

УДК 321.3

**В.В. Курейчик, Вл.Вл. Курейчик**

**БИОИНСПИРИРОВАННЫЙ АЛГОРИТМ РАЗБИЕНИЯ СХЕМ  
ПРИ ПРОЕКТИРОВАНИИ СБИС\***

*Рассмотрена одна из важных задач конструкторского проектирования СБИС – задача разбиения схем. Она относится к классу NP-сложных и трудных задач. В работе описана модель поведения колонии пчел. Предложена модифицированная гибридная архитектура биоинспирированного поиска, использующие многоуровневую эволюцию. Это позволяет распараллеливать процесс решения и частично устранять проблему предварительной сходимости алгоритмов. Принципиальным отличием предложенных методов является разделение процесса поиска на два этапа и применение на каждом из этих этапов различных алгоритмов. Разработан биоинспирированный алгоритм на основе моделирования поведения колонии пчел. Проведенные серии тестов и экспериментов позволили уточнить теоретические оценки временной сложности алгоритмов проектирования и их поведение для схем различной структуры. В лучшем случае временная сложность алгоритмов  $\approx O(n \log n)$ , в худшем случае –  $O(n^3)$ .*

*Гибридный поиск; конструкторское проектирование; биоинспирированный алгоритм; разбиение; генетический алгоритм; пчелиный алгоритм.*

**V.V. Kureichik, Vl.Vl. Kureichik**

**BIOINSPIRED ALGORITHMS FOR PARTITIONING VLSI CIRCUITS  
IN DESIGN**

*The article describes one of the most important tasks of the design of VLSI design - a task partitioning schemes. It belongs to a class of NP-hard and difficult tasks. This paper describes a model of behavior of the colony of bees. A modified hybrid architecture bioinspired search using a multi-level evolution. This allows you to parallelize the process of solving the problem, and partly to eliminate pre-convergence algorithms. The principal difference between the proposed method is found for the separation process and the use of two stages each of the stages of different algorithms. Designed bioinspired algorithm based on modeling the behavior of colonies of bees. Conducted a series of tests and experiments have refined theoretical estimates of the time complexity of designing algorithms and their behavior patterns for different structures. In the best case time complexity of algorithms  $\approx O(n \log n)$ , in the worst case –  $O(n^3)$ .*

*Hybrid search; design engineering; bioinspired algorithm partitioning; genetic algorithm; bees algorithm.*

**Введение.** Одним из важнейших этапов в САПР является конструкторское проектирование. На этапе конструкторского проектирования осуществляется поиск оптимального варианта конструкции, учитывающего как возможности технологической базы производства, так и удовлетворяющего требованиям технического задания. Поэтому математическое обеспечение САПР должно развиваться в направлении поиска новых более эффективных методов структурного синтеза проектных решений.

В связи с трудностями создания общей математической модели, комплексно учитывающей все конструкторско-технологические особенности производства, не представляется возможным предложить алгоритм поиска оптимального конструктивного решения в едином цикле проектирования СБИС [1]. Разработка и реализа-

---

\* Работа выполнена при частичной поддержке РФФИ (проект № 13-01-00371).

ция новых и модифицированных, инспирированных природой методов и алгоритмов для решения отдельных задач этапа конструкторского проектирования до сих пор остаётся актуальной проблемой.

**Основные особенности поведения колонии пчёл.** Алгоритм колонии пчёл был впервые предложен турецким учёным Д. Карабога в 2005 г. [2] для решения задач параметрической оптимизации. В основе этого алгоритма лежит поведение пчёл в живой природе.

В живой природе принцип поиска источника нектара основывается на чётком распределении обязанностей между отдельными индивидами пчелиного улья. Источник нектара характеризуется значимостью, определяемой различными факторами, такими как: удалённость от улья, концентрация нектара, удобство добычи нектара.

Всех пчёл в улье можно разделить на 3 группы:

- 1) рабочие пчёлы;
- 2) пчёлы-исследователи;
- 3) пчёлы-разведчики.

