В. Курейчик.

Бросалин Дмитрий Олегович – АО "НКБ РИСУ"; e-mail: ddimonxt@mail.ru; г. Таганрог, Россия; тел.: +79281615949; м.н.с.

Гуренко Борис Викторович – e-mail: boris.gurenko@gmail.com; тел.: +79281687212; к.т.н.; с.н.с.

Медведев Михаил Юрьевич – НИИ робототехники и процессов управления Южного федерального университета; e-mail: medvmihal@sfedu.ru; г. Таганрог, Россия; тел.: 88634371694; д.т.н.; в.н.с.

Brosalin Dmitry Olegovich – Joint stock Company «Scientific-Design bureau of Robotics and Control Systems»; e-mail: ddimonxt@mail.ru; Taganrog, Russia; phone: +79281615949; junior researcher.

Gurenko Boris Viktorovich – e-mail: boris.gurenko@gmail.com; phone: +79281687212; cand. of eng. sc.; senior researcher.

Medvedev Mikhail Yuryevich – Research Institute of Robotics and Control Processes, Southern Federal University; e-mail: medvmihal@sfedu.ru; Taganrog, Russia; phone: +78634371694; dr. of eng. sc.; leading researcher.

УДК 519.712.2

DOI 10.18522/2311-3103-2023-6-24-34

Л.А. Гладков, Н.В. Гладкова

АЛГОРИТМ СИНТЕЗА КОМБИНАЦИОННЫХ ЛОГИЧЕСКИХ СХЕМ НА ОСНОВЕ ЭВОЛЮЦИОННОГО ПОДХОДА

Появление новых технологий изготовления компонентов цифровых электронных устройств привело к необходимости повышения эффективности методов автоматизированного проектирования. Увеличение требований к элементам вызывает рост размерности решаемых задач. Для решения задач, которые ранее невозможно было автоматизировать разрабатываются новые методы и программные приложения. Перед специалистами стоит задача разработки фундаментальных принципов построения систем проектирования нового поколения. Разработка устройств обладающих такими характеристиками как: надежность, живучесть, автоматическое устранение повреждений, является актуальной задачей. В данной работе предлагается подход к решению задачи синтеза комбинационных

схем на основе использования методов эволюционного проектирования. Под эволюционным проектированием технической системы понимается целенаправленное использование компьютерных моделей эволюции на всех стадиях разработки системы. Целью является обеспечение возможности проектирования в полностью автоматическом режиме. Основная идея самореconfigурируемых аппаратных систем состоит в том, чтобы вместо аппаратных систем общего назначения использовать системы, которые могут адаптироваться под специфику выполняемого программного обеспечения. Синтез программируемой схемы строится по принципу «восходящего» проектирования – от низшего к высшему уровню. Это позволяет конфигурировать аппаратную часть индивидуально, посредством программирования логических элементов. Для реализации данной задачи используют эволюционные алгоритмы. Логические функции могут быть описаны комбинационными схемами. Одним из достоинств комбинационных схем является их высокое быстродействие. Задача состоит в разработке структуры комбинационной логической схемы на основе заданной таблицы истинности и номенклатуры логических элементов. В работе был предложен эволюционный алгоритм синтеза комбинационных логических схем. Были разработаны методика кодирования альтернативных решений и модифицированные эволюционные операторы для синтеза новых решений. Выполнена программная реализация предложенного алгоритма. Проведенные вычислительные эксперименты подтвердили правильность выбранного подхода. Использование эволюционных методов для синтеза комбинационных логических схем позволяет повысить интеллектуальность систем проектирования.

Автоматизация проектирования; интеллектуальные САПР; самореconfigурируемые аппаратные системы; эволюционное моделирование; биоинспирированные алгоритмы; синтез комбинационных схем.

