

Гаврилов Сергей Витальевич – Институт проблем проектирования в микроэлектронике РАН (ИППМ РАН); e-mail: sergey_g@ippm.ru; 124365, Москва, Зеленоград, ул. Советская, 3; тел.: +74997299890; отдел автоматизации проектирования цифровых схем; г.н.с.; д.т.н.; профессор.

Рыжова Дарья Игоревна – e-mail: ryzhova_d@ippm.ru; отдел автоматизации проектирования цифровых схем; н.с.; к.т.н.

Щелоков Альберт Николаевич – e-mail: schan@ippm.ru; с.н.с.; к.ф.-м.н.; доцент.

Gavrilov Sergey Vitalievich – Institute for design problems in microelectronics of Russian Academy of Science (IPPM RAS); e-mail: sergey_g@ippm.ru; 124365, Moscow, Zelenograd, Sovetskaya Street, 3; phone: +74997299890; the department of digital circuits design automation, senior research scientist; dr. of eng. sc.; professor.

Ryzhova Daria Igorevna – e-mail: ryzhova_d@ippm.ru; the department of digital circuits design automation; research scientist; cand. of eng. sc.;

Schelokov Albert Nikolaevich – e-mail: schan@ippm.ru; senior researcher; cand. of phys. and math. sc.; associate professor.

УДК 519.712.2

DOI 10.23683/2311-3103-2018-4-14-24

Л.А. Гладков, Н.В. Гладкова, С.Н. Лейба**ГИБРИДНАЯ МОДЕЛЬ РЕШЕНИЯ ОПТИМИЗАЦИОННЫХ ЗАДАЧ
ПРОЕКТИРОВАНИЯ***

Рассматриваются методы оптимизационных задач автоматизированного проектирования схем цифровой электронно-вычислительной аппаратуры. Отмечена актуальность и важность разработка новых эффективных методов решения подобных задач. Приведена постановка задачи размещения элементов схем электронной аппаратуры, выбраны ограничительные области допустимых решений. Предложена методика определения качества получаемого решения на основе комплексной нормированной оценки сумм штрафов по нескольким выбранным критериям. Предложена новая гибридная модель рассматриваемой задачи на основе сочетания эволюционных методов поиска, математического аппарата нечеткой логики и возможностей параллельной организации вычислительного процесса. Разработаны новые модификации основных генетических операторов. Предложен модифицированный оператор миграции для обмена информацией между популяциями решений в процессе выполнения параллельных вычислений. Разработана структура параллельного гибридного алгоритма. Для обмена решениями между популяциями предложено использовать островную и буферную модели параллельного генетического алгоритма. Предложена реализация модуля нечеткого управления на основе использования многослойной нейронной сети и функции Гаусса. Для повышения качества получаемых результатов в контур эволюции экспертной информации включен нечеткий логический контроллер, регулирующий значения параметров процесса эволюции. Сформулированы основные принципы работы блока нечеткого управления. Представлена структурная схема, разработанного гибридного алгоритма. Подробно рассмотрены особенности программной реализации предложенного гибридного алгоритма. Описана структура интерфейса, представлены основные элементы графического интерфейса разработанного приложения. Представлено краткое описание проведенных вычислительных экспериментов, подтверждающих эффективность предложенного метода.

Автоматизация проектирования; генетический алгоритм; эволюционные вычисления; нечеткое управление; параллельные вычисления; нечеткий логический контроллер.

* Работа выполнена при поддержке РФФИ (проект № 17-01-00627).

L.A. Gladkov, N.V. Gladkova, S.N. Leiba

HYBRID MODEL OF SOLVING OPTIMIZATION DESIGN PROBLEMS

The methods of optimization tasks of computer-aided design in circuits of digital electronic computing equipment are considered herein. The relevance and importance of developing new effective methods for solving similar problems are noted. The formulation of the problem of placing elements in circuits of electronic equipment is given, restrictions of the domain of admissible solutions are chosen. A technique for determining the quality of the solution obtained is proposed on the basis of a complex normalized assessment of the amounts of fines for several selected criteria. A new hybrid model of the problem is proposed on the basis of a combination of evolutionary search methods, a mathematical apparatus of fuzzy logic and the possibilities of parallel organization of the computational process. New modifications of the main genetic operators have been developed. A modified migration operator is proposed to exchange information between solution populations in the process of performing parallel computations. The structure of the parallel hybrid algorithm is developed. To exchange solutions between populations, it was proposed to use the island and buffer models of the parallel genetic algorithm. The implementation of the fuzzy control module based on the use of a multilayer neural network and the Gaussian function is proposed. To improve the quality of the results obtained, a fuzzy logic controller that regulates the values of the parameters of the evolution process is included in the evolution of expert information. The basic principles of the fuzzy control unit are formulated. The structural scheme of the developed hybrid algorithm is presented. Features of software implementation of the proposed hybrid algorithm are considered in detail. The structure of the interface is described, the main elements of the graphic interface of the developed application are presented. A brief description of the computational experiments that confirm the effectiveness of the proposed method is presented.