Пчёлы-исследователи занимаются поиском источников нектара и снабжают информацией о качестве исследованных участков остальных пчёл (рабочих). Пчёлы-исследователи владеют такой информацией о данном источнике нектара, как: расстояние и направление от улья, полезность источника. Рабочие пчелы все это время находятся в улье и получают информацию об объекте исследования только от пчёл-исследователей. Пчёлы-разведчики, в свою очередь, осуществляют случайный поиск новых источников нектара. Среднее количество пчел-разведчиков  $r$  в рое составляет 5–10 %.

Все пчёлы  $N$  сортируются в соответствии со значениями ЦФ тех участков, на которых они находятся. Определяется  $e$  элитных участков, на которые направляются  $n$  пчёл-исследователей. Каждая рабочая пчела может полететь к источнику нектара, следуя за пчелой-исследователем, которая нашла путь к цветку. Это достигается за счёт того, что каждый улей имеет так называемую закрытую площадку для поиска  $S$ , на которую пчёлы-исследователи, обнаружившие источники нектара  $e$  пытаются привлечь других рабочих пчёл ( $l = N - n$ ) последовать за ними для поиска нектара в окрестности обнаруженных  $e$  участков, причем первыми исследуются участки с большим значением ЦФ. По достижению области с нектаром пчела добывает нектар и возвращается в улей, после чего, она может выбрать одну из трех альтернатив:

1. Оставить источник нектара и снова стать пчелой-исследователем.
2. Продолжить сбор нектара на том же источнике, пытаясь привлечь других особей своего улья.
3. Стать пчелой-разведчиком и продолжить случайный поиск новых источников.

Пчела выбирает одну из вышеупомянутых альтернатив с некоторой вероятностью. Механизмы, в соответствии с которыми пчела решает следовать за другой пчелой, исследованы недостаточно хорошо, но предполагается, что вербовка среди пчел с математической точки зрения всегда является функцией качества источника нектара. Также отмечено, что не все пчелы начинают поиск одновременно. Таким образом, выполняется разделение функций между пчелами-исследователями и пчелами-разведчиками на улучшенное изучение найденных областей с нектаром и на нахождение новых областей с нектаром, соответственно. За счёт такого разделения достигается эффективная работа всего роя пчёл [3].

Основные особенности поведения пчёл можно объяснить с помощью примера. Рассмотрим случай, когда пчелой-исследователем уже найдено два элитных источника нектара  $A$  и  $B$ . Данная пчела направляется в улей для привлечения дру-

гих пчел к поиску источников нектара в окрестности найденных участков. Когда пчела становится исследователем, она может направиться, как в источник  $A$ , так и в источник  $B$ .

При этом, когда пчела-исследователь возвращается в улей, она может продолжить сбор нектара на найденном элитном участке  $e$ , если нектар ещё остался, привлекая рабочих пчел для сбора нектара в его окрестности, так и отправится на поиск новых элитных участков. Рабочие пчелы ( $l_i$ ) могут, как последовать за пчелой-исследователем, так и продолжить поиск на ( $m-e$ ) участках. После чего, данный цикл может повторяться до тех пор, пока не будет окончательно полностью использован найденный источник нектара.

**Алгоритм колонии пчел.** Идея пчелиного алгоритма заключается в том, что все пчелы на каждом шаге будут выбирать как элитные участки для исследования, так и участки в окрестности элитных, что позволит, во-первых, разнообразить популяцию решений на последующих итерациях, во-вторых, увеличить вероятность обнаружения близких к оптимальным решения.

В ходе разработки данного алгоритма не моделировалась жизнь пчёл, однозначно копирующая существующую природную экосистему, а использовалась имитация колонии как средство оптимизации, при котором система несколько отличается от естественной. Опишем суть работы пчелиного алгоритма, представленного на рис. 1 [4].