L.A. Gladkov, N.V. Gladkova

ALGORITHM FOR SYNTHESIS OF COMBINATIONAL LOGIC CIRCUITS BASED ON THE EVOLUTIONARY APPROACH

The emergence of new technologies for manufacturing components of digital electronic devices has led to the need to improve the efficiency of computer-aided design methods. Increasing requirements for elements causes an increase in the size of the problems being solved. To solve problems that were previously impossible to automate, new methods and software applications are being developed. Specialists are faced with the task of developing fundamental principles for constructing next-generation design systems. The development of devices with such characteristics as reliability, survivability, and automatic damage repair is an urgent task. This paper proposes an approach to solving the problem of synthesizing combinational circuits based on the use of evolutionary design methods. Evolutionary design of a technical system refers to the purposeful use of computer models of evolution at all stages of system development. The goal is to enable fully automatic design. The main idea of self-reconfigurable hardware systems is to replace general-purpose hardware systems with systems that can adapt to the specifics of the software being executed. The synthesis of a programmable circuit is based on the principle of “bottom-up” design – from the lowest to the highest level. This allows you to configure the hardware individually by programming logic elements. To implement this task, evolutionary algorithms are used. Logic functions can be described by combinational circuits. One of the advantages of combinational circuits is their high performance. The task is to develop the structure of a combinational logic circuit based on a given truth table and nomenclature of logical elements. The work proposed an evolutionary algorithm for the synthesis of combinational logic circuits. A technique for encoding alternative solutions and modified evolutionary operators for synthesizing new solutions were developed. A software implementation of the proposed algorithm has been completed. The computational experiments carried out confirmed the correctness of the chosen approach. The use of evolutionary methods for the synthesis of combinational logic circuits makes it possible to increase the intelligence of design systems.

Design automation; intelligent CAD systems; self-reconfiguring hardware systems; evolutionary modeling; bioinspired algorithms; combinational circuit synthesis.

Введение. В настоящее время совершается переход от традиционных САПР к новому поколению систем проектирования. Совершенствование технологий изготовления сделало возможным увеличение плотности размещения и размерности проектируемых схем, что, в свою очередь, привело к необходимости разработки новых и совершенствования существующих методов проектирования. Новые методы и программные приложения позволяют успешно решать многие задачи, которые ранее невозможно было автоматизировать. Перед специалистами стоит задача разработки фундаментальных принципов построения систем проектирования нового поколения. Одним из перспективных направлений является использование методов вычислительного интеллекта для повышения эффективности современных систем проектирования [1–3].

Одним из важных научных направлений является разработка и использование новых комбинированных, гибридных моделей и методов, позволяющих эффективно решать сложные слабоформализуемые задачи проектирования. Улучшение качества получаемых решений достигается благодаря «синергетическому эффекту», позволяющему усиливать полезные свойства компонентов разрабатываемой системы.

В данной работе предлагается подход к решению задачи синтеза комбинационных схем, основанный на использовании гибридных моделей и методов эволюционного проектирования. Под эволюционным проектированием искусственной (технической) системы понимается целенаправленное использование компьютерных моделей эволюции на всех этапах развития системы. Эволюционное проектирование – это подход, лежащий на стыке теории проектирования и теории самоорганизации [4].

Впервые генетические алгоритмы начали применять в проектировании микроэлектронных средств сравнительно недавно [5, 6]. В начале 90-х годов появился новый класс интегральных схем - программируемые логические интегральные схемы (ПЛИС, FPGA) [7, 8]. Это привело к появлению нового направления исследований в области автоматизированного проектирования микроэлектронных средств. В этот же период публикуются работы С. Луиса и Д. Раулинса [9, 10]. Их новизна состояла в том, что был предложен новый подход к разработке структуры цифровых схем. В данном подходе эволюционный алгоритм рассматривался как инструмент для решения задачи размещения цифровых логических элементов, а сама задача размещения сводилась при этом к реализации какой-либо функции. Основной идеей предложенного подхода было обеспечение возможности проектирования в полностью автоматическом режиме. Данная идея радикально отличалась от существующих технологий ориентированных на автоматизированное проектирование, т.е. проектирование с участием человека.

Направление исследований, в котором эволюционные алгоритмы используются для автоматического проектирования схем на реконфигурируемых платформах [7, 8], называется эволюционным аппаратным обеспечением [11, 12] или эволюционной электроникой [13].

Постановка задачи. Основная идея самореконфигурируемых аппаратных систем состоит в том, чтобы вместо аппаратных систем общего назначения использовать системы, которые могут адаптироваться под специфику выполняемого программного обеспечения. Для этого синтез программируемой схемы должен строиться по принципу «восходящего» проектирования – от низшего к высшему уровню, начиная от сборки самых элементарных элементов («кубиков»). Это позволит конфигурировать аппаратную часть индивидуально, посредством программирования логических элементов. Для реализации этой задачи используют эволюционные алгоритмы.