Design automation; genetic algorithm; evolutionary computation; fuzzy control; parallel computing; fuzzy logic controller.

Введение. Большинство оптимизационных задач, решаемых на конструкторском этапе автоматизированного проектирования электронно-вычислительной аппаратуры требуют огромных затрат времени и вычислительных ресурсов. Это обусловлено необходимостью перебора огромного числа различных вариантов решений. При этом по своей вычислительной сложности задачи проектирования относятся к классу, так называемых, NP-полных задач, т.е. задач, для которых не существует детерминированного полиномиального алгоритма.

Следовательно, для гарантированного нахождения наилучшего решения (глобального экстремума) в таких задачах требуется выполнить полный перебор, что не представляется возможным их большой размерности [1–3]. Поэтому для решения таких задач разрабатываются метаэвристические алгоритмы, позволяющие находить близкие к оптимальным (квазиоптимальные) решения.

Одним из подходов, позволяющих успешно решать задачу повышения эффективности и качества решения сложных оптимизационных задач большой размерности является интеграция различных научных методов характерных для таких направлений вычислительного интеллекта как биоинспирированные алгоритмы, нечеткие вычисления, искусственные нейронные сети [4–6].

В данной статье предлагается гибридный подход к решению рассматриваемых оптимизационных задач на основе интеграции различных подходов, таких как, эволюционные алгоритмы поиска, искусственные нейронные сети, нечеткие модели управления параметрами алгоритма и механизмы распараллеливания вычислений для повышения эффективности организации процесса поиска [7–14].

Постановка задачи. К задачам конструкторского этапа проектирования ЭВА относятся задачи компоновки, размещения, трассировки, планирования, верификации. Наиболее сложными и ответственными с точки зрения качества будущих изделий являются задачи размещения и трассировки. Эти задачи тесно взаимосвязаны друг с другом, поскольку результат решения задачи размещения элементов является исходной информацией для задачи трассировки, а качество решения за-

дачи размещения непосредственно влияет на сложность и качество задачи трассировки. Поэтому представляет практический интерес разработка интегрированных методов позволяющих решать эти задач в едином цикле с учетом взаимных ограничений и текущих результатов [15].

Поскольку основной целью задачи размещения является определение местоположения элементов схемы в проекте, сначала необходимо определить область размещения P . Обычно область размещения представляет собой область прямоугольника, определяемую координатами $(x_{low}, y_{low}, x_{high}, y_{high})$. Наличие прямоугольной области не является необходимым условием в современных задачах размещения, поскольку в последнее время в некоторых задачах используются области размещения, имеющие L -образную или T -образную форму. Однако для задачи глобального размещения нормой по-прежнему является прямоугольная область размещения. Список соединений схемы представлен в виде графа $G = (V, E)$, где V – это множество элементов схемы (вершин), а E – множество соединений между ними (ребер). Множество Вершин V состоит из двух непересекающихся подмножеств, MV – множество передвигаемых элементов и FV – множество фиксированных элементов. Для каждой вершины $v \in FV$ местоположение уже определено и в процессе решения задачи размещения не изменяется. Расположение каждой вершины $v \in MV$ определяется в процессе решения, и оно должно находиться в пределах заданной области размещения P [2].

Цель размещения – определить доступное местоположение для каждого подвижного элемента схемы при выполнении условия минимизации заданной целевой функции. Пусть задано множество элементов E :

$$E = \{e_i \mid i = 1, \dots, N\},$$

где e_i – размещаемый элемент, N – количество размещаемых элементов.

$$e_i = (l_i, h_i, T_i),$$

где l_i – длина элемента, h_i – высота элемента, T_i – список контактов элемента размещения.

$$T_i = \{t_j \mid j = 1, \dots, K\},$$

где t_j – контакт, K – количество контактов элемента.

$$t_j = (x_j, y_j),$$

где x_j, y_j – координаты контакта относительно базовой точки элемента.

Необходимо найти вариант размещения элементов на монтажном пространстве

$$V = \{(x_i, y_i) \mid i = 1, \dots, N\},$$

где (x_i, y_i) – координаты верхнего левого угла установочной площади элемента размещения i , такие, чтобы суммарная площадь перекрытия размещенных элементов была равна нулю, а сумма значений остальных критериев минимальной.

Для каждой цепи необходимо найти список позиций коммутационного поля, через которые она проходит:

$$W_h = \{(x_q, y_q) \mid i = 1, \dots, Q\},$$

где Q – количество позиций, через которые проходит h -я цепь.

