УДК 681.3.016 DOI 10.31114/2078-7707-2018-1-180-187

Гибридный биоинспирированный алгоритм формирования линеек стандартных ячеек при проектировании топологии СБИС

Б.К. Лебедев, О.Б. Лебедев

Южный федеральный университет, г. Таганрог, lebmius45@mail.ru

***Аннотация* — Предлагается гибридный алгоритм одномерной упаковки, основанный на интеграции муравьиного и генетического алгоритмов для решения задачи формирования линеек стандартных ячеек при проектировании топологии СБИС. Описываются поисковые процедуры в пространстве решений, способы отложения и испарения феромона, механизмы генетического поиска. Для проведения экспериментов были использованы известные тестовые задачи, представленные в библиотеке OR-объектов - (OR-Library Beasley). Временная сложность алгоритма, полученная экспериментальным путем, совпадает с теоретическими исследованиями и для рассмотренных тестовых задач составляет $O(n^2)$ - $O(n^3)$. Разработанный алгоритм позволил получить оптимальные решения для всех задач набора.**

***Ключевые слова* — СБИС, размещение стандартных ячеек, одномерная упаковка, муравьиный алгоритм, генетический поиск, гибридизация, оптимизация.**

I. Введение

Структура топологии полузаказной СБИС на основе базовых стандартных библиотечных элементов отличается использованием заранее спроектированных библиотечных элементов и макроблоков. Размещение блоков и трассировка межсоединений определяется макрокомпоновкой топологии полузаказной СБИС. Библиотечные элементы-ячейки в такой СБИС могут располагаться в виде совокупности линеек, имеющих определенные размеры. Размеры линеек определяются размерами входяших в их состав стандартных библиотечных элементов. Полузаказная структура СБИС строится путем соединения этих ячеек-элементов в логические схемы в соответствии с принципиальной схемой логического устройства, оформленной в виде списка логических сетей в формате EDIF или структурного кода VHDL/Verilog. Такой подход предполагает решение задачи размещения в три этапа. В этой задаче минимизируется количество линеек, занятых элементами, а ее допустимые решения представляют собой все возможные перестановки линеек и элементов в линейках. Основным критерием при формировании множества линеек служит их количество (*F*). Минимизация *F* направлена на минимизацию общей площади кристалла. На первом этапе множество ячеек разбивается на подмножества. Каждое подмножество ячеек входит в состав одной линейки. На втором этапе решается задача относительного расположения линеек друг относительно друга. На третьем этапе решается задача относительного расположения ячеек друг относительно друга в пределах каждой линейки.

Задача распределения множества элементов одинаковой высоты в минимально возможном количестве линеек является классической задачей комбинаторной оптимизации, называемой задачей одномерной упаковки (1 Dimensional Bin Packing, 1DBP) [1]. Задача упаковки одномерных элементов в блоки является *NP*-полной. Поэтому основные усилия исследователей направлены на построение эффективных приближенных алгоритмов.

Несмотря на высокую изученность задачи, существование огромного количества различных методов ее решения [1-7] для ряда контрольных (тестовых) задач оптимальное решение не получено. Более того, на данный момент не существует представленного в литературе универсального алгоритма, способного одинаково эффективно решать все тестовые задачи. Недостатки известных алгоритмов, применяемых для решения задачи упаковки блоков, заключаются в низкой степени учета особенностей задачи, что приводит к излишним требованиям по объему памяти, времени работы и ухудшению качества получаемых решений. Анализ решений, получаемых алгоритмами, построенными на различных метаэвристиках и эвристиках [1-7] показывает, что большинство из них может быть улучшено.

Основу любого алгоритма одномерной упаковки составляют две процедуры: процедура выбора очередного элемента (ПВЭ) и процедура выбора очередного блока (ПВБ) для назначения в него выбранного элемента. В зависимости от способов, лежащих в основе этих процедур, алгоритмы одномерной упаковки можно разделить на 4 класса.

1. Порядок выбора элементов и блоков в ПВЭ и ПВБ задается априори. 2. Порядок выбора элементов в ПВЭ задается априори, блоков - в ПВБ на основе эвристик. 3. Порядок выбора элементов в ПВЭ на основе эвристик, блоков в ПВБ - априори. 4. Порядок выбора элементов и блоков в ПВЭ и ПВБ на основе

эвристик. Наименьшей трудоемкостью обладают алгоритмы 1 класса, наибольшей - 4 класса.