Также реконфигурируемые аппаратные средства способны динамически изменять свою архитектуру в режиме реального времени в зависимости от изменения внешних факторов. Таким образом, система может самостоятельно исправлять возможные сбои и ошибки в работе даже после начала производства. Таким образом, реконфигурируемые вычислительные системы можно определить как аппаратные платформы, архитектура которых может изменяться программно, посредством реконфигурации внутренней архитектуры системы.

Основным инструментом проектирования реконфигурируемых электронных устройств являются эволюционные алгоритмы. При реализации подобных систем эволюционные алгоритмы могут либо храниться на внешнем носителе, либо встраиваются в отдельный модуль на проектируемом кристалле.

Известно, что в процессе оптимизации решаются задачи анализа и синтеза. Также обстоит дело и при проектировании. Задача синтеза подразумевает разработку структуры комбинационной логической схемы на основе заданной таблицы истинности и номенклатуры логических элементов. Элементами комбинационных схем являются вентили, реализующие различные типы логических функций (НЕ, И, ИЛИ и т.д.). Их реализация зависит от используемой технологии. Эти три элемента (НЕ, И, ИЛИ) образуют функционально полный базис, т.е. полный набор, с помощью которого можно реализовать любую логическую функцию.

В общем случае комбинационная схема имеет множество входов $X = (x_1, x_2, \dots, x_n)$ и множество выходов $Y = (Y_1, Y_2, \dots, Y_m)$. Значение любой переменной Y_i из множества Y представляет собой функцию входных переменных, а комбинационная схема может быть описана следующей системой уравнений:

$$Y_1 = f_1(x_1, x_2, \dots, x_n);$$

$$Y_2 = f_2(x_1, x_2, \dots, x_n);$$

...

$$Y_m = f_m(x_1, x_2, \dots, x_n);$$

Следовательно, комбинационная схема может быть представлена как набор логических элементов и логических операций над ними. Число входов логических элементов должно соответствовать числу аргументов воспроизводимой булевой функции. Графическое отображение комбинационной схемы, представленной в виде условных обозначений логических элементов и связей между ними, называется функциональной схемой.

Таким образом, синтез комбинационной схемы можно выполнить на основе имеющегося набора логических элементов, таким образом, чтобы на выходе синтезированной комбинационной была реализовывать заданная булева функция. Очевидно, что в процессе синтеза могут появляться различные варианты схемы с разными наборами параметров. В этом случае проектировщик имеет возможность выбрать наиболее оптимальный, с его точки зрения, вариант.

Логические элементы и операции над ними могут быть достаточно легко представлены в виде деревьев. Представление булевой функции в виде дерева выполняется «снизу-вверх» с учетом существующего приоритета выполнения логических операций в обратном порядке, т.е. начинается с последней вычисляемой операции выражения. Например, логическая функция $\overline{((x_1 \vee x_2) \wedge \overline{x_3})} \oplus x_4$ может быть представлена в виде следующего дерева (рис. 1):

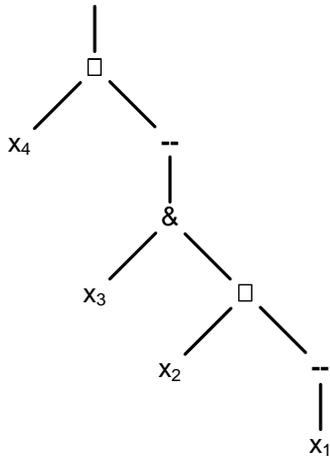


Рис. 1. Представление булевой функции в виде дерева

После замены каждой логической операции соответствующим логическим элементом мы получим функциональную схему этой логической функции (рис. 2)

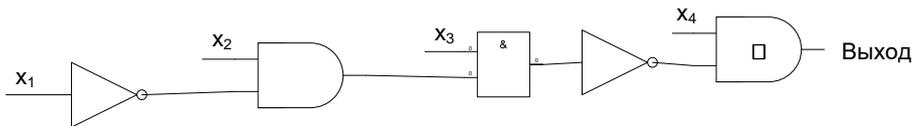


Рис. 2. Функциональная схема

Эволюционный алгоритм синтеза комбинационных схем. Предлагаемый эволюционный алгоритм синтеза комбинационных логических схем может быть описан следующим образом:

1. Ввод исходных данных.
2. Создание начальной популяции решений.
3. Расчет значений целевой функции и оценка среднего и лучшего значения по текущей популяции.
4. Проверка выполнения условия останова алгоритма.
5. Применения эволюционных операторов для синтеза новых вариантов решений и возврат к пункту 3.