Разработка алгоритма. При кодировании решений множество позиций представляется в виде регулярной структуры (решетки). Каждая позиция p_i имеет координаты x_i, y_i . Позиции нумеруются по возрастанию координаты x_i в пределах строки слева направо, а строки в свою очередь упорядочиваются по возрастанию координаты y_i сверху вниз.

Каждый элемент имеет базовую точку O_i^δ и базовые оси координат $O_i^\delta X_i^\delta, O_i^\delta Y_i^\delta$, относительно которых задается контурное описание элемента e_i . Базовой точкой является нижний левый угол элемента. Контурное описание имеет прямоугольную форму. Будем считать, что элемент e_i назначен в позицию p_j , если его базовая точка O_i^δ совмещена с точкой коммутационного поля, имеющей координаты x_j, y_j . Точки пересечения пунктирных линий соответствуют посадочным позициям на коммутационном поле.

Каждое решение представляется в виде хромосомы H_i . Порядковый номер гена в хромосоме соответствует порядковому номеру размещаемого элемента. Значение гена соответствует номеру позиции на коммутационном поле. Число генов в хромосоме равно количеству размещаемых элементов.

При расчёте значения ЦФ размещения вычисляется нормированная оценка суммы штрафа за перекрытие площадей размещаемых элементов, оценки длин межсоединений, показателя трассируемости, а также оценки тепловой и электромагнитной совместимости элементов (Рис. 1).

$$F_p = k_1 * S + k_2 * L + k_3 * T + k_4 * J + k_5 * Q,$$

где S – суммарная площадь перекрытия элементов, L – оценка длин межсоединений, T – показатель трассируемости, J – суммарное электромагнитное воздействие элементов друг на друга, Q – суммарное тепловое воздействие элементов друг на друга, k_1, k_2, k_3, k_4, k_5 – коэффициенты, определяющие влияние каждой составляющей функции принадлежности на общую оценку.

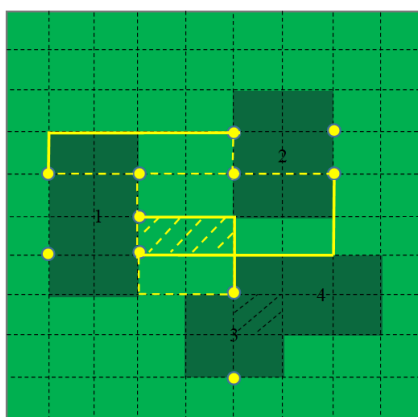


Рис. 1. Пример определения качества размещения

Площадь пересечения двух элементов вычисляется следующим образом:

$$x_{11} = \max(\min(x_1, x_2 + d), \min(x_3, x_4 + d));$$

$$x_{12} = \min(\max(x_1, x_2 + d), \max(x_3, x_4 + d));$$

$$y_{11} = \max(\min(y_1, y_2 + d), \min(y_3, y_4 + d));$$

$$y_{12} = \min(\max(y_1, y_2 + d), \max(y_3, y_4 + d));$$

ЕСЛИ $((x_{12} - x_{11} \geq 0) \text{ И } (y_{12} - y_{11} \geq 0))$

ТО $R_{ij} = (x_{12} - x_{11}) * (y_{12} - y_{11})$ **ИНАЧЕ** $R_{ij} = 0$,

где x_1, y_1 – координаты верхнего левого угла первого прямоугольника,
 x_2, y_2 – координаты нижнего правого угла первого прямоугольника,
 x_3, y_3 – координаты верхнего левого угла второго прямоугольника,
 x_4, y_4 – координаты нижнего правого угла второго прямоугольника,
 d – минимально допустимое расстояние между элементами.

Размер штрафа за перекрытие площадей размещаемых элементов зависит от суммарной площади пересечения всех элементов. При этом учитывается минимально допустимое расстояние между элементами.

Для оценки длин межсоединений целесообразно использовать полупериметр описывающего прямоугольника цепи.

При решении задачи трассировки показателем качества решения является доля (процент) непротрассированных соединений. Дополнительным критерием качества трассировки является суммарная площадь пересечения областей описывающих прямоугольников всех цепей (рис. 2).

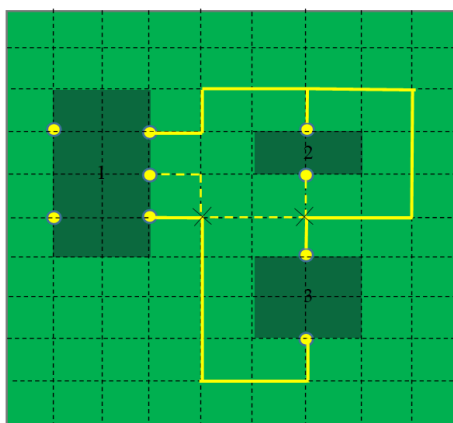


Рис. 2. Пример определения качества трассировки

Для совместного решения задач размещения и трассировки используется параллельный многопопуляционный генетический алгоритм [12-14]. Он предполагает параллельное выполнение эволюционных процессов на нескольких популяциях. Для обмена особями используются островная и буферная модели параллельного генетического алгоритма.