1. В соответствии с постановкой задачи разбиения и исходными данными формируется популяция пчёл (хромосом).
2. Отправка пчёл-исследователей. Определение месторасположения источников нектара  $v_i$ . Для каждой пчелы случайным образом задается начальная позиция  $x_i$ .
3. Оценка ЦФ пчёл в популяции. Выбор источника нектара пчелой-исследователем с определённой вероятностью, в зависимости от его качества. Для каждой пчелы определяется лучший (элитный) участок, который она посещала с начала первой итерации, и значение целевой функции  $fit_i$  на этом участке. Участки  $e$ , на которых значения ЦФ больше (элитные участки) отбираются для поиска решений в их окрестностях.
4. Выбор пчёл с лучшими значениями ЦФ с каждого источника.
5. Если решение на исследуемом участке не улучшается с течением нескольких итераций переход к п. 6, иначе п. 3.
6. Отправка пчёл-разведчиков, осуществляющих случайный поиск и оценку их ЦФ.
7. Формирование новой популяции пчёл, в состав которой будут входить как пчелы с лучшими значениями ЦФ с элитных участков, так и пчелы со случайными значениями ЦФ.
8. Конец работы алгоритма.

На 1-ом шаге алгоритма случайным образом задаётся пространство поиска решений  $x_i$  ( $i=1, \dots, m$ ),  $m$  – число источников нектара. На 2-ом шаге  $N$  пчёл располагаются случайно на поверхности (в окрестности)  $m$  участков. По аналогии с эволюционными стратегиями, колонию пчел можно трактовать как популяцию, а отдельную пчелу как индивида (хромосому). Это дает возможность построения гибридной структуры поиска решения, основанной на сочетании эволюционно-генетического поиска с методами роевого интеллекта. Связующим звеном такого подхода является структура данных, описывающая решение задачи в виде хромосомы. Если каждая пчела в колонии представлена как хромосома, то число генов в хромосоме равно числу вершин гиперграфа.

В этом случае возникают некоторые требования к структуре хромосомы и значениям генов:

1. Значения генов могут быть непрерывными или дискретными.
2. Значения генов должны быть независимыми друг от друга, то есть хромосомы должны быть гомологичными.

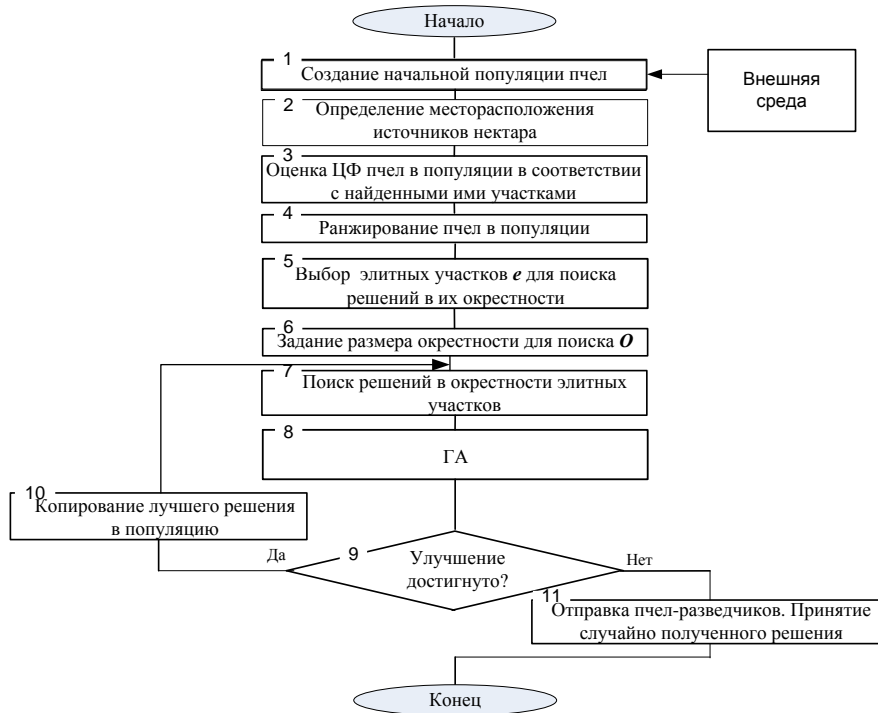


Рис. 1. Схема работы алгоритма колонии пчел

На 3-ем шаге определяются ЦФ участков. Для каждой пчелы-исследователя, число которых определяется как  $\frac{1}{2} \cdot m$ , участок для поиска выбирается по формуле:

$$v_{ij} = x_{ij} + \phi_{ij} * (x_{ij} - x_{kj}), \quad (1)$$

где  $\phi_{ij}$  – случайное число в диапазоне  $[-1; 1]$ ,  $k$  – индекс решения выбранного случайно из колонии ( $k = \text{int}(\text{rand} * m) + 1$ ),  $j = 1, \dots, D$ ,  $D$  – размерность задачи. После определения участка  $v_i$ , полученное решение сравнивается с  $x_i$  и рабочая пчела отправляется на лучший (элитный) из двух участков.

Далее для поиска решений в окрестностях элитных участков решения отбираются с вероятностью, рассчитываемой по следующей формуле:

$$p_i = \frac{fit_i}{\sum_{j=1}^s fit_j}, \quad (2)$$

где  $fit_i$  – значение ЦФ  $x_i$ .

Причём, на этих участках проводятся более детальные исследования, т.е. отправляются больше пчёл, чем на каждый из  $m$ -е участков.

На 4-ом шаге проводится оценка значений ЦФ и выбираются лучшие пчелы в соответствии со значениями ЦФ исследуемых ими участков.

Если по истечении заданного количества итераций решение на данном участке не улучшается, то он исключается из дальнейшего рассмотрения (шаг 5). А пчёлы-исследователи данного участка становятся пчёлами-разведчиками и отбор решений осуществляется случайно (шаг 6) по формуле:

$$x_{ij} = x_j^{min} + (x_j^{max} - x_j^{min}) * rand. \quad (3)$$

Эти пчёлы формируют новую популяцию решений, которая будет участвовать в следующей итерации. Данная операция продолжается до тех пор, пока не будет достигнут критерий останова алгоритма. В конце работы алгоритма популяция решений будет состоять из двух частей: пчёлы с лучшими значениями ЦФ элитных участков, а также группа пчёл-разведчиков со случайными значениями ЦФ [2–4].

**Модифицированные архитектуры поиска.** Основной трудностью решения задач проектирования с большим количеством локальных оптимумов является предварительная сходимость алгоритмов. Другими словами, попадание решения в один, далеко не самый лучший, локальный оптимум. Для эффективного решения этой проблемы предлагаются параллельные методы и архитектуры [5–8]. Это позволит сократить количество компьютерных ресурсов, время поиска и позволит получать оптимальные и квазиоптимальные результаты за полиномиальное время.

Предлагается идея многоуровневой эволюции, когда биоинспирированный поиск используется на двух уровнях. Архитектура такого поиска приведена на рис. 2.