Исходными данными для начала работы эволюционного алгоритма являются таблицы истинности, номенклатура логических элементов, перечень необходимых параметров, вид выходной функции, предельное число элементов в схеме и т.д.

В качестве критерия останова могут использоваться стандартные (например, число итераций алгоритма или достижения предельного значения функции пригодности) или специальные условия и значения.

Важнейшим моментом при разработке эволюционных алгоритмов является подбор методики кодирования альтернативных решений. В предлагаемом алгоритме каждая хромосома состоит из набора блоков. В свою очередь каждый блок содержит информацию об одном логическом элементе (номера его входов и тип элемента) (рис. 3).

Вход 1	...	Вход n	Тип элемента
--------	-----	--------	--------------

Рис. 3. Пример кодирования хромосомы

Возьмем для примера некий блок B_i . Блоки, расположенные в хромосоме слева от блока B_i могут являться входами для него, а сам блок B_i может являться входом для следующих блоков B_{i+1} , B_{i+2} и т.д. Информация о синтезируемой комбинационной схеме задается в виде двумерного массива, в котором перечислены используемые типы логических элементов и число входов логического элемента каждого типа. Пример подобной матрицы для комбинационной схемы, которая содержит логические элементы трех типов с двумя входами показан на рис. 4.

121	...	123	222	...	213
...
232	...	233	121	...	321

Рис. 4. Матрица логических элементов

При создании начальной популяции на основании информации о типах логических элементов и таблицах истинности случайным образом генерируются хромосомы-решения.

Пример формирования хромосом начальной популяции на основе заданной таблицы истинности (табл. 1) приведен на рис. 5.

Таблица 1

X	Y	Z	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Таким образом, мы при создании хромосомы реализуем вышеупомянутую стратегию «восходящего» проектирования комбинационных схем.

123	124	124
321	312	233
232	215	115

123	124	124	321	312	233	232	215	115
-----	-----	-----	-----	-----	-----	-----	-----	-----

Рис. 5. Пример кодирования альтернативного решения

Основными эволюционными операторами, формирующими новые варианты альтернативных решений в популяции являются операторы кроссинговера и мутации. Оператор мутации создает новые решения путем генерации случайных изменений в структуре хромосомы. Таким образом, с помощью оператора мутации мы можем получать принципиально новые варианты решений, которых не существовало в текущей популяции. Пример работы модифицированного оператора мутации показан на рис. 6. Как мы видим, оператор случайным образом выбирает один разряд хромосомы и меняет его значение.

121	...	123	122	...	213	212
121	...	233	122	...	213	212

Рис. 6. Пример работы модифицированного оператора мутации

Оператор кроссинговера также используется для создания новых решений. При выполнении операции скрещивания новые решения получаются за счет обмена частями двух или нескольких хромосом. В результате новое решение наследует определенные свойства и особенности своих «родителей». Таким образом, в популяции сохраняется «потомственность» (связь между различными поколениями). Пример работы модифицированного оператора кроссинговера показан на рис. 7.

124	...	123	122	234	...	212
121	...	214	323	125	...	123
...
124	...	123	323	234	...	212
121	...	214	122	125	...	123

Рис. 7. Пример работы модифицированного оператора мутации

При решении задачи синтеза комбинационных логических схем значение функции пригодности может соответствовать числу правильно выполняемых логических операций в строках таблицы истинности моделируемой логической функции.

Результаты вычислительных экспериментов. Была выполнена программная реализация предложенного эволюционного алгоритма синтеза комбинационных схем на языке C#. Результатом работы алгоритма является решение, соответствующее варианту с наибольшим значением функции пригодности. Для полученного на выходе варианта также возможно получение дополнительной информации, которая может потребоваться проектировщику (время работы алгоритма, число элементов, описание закодированного решения и т.д.).

Были проведены серии вычислительных экспериментов с целью оценки качества получаемых решений, временных характеристик разработанного алгоритма и оптимальных значений управляющих параметров алгоритма. Одним из важных параметров эволюционного алгоритма является размер популяции. Размер популяции непосредственно влияет на скорость нахождения оптимального решения, время работы алгоритма, а также разнообразие вариантов в текущей популяции. Были проведены серии экспериментов для изучения влияния размера популяции на время работы. В качестве критерия останова использовалось число итераций алгоритма, значения вероятности выполнения эволюционных операторов при этом не изменялись. Полученные результаты отображены на рис. 8.