В островной модели синхронизация асинхронных процессов осуществляется в точках миграции. Оператор миграции применяется для обмена особями между популяциями. Отбор особей для миграции выполняется из некоторого количества хромосом популяции, имеющих наилучшее значение ЦФ размещения при этом учитывается оценка количества непротрассированных соединений.

В буферной модели обмен особями между популяциями осуществляется через общий промежуточный буфер хромосом. Обмен выполняется в точках миграции. К популяции, которая достигла точки миграции, применяется оператор миграции.

Для каждого варианта размещения, описываемого хромосомой, выполняется трассировка с помощью волнового алгоритма. Затем, из одной популяции в другую копируется некоторое количество хромосом с наилучшим значением данного показателя. При этом из популяций удаляется такое же количество хромосом с наихудшим значением показателя. На рис. 3 представлена схема модели параллельного генетического алгоритма, выполняемого на 2-х популяциях. На практике количество популяций может быть значительно больше.

Для трассировки используется волновой алгоритм. Работа алгоритма включает в себя три этапа: инициализацию, распространение волны и восстановление пути. Во время инициализации строится образ множества ячеек печатной платы, каждой ячейке приписываются атрибуты проходимости/непроходимости, запоминаются стартовая и финишная ячейки.

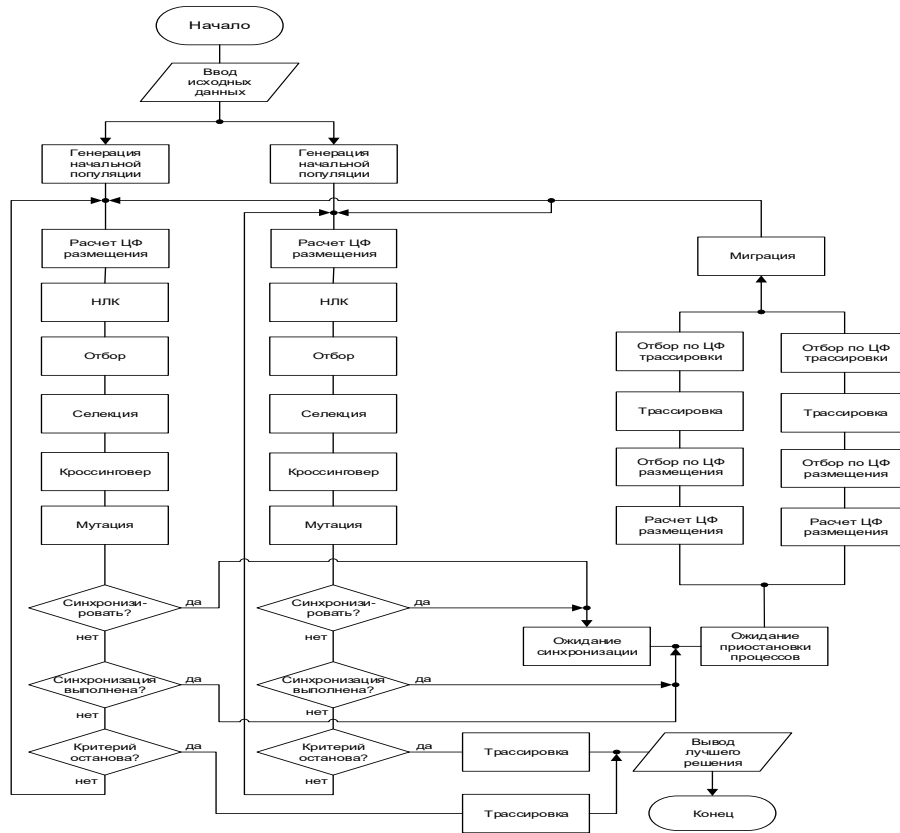


Рис. 3. Структурная схема островной модели алгоритма размещения

Далее, от стартовой ячейки порождается шаг в соседнюю ячейку, при этом проверяется, проходима ли она, и не принадлежит ли ранее меченной в пути ячейке. Для классификации соседних ячеек используется окрестность фон Неймана. Соседними ячейками считаются 4 ячейки по вертикали и горизонтали. При выполнении условий проходимости и непринадлежности её к ранее помеченным в пути ячейкам, в атрибут ячейки записывается число, равное количеству шагов от стартовой ячейки, на первом шаге это будет 1. Каждая ячейка, меченная числом шагов от стартовой ячейки, становится стартовой и из неё порождаются очередные шаги в соседние ячейки. Очевидно, что при таком переборе будет найден путь от начальной ячейки к конечной, либо очередной шаг из любой порождённой в пути ячейки будет невозможен.