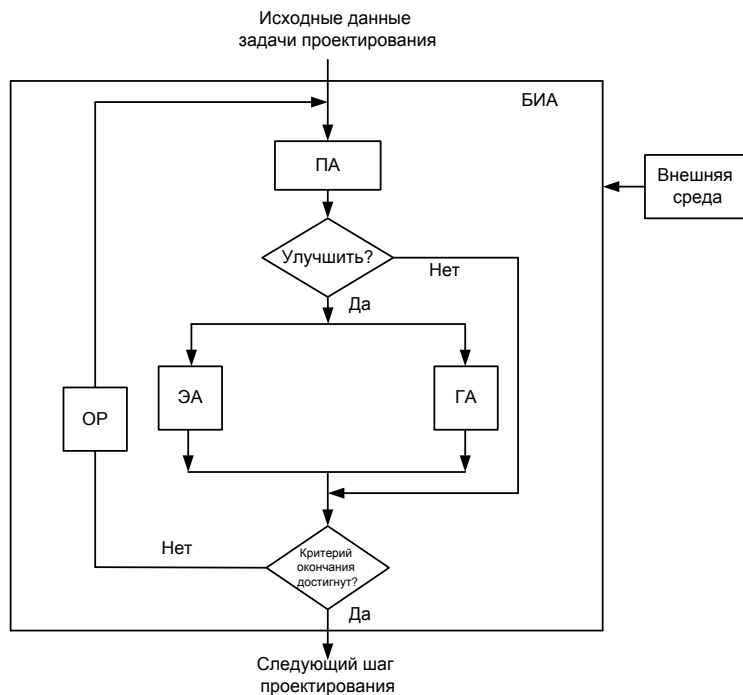


Рис. 2. Архитектура поиска на основе островов

Здесь на первом уровне предлагается использовать пчелиный алгоритм (ПА), позволяющий быстро разбивать поисковое пространство на области высоких значений ЦФ (модель островов).

В таком алгоритме введен адаптивный фильтр, отсекающий решения с низким значением ЦФ. Причем, нижняя граница ЦФ адаптивна в смысле зависимости от решений на каждой генерации. Здесь популяции в генетическом алгоритме (ГА) и в эволюционном алгоритме (ЭА) эволюционируют в течение нескольких заданных генераций независимо друг от друга на определенных «островах», а затем производится обмен генетическим материалом и снова независимая эволюция [9, 10]. Этот процесс происходит в цикле разнообразными способами.

**Заключение.** Приведенные методы и алгоритмы позволяют распараллеливать процесс поиска, эффективно им управлять и получать оптимальные и квази-оптимальные результаты за полиномиальное время. Принципиальным отличием предложенных методов является разделение процесса поиска на два этапа и применение на каждом из этих этапов различных алгоритмов. Это позволяет частично устранить преждевременную сходимость алгоритмов. Разработана программная среда на языке C++. Проведен вычислительный эксперимент. В ходе проведения вычислительного эксперимента были установлены эмпирические зависимости, диапазоны изменения входных параметров и выработан ряд рекомендаций по их оптимальному выбору. Проведенные серии тестов и экспериментов позволили уточнить теоретические оценки временной сложности алгоритмов проектирования 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. Курейчик В.В., Запорожец Д.Ю. Современные проблемы при размещении элементов СБИС // Известия ЮФУ. Технические науки. – 2011. – № 7 (120). – С. 68-73.
2. Karaboga D. An idea based on honey bee swarm for numerical optimization // Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
3. Курейчик В.В., Запорожец Д.Ю. Роевой алгоритм в задачах оптимизации // Известия ЮФУ. Технические науки. – 2010. – № 7 (108). – С. 28-32.
4. Курейчик В.В., Полупанова Е.Е. Эволюционная оптимизация на основе алгоритма колоний пчел // Известия ЮФУ. Технические науки. – 2009. – № 12 (101). – С. 41-46.
5. Курейчик В.М., Кныш Д.С. Проблемы, обзор и параллельные генетические алгоритмы: состояние // Известия РАН. Теория и системы управления. – 2010. – № 4. – С. 72-82
6. Курейчик В.М., Лебедев Б.К., Лебедев О.Б. Разбиение на основе моделирования адаптивного поведения биологических систем. Нейрокомпьютеры. Разработка применение. – 2010. – № 2. – С. 28-33.
7. Курейчик В.В., Курейчик В.М., Гладков Л.А., Сороколетов П.В. Бионспирированные методы в оптимизации. – М.: Физматлит, 2009.
8. Курейчик В.В., Курейчик Вл.Вл. Архитектура гибридного поиска при проектировании // Известия ЮФУ. Технические науки. – 2012. – № 7 (132). – С. 22-27.
9. Гладков Л.А., Курейчик В.В., Курейчик В.М. Генетические алгоритмы. – М.: Физматлит, 2010.
10. Курейчик В.В., Родзин С.И. О правилах представления решений в эволюционных алгоритмах // Известия ЮФУ. Технические науки. – 2010. – № 7 (108). – С. 13-21.