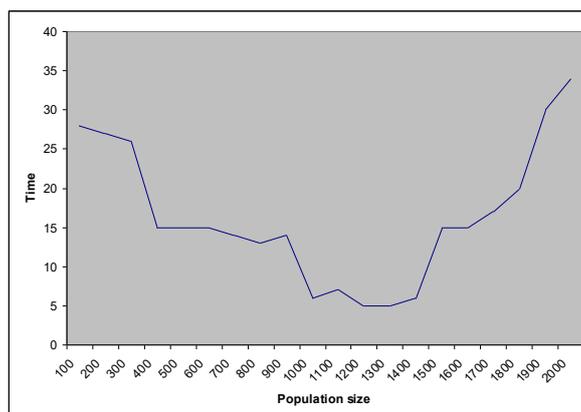


Рис. 8. Зависимость времени работы алгоритма от размера популяции

Анализ полученных данных позволяет сделать вывод о том, что оптимальным является размер популяции 1000-1400 хромосом, при котором обеспечивается наибольшее разнообразие вариантов решений. Увеличение или сокращение популяции дает отрицательный эффект.

Также в ходе вычислительных экспериментов анализировалось влияние вероятности выполнения эволюционных операторов на время работы алгоритма (рис. 9). При этом размер текущей популяции и число итераций алгоритма оставались постоянными. Анализ полученных данных показывает, что зависимость времени работы алгоритма от значения вероятности мутации является немонотонной функцией. Наибольший положительный эффект достигается при задании вероятности выполнения операции мутации в диапазоне 0,45-0,75. Увеличение значения вероятности ведет к росту тупиковых вариантов решений, а при снижении значения вероятности увеличивается время работы.

В ходе экспериментов проводился анализ возможностей синтеза комбинационных логических схем с использованием данных из библиотеки промышленных бенчмарков MCNC [14]. Целью эксперимента является анализ возможности синтеза схем для промышленных таблиц истинности. Полученные результаты показали, что предложенный алгоритм позволяет решать задачу синтеза промышленных комбинационных логических схем малых размеров, однако качество получаемых решений повышается при применении декомпозиции.

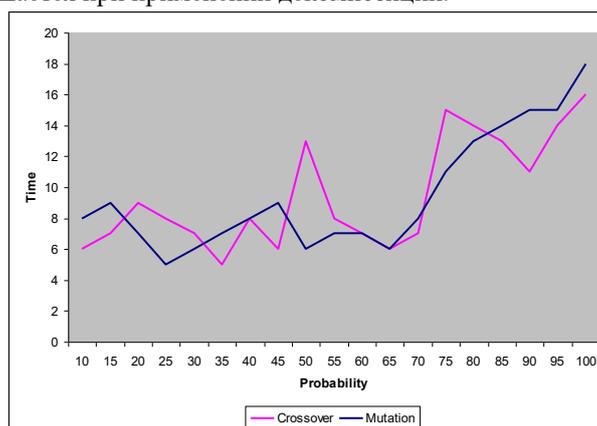


Рис. 9. Зависимость времени работы алгоритма от вероятности выполнения генетических операторов

Также исследовалось влияние вероятности выполнения оператора кроссинговера. Анализ данных экспериментальных данных показывает, что лучшие результаты алгоритм показывает при значениях вероятности кроссинговера в интервале 40–80%.

Заключение. В данной статье предложен подход к решению задачи автоматического синтеза комбинационных схем на основе эволюционных алгоритмов. Проведенные вычислительные эксперименты показали, что предлагаемый подход имеет определенный потенциал и возможности для дальнейшего развития. Очевидно, что данный подход вполне применим к решению задач синтеза логических схем средней размерности (порядка 250-500 вентилей).

На основе анализа комбинационных логических схем, полученных в процессе вычислительных экспериментов с использованием разработанного программно-алгоритмического комплекса, можно сделать вывод, что применение предложенного алгоритма позволяет повысить качество синтезируемых комбинационных

схем в среднем на 10–15%. Новые возможности синтеза комбинационных схем различной топологии позволяют применять предлагаемый подход для работы с новыми проектными нормами и технологиями. Также использование разработанного программного обеспечения будет полезно в процессе принятия решения при анализе и выборе оптимального варианта из сформированных в ходе решения альтернативных вариантов комбинационных логических схем.