Для оценки эффективности эволюции используются параметры, характеризующие динамику изменений среднего значения целевой функции популяции [16–20].

Для расчёта выходных значений используется блок нечёткого управления, представленный в пункте 3. В качестве входных переменных x_i будем использовать параметры e_i . Полученные значения \bar{u} будут эквивалентны параметрам ΔP_s , ΔP_m , $P_s(t)$. Для вычисления каждого из данных параметров будет использоваться отдельный модуль нечёткого управления.

Завершающий этап в процессе проектирования модуля нечеткого управления – это определение формы представления нечетких множеств A_i^k , $1, \dots, n$; $k = 1, \dots, N$. После объединения всех элементов функция для модуля нечеткого управления приобретает окончательный вид:

$$\bar{y} = \frac{\sum_{k=1}^N \bar{y}^k \left(\prod_{i=1}^n \exp \left(- \left(\frac{\bar{x}_i - \bar{x}_i^k}{\sigma_i^k} \right)^2 \right) \right)}{\sum_{k=1}^N \left(\prod_{i=1}^n \exp \left(- \left(\frac{\bar{x}_i - \bar{x}_i^k}{\sigma_i^k} \right)^2 \right) \right)}.$$

Каждый элемент этой формулы можно задать в форме функционального блока (сумма, произведение, функция Гаусса), что после соответствующего объединения позволяет создать многослойную нейронную сеть. В нашем случае нейронная сеть будет содержать 4 слоя. Каждый элемент первого слоя реализует функцию принадлежности нечеткого множества A_i^k , $1, \dots, n$; $k = 1, \dots, N$. В этот слой поступают входные сигналы \bar{x}_i , а на его выходе формируются значения функции принадлежности для этих сигналов. На выходе первого слоя формируются значения функции принадлежности нечетким множествам. Конфигурация связей слоя 2 соответствует базе правил, а мультипликаторы - блоку вывода. Применение мультипликаторов в качестве узлов слоя 2 обусловлено тем, что в нечетких операциях используется операция умножения. Количество элементов этого слоя равно количеству правил хранящихся в базе. Слои 3 и 4 реализуют функции блока дефазификации.

Очевидно, что описанная структура представляет собой многослойную сеть, основанную на идее нечеткого вывода. В отличие от «чистых» нейронных сетей, каждый слой в целом и отдельные составляющие его элементы, также как и конфигурация связей, все параметры и веса имеют физическую интерпретацию. Это свойство оказывается необычайно важным, поскольку знания не распределяются по сети и могут быть легко локализованы и при необходимости откорректированы экспертом.

Результаты экспериментов. Приложение состоит из следующих модулей: алгоритмы размещения и трассировки, модуль нечеткого управления, синтаксический анализатор и графический интерфейс. Алгоритмы размещения и трассировки реализованы в виде обобщенных алгоритмов, которые могут обрабатывать различные входные данные. Алгоритм совместного решения задач размещения и трассировки реализован на основе обобщенного алгоритма размещения, для которого определен метод расчёта целевой функции размещения. На вход алгоритма подаются данные о топологии печатной платы. Алгоритм трассировки является вспомогательным и используется для оценки получаемых вариантов размещения. Модуль нечеткого управления используется для динамической настройки генетического алгоритма. База правил модуля нечеткого управления считывается из файла. Синтаксический анализатор используется для считывания данных о топологии печатной платы из файла, а также для записи данных в файл. Разработанная архитектура позволяет дополнять систему новыми свойствами и моделями поведения.

Для анализа эффективности разрабатываемых алгоритмов используются графики изменения среднего и минимального значения целевой функции размещения. На каждой итерации рассчитываются средние значения целевой функции всех популяций, в которых запущен эволюционный процесс. Также используются графики среднего и минимального значения целевой функции, значения которой рассчитываются в точках миграции.

Для анализа работы нечеткого логического контроллера используются графики изменения входных и выходных параметров контроллера (рис. 4).