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

**Курейчик Владимир Викторович** – Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Южный федеральный университет»; e-mail: vkur@tsure.ru; 347928, г. Таганрог, пер. Некрасовский, 44; тел.: 88634371651; кафедра систем автоматизированного проектирования; зав. кафедрой; д.т.н.; профессор.

**Курейчик Владимир Владимирович** – e-mail: kureichik@yandex.ru; кафедра систем автоматизированного проектирования; студент.

**Kureichik Vladimir Victorovich** – Federal State-Owned Autonomy Educational Establishment of Higher Vocational Education “Southern Federal University”; e-mail: vkur@tsure.ru; 44, Nekrasovskiy, Taganrog, 347928, Russia; phone: +78634371651; the department of computer aided design; head of department; dr. of eng. sc.; professor.

**Kureichik Vladimir Vladimirovich** – e-mail: kureichik@yandex.ru; 44, Nekrasovskiy, Taganrog, 347928, Russia; the department of computer aided design; student.

УДК 004.89

**А.Ю. Кудяев, А.А. Лежебоков, З.В. Нагоев**

**ВИРТУАЛЬНОЕ ПРОТОТИПИРОВАНИЕ В ИНТЕГРИРОВАННЫХ САПР  
МАШИНОСТРОЕНИЯ И ЭЛЕКТРОНИКИ НА ОСНОВЕ  
ОНТОНЕЙРОМОРФОГЕНЕТИЧЕСКОГО МОДЕЛИРОВАНИЯ\***

*Разработаны основные принципы создания систем виртуального прототипирования для интегрированных САПР машиностроения и электроники, основанные на онтонейроморфогенетических моделях виртуальных прототипов конструкций, материалов, элементной и схмотехнической базы. Построены архитектура интегрированных многомодальных моделей на основе процессов онтонейроморфогенеза и алгоритмы построения онтонейроморфогенетических моделей реальных объектов на основе процессов самоорганизации мультиагентных рекурсивных когнитивных архитектур.*

*Виртуальное прототипирование; онтонейроморфогенетическое моделирование; интегрированные САПР; искусственный интеллект; мультиагентные системы.*

**A.Y. Kudyaev, A.A. Lezhebokov, Z.V. Nagoev**

**VIRTUAL PROTOTYPING IN INTEGRATED CADs OF ENGINEERING  
AND ELECTRONICS BASED ON THE ONTONEURO MORPHOGENETIC  
MODELING**

*The basic principles of creation of systems of virtual prototyping for integrated CADs of engineering and electronics, based on the ontoneuromorphogenetic models of virtual prototypes of constructions, materials, element and circuit base, are developed. The architecture of integrated multimodal models based on the ontoneuromorphogenesis processes and the algorithms of building ontoneuromorphogenetic models of real objects based on processes of self-organization of multiagent recursive cognitive architectures, are built.*

*Virtual prototyping; ontoneuromorphogenetic simulation; integrated CAD systems; artificial intelligence; multi-agent systems.*

**Введение.** Системы виртуального прототипирования (ВП) возникли как современное развитие САПР, опирающееся на два основных научных и технологических достижения – создание систем виртуальной реальности и систем быстрого прототипирования [1]. Системы виртуальной реальности расширили функционал трехмерной визуализации и т.н. «физических движков» САПР, реализующих инженерные расчеты, за счет интерфейсов погружения в иммерсивные [2] виртуальные среды (виртуальные среды полного погружения), а также за счет применения т.н. виртуальных физически корректных сред [3], позволяющих строить динамическую визуализацию в виртуальной среде на основе данных инженерных расчетов, осуществляемых в режиме реального времени. Новые технические средства и возможности актуализировали ряд фундаментальных и прикладных проблем. В част-

---

\* Работа выполнена при поддержке грантов РФФИ № 12-07-00624-а, № 13-07-01003-а.