Авторами накоплен значительный опыт разработки и применения гибридных моделей и методов для решения различных задач оптимизации и проектирования [15–18], что позволяет рассчитывать на повышение качества работы алгоритма синтеза комбинационных схем.

В дальнейшем предполагается разработать гибридный нечетко-генетический алгоритм, что позволит в процессе работы алгоритма регулировать значения вероятностей выполнения генетических операторов и других управляющих параметров [19, 20].

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Alpert Charles J., Mehta Dinesh P., Sapatnekar Sachin S.* Handbook of algorithms for physical design automation. – CRC Press, New York, USA, 2009.
2. *Cohon J.P., Karro J., Lienig J.* Evolutionary Algorithms for the Physical Design of VLSI Circuits. Advances in Evolutionary Computing: Theory and Applications / Ghosh A., Tsutsui S. (eds.). – Springer Verlag, London, 2003.
3. *Baqais A.A.B.* A Multi-view Comparison of Various Metaheuristic and Soft Computing Algorithms // I.J. Mathematical Sciences and Computing. – 2017. – 1.3 (4). – P. 8-19.
4. *Тарасов В.Б., Голубин А.В.* Эволюционное проектирование: на границе между проектированием и самоорганизацией // Известия ТРТУ.. – 2006. – № 8 (63). – С. 77-82.
5. *Sushil J. Louis and Gregory J. Rawlins.* Using genetic algorithms to design structures. Technical Report 326, Computer Science Department, Indiana University. – Bloomington, Indiana, 1991.
6. *Koza J.R., Bennett F.H., Andre D., Keane M.A.* Automated WYWIWYG design of both the topology and component values of electrical circuits using genetic programming // Proceedings of the First Annual Conference on Genetic Programming, Cambridge, Massachusetts, Stanford University. – The MIT Press, 1996. – P. 123-131.
7. *Blondet B., Roxby P.J., Keller E., McMillan S., Sundararajan P.* A Self-reconfiguring Platform // Field-Programmable Logic and Applications. 13th International Conference, FPL 2003 Proceedings, pp. 565-574.
8. *Eldredge J.G., Hutchings B.L.* Run-Time Reconfiguration: A Method for Enhancing the Functional Density of SRAM-Based FPGAs // in Journal of VLSI Signal Processing. – 1996. – Vol. 12. – P. 67-86.
9. *Louis S.J., Rawlins G.J.E.* Syntactic Analysis of Convergence in Genetic Algorithms. Foundations of Genetic Algorithms / 2 ed. by L.D. Whitley. – San Mateo, CA: Morgan Kaufmann, 1993. – 141 p.
10. *Louis S.J., Rawlins J.E.* Designer genetic algorithms: genetic algorithms in structure design // ICGA-91 // in Proceedings of the Fourth International Conference on Genetic Algorithms / Belew, K.K. and Booker, L.B., eds. – Booker, Morgan Kaufman, San Manteo, CA, 1991.
11. *Higuchi T. et al.* Evolvable hardware and its applications to pattern recognition and fault-tolerant systems // in Towards Evolvable Hardware: An International Workshop, Lausanne, Swiss, 1995, 2. Chapter of book: Towards Evolvable Hardware: The evolutionary engineering approach / Sanchez E., and Tomassini M., eds. Vol. 1062. – LNCS, Springer-Verlag, 1996. – 118 p.
12. *Hemmi H., Mizoguchi J., Shimonara K.* Development and evolution of hardware behaviours // in Towards Evolvable Hardware: An International Workshop, Lausanne, Swiss, 1995, 2. Chapter of book: Towards Evolvable Hardware: The evolutionary engineering approach / Sanchez E., and Tomassini M. eds. Vol. 1062. – LNCS, Springer-Verlag, 1996.
13. *Zebulum R.S., Pacheco M.A., Vellasco M.M.* Evolutionary Electronics: Automatic Design of Electronic Circuits and Systems by Genetic Algorithms. – USA, CRC Press LLC, 2002.