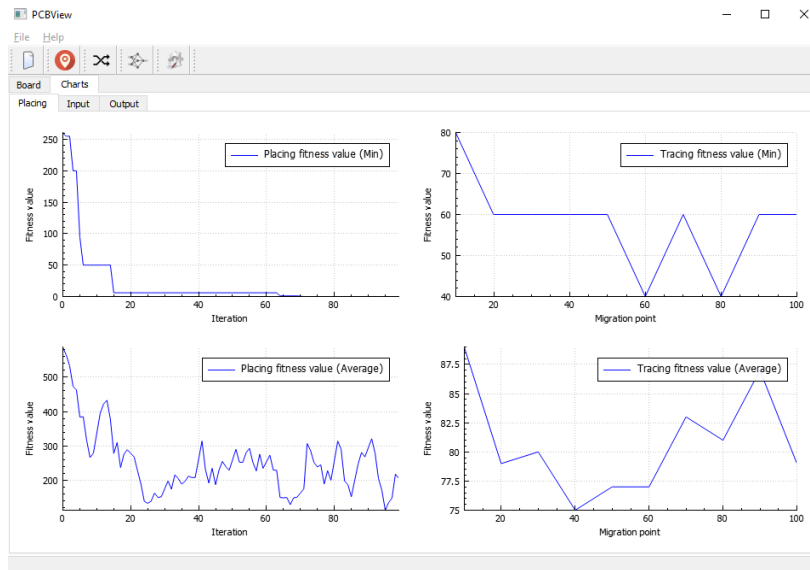


Рис. 4. Графики целевых функций

Заключение. Эффективность работы контроллера можно повысить введением блока обучения на базе нейронной сети. Коэффициенты сети можно искать случайным или направленным поиском. В качестве инструмента направленного поиска возможно использование генетического алгоритма. В результате использования блока обучения были определены значения соответствующие оптимальным параметрам алгоритма.

Проанализировав полученные данные работы параллельного алгоритма, можно сделать вывод о том, что для получения наилучшего решения задач размещения и трассировки необходимо использовать 4 параллельных популяции (потока). Дальнейшее увеличение количества потоков негативно отражается на результате решения задач.

Анализ результатов проведенных вычислительных экспериментов позволяет сделать однозначный вывод о том, что использование предложенного гибридного подхода к решению поставленной задачи позволяет добиться существенного улучшения качества получаемых решений. Выигрыш в качестве получаемых решений при использовании нечеткого логического контроллера составляет порядка 25 % (25,6 % – для задач с 50 элементами; 24,44 % – для 100 элементов; 24,78 % – для 150 элементов) по сравнению с задачами, решаемыми без использования нечеткого логического контроллера.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *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. – P. 683-712.
2. *Charles J. Alpert, Dinesh P. Mehta, Sachin S. Sapatnekar.* Handbook of algorithms for physical design automation. CRC Press, New York, USA, 2009.
3. *Shervani N.* Algorithms for VLSI physical design automation. Kluwer Academy Publisher, USA, 1995. – 538 p.
4. *Гладков Л.А., Курейчик В.М., Курейчик В.В., Сороколетов П.В.* Биоинспирированные методы в оптимизации. – М.: Физматлит, 2009. – 384 с.

5. *Гладков Л.А., Курейчик В.В., Курейчик В.М.* Генетические алгоритмы. – М.: Физматлит, 2010. – 368 с.
6. *Гладков Л.А.* О некоторых подходах к построению гибридных интеллектуальных систем для решения графовых задач // *Новости искусственного интеллекта.* – 2000. – № 3. – С. 71-90.
7. *Michael A., Takagi H.* Dynamic control of genetic algorithms using fuzzy logic techniques // *Proceedings of the Fifth International Conference on Genetic Algorithms.* – Morgan Kaufmann, 1993. – P. 76-83.
8. *Lee M.A., Takagi H.* Integrating design stages of fuzzy systems using genetic algorithms // *Proceedings of the 2nd IEEE International Conference on Fuzzy System.* – 1993. – P. 612-617.
9. *Herrera F., Lozano M.* Fuzzy Adaptive Genetic Algorithms: design, taxonomy, and future directions // *Soft Computing.* – Springer-Verlag, 2003. – No. 7. – P. 545-562.
10. *Ярушкина Н.Г.* Основы теории нечетких и гибридных систем. – М.: Финансы и статистика, 2004. – 320 с.
11. *Батыришин И.З., Недосекин А.О. и др.* Нечеткие гибридные системы. Теория и практика / под ред. Н.Г. Ярушкиной. – М.: Физматлит, 2007.
12. *Кныш Д.С., Курейчик В.М.* Параллельные генетические алгоритмы: Проблемы, обзор и состояние // *Известия РАН. Теория и системы управления.* – 2010. – № 4. – С. 72-82.
13. *Rodriguez M.A., Escalante D.M., Peregrin A.* Efficient distributed genetic algorithm for rule extraction // *Applied Soft Computing.* – 2011. – Vol. 11. – P. 733-743.
14. *Alba E., Tomassini M.* Parallelism and evolutionary algorithms // *IEEE T. Evolut. Comput.* – 2002. – Vol. 6. – P. 443-461.
15. *Гладков Л.А.* Решение задач поиска и оптимизации решений на основе нечетких генетических алгоритмов и многоагентных подходов // *Известия ТРТУ.* – 2006. – № 8 (63). – С. 83-88.
16. *Пегат А.* Нечеткое моделирование и управление. – М.: БИНОМ. Лаборатория знаний, 2009. – 359 с.
17. *Gladkov L.A., Gladkova N.V., Leiba S.N.* Manufacturing Scheduling Problem Based on Fuzzy Genetic Algorithm // *Proc. of IEEE East-West Design & Test Symposium (EWDTS'2014).* Kiev, Ukraine, September 26 – 29, 2014. – P. 209-213.
18. *Gladkov L.A., Gladkova N.V., Legebokov A.A.* Organization of Knowledge Management Based on Hybrid Intelligent Methods // *Software Engineering in Intelligent Systems. Proceedings of the 4th Computer Science On-line Conference 2015 (CSOC 2015), Vol 3: Software Engineering in Intelligent Systems.* – Springer International Publishing, Switzerland 2015. – P. 107-113.
19. *Гладков Л.А., Гладкова Н.В., Лейба С.Н.* Размещение элементов схем ЭВА на основе гибридных интеллектуальных методов // *Известия ЮФУ. Технические науки.* – 2015. – № 4 (165). – С. 25-36.
20. *Гладков Л.А., Лейба С.Н., Тарасов В.Б.* Разработка и программная реализация гибридного алгоритма решения оптимизационных задач автоматизированного проектирования // *Программные продукты и системы.* – 2018. – Т. 31, № 3. – С. 569-580.