14. Yang S. Logic synthesis and optimization benchmark user guide version 3.0, MCNC, 1991.
15. Gladkov L., Kureychik V.I., Kureychik V., Sorokoletov P. Bio-inspired methods in optimization. – Moscow: Fizmatlit, 2009.
16. Gladkov L.A., Gladkova N.V., Gusev N.Y., Semushina N.S. Integrated approach to the solution of computer-aided design problems // Proceedings of the 4th International Scientific Conference “Intelligent Information Technologies for Industry” (IITI’19). Advances in Intelligent Systems and Computing. – Springer, Cham, 2020. – Vol. 1156. – P. 465-476.
17. Gladkov L.A., Gladkova N.V., Leiba S.N., Strakhov N.E. Development and research of the hybrid approach to the solution of optimization design problems // Advances in Intelligent Systems and Computing. Vol. 875. International Conference on Intelligent Information Technologies for Industry IITI’18. – Springer Nature Switzerland AG, 2019. – Vol. 2. – P. 246-257.
18. Gladkov L.A., Veselov G.E., Gladkova N.V. Development and research of algorithms for the synthesis of combinational logic circuits based on the evolutionary approach // Lecture Notes in Networks and Systems. Vol. 776 “Proceedings of the 7th International Scientific Conference “Intelligent Information Technologies for Industry” (IITI’23)”. – Springer Nature Switzerland AG, 2023. – Vol. 1. – P. 210-221.
19. Batyrshin I. etc. Fuzzy hybrid systems. Theory and practice. – Moscow: Fizmatlit, 2007.
20. Borisov V., Kruglov V., Fedulov A. Nechetkie modeli i seti (Fuzzy models and networks). – Moscow: Goryachaya liniya – Telekom, 2007.

REFERENCES

1. Alpert Charles J., Mehta Dinesh P., Sapatnekar Sachin S. Handbook of algorithms for physical design automation. CRC Press, New York, USA, 2009.
2. Cohoon J.P., Karro J., Lienig J. Evolutionary Algorithms for the Physical Design of VLSI Circuits. Advances in Evolutionary Computing: Theory and Applications, Ghosh A., Tsutsui S. (eds.). Springer Verlag, London, 2003.
3. Baqais A.A.B. A Multi-view Comparison of Various Metaheuristic and Soft Computing Algorithms, *I.J. Mathematical Sciences and Computing*, 2017, 1.3 (4), pp. 8-19.
4. Tarasov V.B., Golubin A.V. Evolyutsionnoe proektirovanie: na granitse mezhdru proektirovaniem i samoorganizatsiey [Evolutionary design: on the border between design and self-organization], *Izvestiya TRTU [Izvestiya TSURE]*, 2006, No. 8 (63), pp. 77-82.
5. Sushil J. Louis and Gregory J. Rawlins. Using genetic algorithms to design structures. Technical Report 326, Computer Science Department, Indiana University. Bloomington, Indiana, 1991.
6. Koza J.R., Bennett F.H., Andre D., Keane M.A. Automated WYWIWYG design of both the topology and component values of electrical circuits using genetic programming, *Proceedings of the First Annual Conference on Genetic Programming, Cambridge, Massachusetts, Stanford University*. The MIT Press, 1996, pp. 123-131.
7. Blondet B., Roxby P.J., Keller E., McMillan S., Sundararajan P. A Self-reconfiguring Platform // Field-Programmable Logic and Applications. 13th International Conference, FPL 2003 Proceedings, pp. 565-574.
8. Eldredge J.G., Hutchings B.L. Run-Time Reconfiguration: A Method for Enhancing the Functional Density of SRAM-Based FPGAs, in *Journal of VLSI Signal Processing*, 1996, Vol. 12, pp. 67-86.
9. Louis S.J., Rawlins G.J.E. Syntactic Analysis of Convergence in Genetic Algorithms. Foundations of Genetic Algorithms, 2 ed. by L.D. Whitley. San Mateo, CA: Morgan Kaufmann, 1993, 141 p.
10. Louis S.J., Rawlins J.E. Designer genetic algorithms: genetic algorithms in structure design // ICGA-91 // in Proceedings of the Fourth International Conference on Genetic Algorithms / Belew, K.K. and Booker, L.B., eds. Booker, Morgan Kaufman, San Manteo, CA, 1991.
11. Higuchi T. et al. Evolvable hardware and its applications to pattern recognition and fault-tolerant systems, in *Towards Evolvable Hardware: An International Workshop, Lausanne, Swiss, 1995*, 2. Chapter of book: *Towards Evolvable Hardware: The evolutionary engineering approach*, Sanchez E., and Tomassini M., eds. Vol. 1062. LNCS, Springer-Verlag, 1996, 118 p.
12. Hemmi H., Mizoguchi J., Shimonara K. Development and evolution of hardware behaviours, in *Towards Evolvable Hardware: An International Workshop, Lausanne, Swiss, 1995*, 2. Chapter of book: *Towards Evolvable Hardware: The evolutionary engineering approach*, Sanchez E., and Tomassini M. eds. Vol. 1062. LNCS, Springer-Verlag, 1996.