REFERENCES

1. *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, pp. 683-712.
2. *Charles J. Alpert, Dinesh P. Mehta, Sachin S. Sapatnekar.* Handbook of algorithms for physical design automation. CRC Press, New York, USA, 2009.
3. *Shervani N.* Algorithms for VLSI physical design automation. Kluwer Academy Publisher, USA, 1995, 538 p.
4. *Gladkov L.A., Kureychik V.M., Kureychik V.V., Sorokoletov P.V.* Bioinspirirovannye metody v optimizacii [Bioinspired methods in optimization]. Moscow: Fizmatlit, 2009, 384 p.
5. *Gladkov L.A., Kureychik V.V., Kureychik V.M.* Geneticheskie algoritmy [Genetic algorithms]. Moscow: Fizmatlit, 2010, 368 p.

6. Gladkov L.A. O nekotorykh podkhodakh k postroeniyu gibridnykh intellektual'nykh sistem dlya resheniya grafovyykh zadach [On some approaches to the construction of hybrid intelligent systems for solving graph problems], *Novosti iskusstvennogo intellekta* [News of artificial intelligence], 2000, No. 3, pp. 71-90.
7. Michael A., Takagi H. Dynamic control of genetic algorithms using fuzzy logic techniques, *Proceedings of the Fifth International Conference on Genetic Algorithms*. Morgan Kaufmann, 1993, pp. 76-83.
8. Lee M.A., Takagi H. Integrating design stages of fuzzy systems using genetic algorithms // Proceedings of the 2nd IEEE International Conference on Fuzzy System. – 1993. – P. 612-617.
9. Herrera F., Lozano M. Fuzzy Adaptive Genetic Algorithms: design, taxonomy, and future directions, *Soft Computing*. Springer-Verlag, 2003, No. 7, pp. 545-562.
10. Yarushkina N.G. Osnovy teorii nechetkikh i gibridnykh sistem [Fundamentals of the theory of fuzzy and hybrid systems]. Moscow: Finansy i statistika, 2004, 320 p.
11. Batyrshin I.Z., Nedosekin A.O. i dr. Nechetkie gibridnye sistemy. Teoriya i praktika [Fuzzy hybrid systems. Theory and practice], ed. by N.G. Yarushkinoy. Moscow: Fizmatlit, 2007.
12. Knysh D.S., Kureychik V.M. Parallelnye geneticheskie algoritmy: Problemy, obzor i sostoyanie [Parallel genetic algorithms: Problem overview and state], *Izvestiya RAN. Teoriya i sistemy upravleniya* [Journal of Computer and Systems Sciences International], 2010, No. 4, pp. 72-82.
13. Rodriguez M.A., Escalante D.M., Peregrin A. Efficient distributed genetic algorithm for rule extraction, *Applied Soft Computing*, 2011, Vol. 11, pp. 733-743.
14. Alba E., Tomassini M. Parallelism and evolutionary algorithms, *IEEE T. Evolut. Comput.*, 2002, Vol. 6, pp. 443-461.
15. Gladkov L.A. Reshenie zadach poiska i optimizatsii resheniy na osnove nechetkikh geneticheskikh algoritmov i mnogoagentnykh podhodov [The solution of problems of search and optimization of solutions based on fuzzy genetic algorithms and multi-agent approaches], *Izvestiya TRTU* [Izvestiya TSURE], 2006, No. 8 (63), pp. 83-88.
16. Pegat A. Nechetkoe modelirovanie i upravlenie [Fuzzy modeling and control]. Moscow: BINOM. Laboratoriya znaniy, 2009, 359 p.
17. Gladkov L.A., Gladkova N.V., Leiba S.N. Manufacturing Scheduling Problem Based on Fuzzy Genetic Algorithm, *Proc. of IEEE East-West Design & Test Symposium (EWDTS'2014)*. Kiev, Ukraine, September 26–29, 2014, pp. 209-213.
18. Gladkov L.A., Gladkova N.V., Legebokov A.A. Organization of Knowledge Management Based on Hybrid Intelligent Methods, *Software Engineering in Intelligent Systems. Proceedings of the 4th Computer Science On-line Conference 2015 (CSOC 2015), Vol 3: Software Engineering in Intelligent Systems*. – Springer International Publishing, Switzerland 2015, pp. 107-113.
19. Gladkov L.A., Gladkova N.V., Leyba S.N. Razmeshchenie elementov skhem EVA na osnove gibridnykh intellektual'nykh metodov [The placement of the circuit elements of EVA based on hybrid intelligent methods], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2015, No. 4 (165), pp. 25-36.
20. Gladkov L.A., Leyba S.N., Tarasov V.B. Razrabotka i programmaya realizatsiya gibridnogo algoritma resheniya optimizatsionnykh zadach avtomatizirovannogo proektirovaniya [Development and software implementation of a hybrid algorithm for solving optimization problems of computer-aided design], *Programmnye produkty i sistemy* [Software products and systems], 2018, Vol. 31, No. 3, pp. 569-580.

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

Гладков Леонид Анатольевич – Южный федеральный университет; e-mail: leo@tgn.sfedu.ru; 347928, г. Таганрог, пер. Некрасовский, 44; тел.: 88634371625; кафедра САПР; доцент.

Гладкова Надежда Викторовна – e-mail: leo_gladkov@mail.ru; тел.: 88634393260; кафедра ДМ и МО; старший преподаватель.

Лейба Сергей Николаевич – e-mail: lejba.sergej@mail.ru; тел.: 88634371625; кафедра САПР; аспирант.

Gladkov Leonid Anatol'evich – Southern Federal University; e-mail: leo@tgn.sfedu.ru; 44, Nekrasovskiy lane, Taganrog, 347928, Russia; phone: +78634371625; CAD department; associated professor.

Gladkova Nadezhda Viktorovna – e-mail: leo_gladkov@mail.ru; phone: +78634393260; DM&MO department; senior teacher.

Leiba Sergey Nikolaevich – e-mail: lejba.sergej@mail.ru; phone: +78634371625; CAD department; postgraduate student.

УДК 681.5.01:658.512.2

DOI 10.23683/2311-3103-2018-4-24-38

В.Н. Гридин, В.И. Анисимов, М.М. Абухазим**МЕТОДЫ ПОВЫШЕНИЯ ПРОИЗВОДИТЕЛЬНОСТИ СИСТЕМ
МОДЕЛИРОВАНИЯ БОЛЬШИХ ИНТЕГРАЛЬНЫХ СХЕМ НА ОСНОВЕ
КОМПАКТНОЙ ОБРАБОТКИ РАЗРЕЖЕННЫХ МАТРИЦ**

Рассматриваются методы повышения производительности систем моделирования больших интегральных схем на основе компактной обработки разреженных матриц. Показывается, что вследствие наличия в математическом описании разреженных матриц возникает задача изменения традиционных подходов к формированию и решению систем уравнений на основе полного описания используемых матриц. Отмечается, что известные методы компактной обработки разреженных матриц значительно отличаются друг от друга по своей эффективности, а также по трудовым затратам при разработке программного обеспечения, реализующего сжатие данных на этапе формирования описания компонентов и в процессе решения уравнений. Приводится детальный сопоставительный анализ возможных методов компактной обработки разреженных матриц с целью установления рекомендаций по разработке высокопроизводительного программного обеспечения систем моделирования больших интегральных схем на основе сжатия данных. Делается вывод, что наибольшую эффективность с точки зрения экономии памяти представляет метод структурно-симметричных матриц. Вместе отмечается практическая значимость использования метода индексно-адресных матриц, который позволяет при создании высокопроизводительных систем автоматизированного проектирования наиболее простым путем реализовать построение программного обеспечения системы моделирования. Отмечается, что классическая форма всех методов фиксированного формата не позволяет перечислять ненулевые элементы в произвольном порядке и включать дополнительные элементы в компактное описание. Поэтому в классической форме метод структурно-симметричных матриц не позволяет непосредственно применять его для обработки компактного описания при решении систем уравнений вследствие неизбежного появления новых ненулевых элементов в процессе этого решения. В основе предложенной в статье методики сжатия данных на основе структурно-симметричных матриц лежит двухэтапная технология построения программного обеспечения системы, при этом на первом, топологическом этапе решается задача определения формата описания с учетом возможного появления новых ненулевых элементов в процессе решения уравнений моделируемой системы, а на втором этапе формируется математическое описание задачи в полученном на первом этапе расширенном формате.

Системы автоматизированного проектирования; автоматизации схемотехнического проектирования; сжатие данных; моделирование систем; компактная обработка; разреженные матрицы.