13. Zebulum R.S., Pacheco M.A., Vellasco M.M. Evolutionary Electronics: Automatic Design of Electronic Circuits and Systems by Genetic Algorithms. USA, CRC Press LLC, 2002.
14. Yang S. Logic synthesis and optimization benchmark user guide version 3.0, MCNC, 1991.
15. Gladkov L., Kureychik V.I., Kureychik V., Sorokoletov P. Bio-inspired methods in optimization. Moscow: Fizmatlit, 2009.
16. Gladkov L.A., Gladkova N.V., Gusev N.Y., Semushina N.S. Integrated approach to the solution of computer-aided design problems, *Proceedings of the 4th International Scientific Conference "Intelligent Information Technologies for Industry" (IITI'19). Advances in Intelligent Systems and Computing*. Springer, Cham, 2020, Vol. 1156, pp. 465-476.
17. Gladkov L.A., Gladkova N.V., Leiba S.N., Strakhov N.E. Development and research of the hybrid approach to the solution of optimization design problems, *Advances in Intelligent Systems and Computing*. Vol. 875. *International Conference on Intelligent Information Technologies for Industry IITI'18*. Springer Nature Switzerland AG, 2019, Vol. 2, pp. 246-257.
18. Gladkov L.A., Veselov G.E., Gladkova N.V. Development and research of algorithms for the synthesis of combinational logic circuits based on the evolutionary approach, *Lecture Notes in Networks and Systems*. Vol. 776 "Proceedings of the 7th International Scientific Conference "Intelligent Information Technologies for Industry" (IITI'23)". Springer Nature Switzerland AG, 2023, Vol. 1, pp. 210-221.
19. Batyrshin I. etc. Fuzzy hybrid systems. Theory and practice. Moscow: Fizmatlit, 2007.
20. Borisov V., Kruglov V., Fedulov A. Nechetkie modeli i seti (Fuzzy models and networks). Moscow: Goryachaya liniya – Telekom, 2007.

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

Гладков Леонид Анатольевич – Южный федеральный университет; e-mail: lagladkov@sfedu.ru; г. Таганрог, Россия; кафедра САПР; профессор.

Гладкова Надежда Викторовна – e-mail: nvgladkova@sfedu.ru; кафедра САПР; старший преподаватель.

Gladkov Leonid Anatol'evich – Southern Federal University; e-mail: lagladkov@sfedu.ru; Taganrog, Russia; the department of CAD; professor.

Gladkova Nadezhda Viktorovna – e-mail: nvgladkova@sfedu.ru; the department of CAD; senior teacher.

УДК 621.396

DOI 10.18522/2311-3103-2023-6-34-44

Д.А. Гужва, К.О. Север, И.И. Турулин

8-ПОЛОСНЫЙ НЕОДНОРОДНЫЙ БАНК БЫСТРОДЕЙСТВУЮЩИХ РЕКУРСИВНЫХ ФИЛЬТРОВ С КОНЕЧНОЙ ИМПУЛЬСНОЙ ХАРАКТЕРИСТИКОЙ ДЛЯ СЛУХОВЫХ АППАРАТОВ

Рассмотрены рекурсивные фильтры с конечной импульсной характеристикой и банки фильтров. Банк фильтров – это массив полосовых фильтров. Блок фильтров анализа разделяет входной сигнал на несколько компонентов, причем каждый из подфильтров несет один частотный поддиапазон исходного сигнала. Напротив, блок фильтров синтеза объединяет выходные данные поддиапазонов для восстановления исходного входного сигнала. В большинстве приложений определенные частоты являются более важными, чем другие. Блоки фильтров могут изолировать различные частотные компоненты в сигнале. Таким образом, мы можем приложить больше усилий для обработки более важных компонентов и меньше усилий для обработки менее важных компонентов. Фильтры поддиапазонов могут комбинироваться с понижающей или повышающей дискретизацией для формирования банка многоскоростных фильтров. Банки фильтров широко используются для распознавания речи и улучшения качества речи. В настоящее время банки фильтров расширили свое применение до обработки видео и изображений. Кроме того, блоки фильтров