Анализ существующих методов и алгоритмов упаковки в блоки [1-7] показал, что по существу в их основе в явном или неявном виде лежит эвристическая процедура упорядочивания исходного списка элементов, последовательно распределяемых по блокам. Как правило, при распределении элементов в соответствии с эвристиками, заложенными в конкретный алгоритм *A*, параллельно формируется список S_a, который легко можно трансформировать в список S_c, и уже на базе S_c с помощью стандартной процедуры упаковки (СПУ) сформировать решение, найденное алгоритмом *A*. Отсюда следует, что центральной задачей одномерной упаковки является задача синтеза последовательности элементов, распределяемых с помощью СПУ.

Результатом непрекращающегося поиска наиболее эффективных методов упаковки стало использование бионических методов и алгоритмов [8-15]. Особенно интенсивно разрабатывается научное направление, объединяющее математические методы, в которых заложены принципы природных механизмов принятия решений [10]. Среди них особенно активно развиваются методы генетического поиска [11] и роевого интеллекта (Swarm Intelligence) [9, 12-14], в которых совокупность простых агентов конструирует стратегию своего поведения без наличия глобального управления. Эффективность биоинспирированного поиска во многом определяется как учетом специфики решаемой задачи, так и использованием новых и модифицированных процедур поиска.

В настоящее время одним из путей повышения эффективности методов решения задач глобального поиска является разработка гибридных алгоритмов. В гибридных алгоритмах преимущества одного алгоритма могут компенсировать недостатки другого. В работе излагается метод решения задачи одномерной упаковки на основе интеграции модифицированных моделей адаптивного поведения муравьиной колонии и генетической эволюции [15]. Предложена композитная архитектура многоагентной системы бионического поиска.

II. Постановка задачи

Проблема одномерной упаковки может быть сформулирована следующим образом. Дано множество элементов $A=\{a_i|i=1,2,\ldots,n\}$. Вес элементов задается множеством $W=\{w_i \mid i=1,2,\ldots,n\}$. Необходимо сформировать *F* узлов, т.е. множество *A* разбить на *F* непустых и непересекающихся подмножеств A_j. $A=\cup A_j$, $A_j\neq\varnothing$, $(\forall i,j)\ [A_i\cap A_j=\varnothing]$. Обозначим через W_j вес элементов, назначенных в узел A_j. $W_j=\sum w_i$, для $\{i|a_i\in A_j\}$. Задается максимально допустимый суммарный вес - *C* элементов, назначенных в каждый узел: $W_j\leq C$. Критерий оптимизации – число узлов *F*. Цель оптимизации - минимизация *F*. Отметим, что число узлов *F* - это фактически количество упакованных линеек.

В работе в качестве интерпретации решения - структуры данных, несущей информацию об упаковке, используется последовательность элементов (упорядоченный список), задающая порядок их упаковки [1-3].

Данное представление удобно для его использования в различных метаэвристиках (генетические алгоритмы, роевые алгоритмы), поскольку они как раз и работают с последовательностями, кодирующими упаковку.

Рассмотрим стандартную процедуру упаковки (СПУ) [2]. Дан список S элементов для упаковки. Предположим, что элементы для упаковки отсортированы некоторым образом. Первый элемент списка S упаковывается в блок A1. Элементы 2, …, n рассматриваются в порядке возрастания их индексов: рассматриваемый элемент упаковывается в текущий блок, если не происходит переполнения блока; в противном случае он упаковывается в новый блок, который становится текущим. Процесс упаковки завершается после просмотра всех элементов списка S. Временная сложность алгоритма - O(n). Очевидно, что существует такая последовательность элементов в списке, при которой решение задачи упаковки будет оптимальным. Таким образом, решение задачи упаковки сводится к нахождению списка используемого стандартной процедурой упаковки.

Результатом непрекращающегося поиска наиболее эффективных методов упаковки стало использование бионических методов и алгоритмов.

III. Основные принципы построения гибридного алгоритма

В работе рассматривается гибридный алгоритм, построенный на интеграции муравьиного и генетического алгоритмов. Построение алгоритма производилось в соответствии с классификацией Ванга [10] на основе низкоуровневой гибридизация вложением, предполагающей сращивание гибридизируемых алгоритмов. Отличительным признаком низкоуровневой гибридизации является то, что объединяемые алгоритмы сохраняют значительную автономию. Работа этих алгоритмов базируется на использовании коллективной эволюционной памяти (КЭП). КЭП муравьиной колонии представляет собой набор статистических показателей, отражающих для каждого фрагмента решения число θ вхождений фрагмента в лучшие решения на предыдущих итерациях алгоритма и число ψ показывающее полезность фрагмента при построении решений на предыдущих итерациях алгоритма. КЭП генетического алгоритма это популяция лучших решений, сформированная в результате эволюционных преобразований на предыдущих итерациях алгоритма.

Интеграция метаэвристик популяционных алгоритмов обеспечивает более широкий обзор пространства поиска и более высокую вероятность локализации глобального экстремума задачи.

Связующим звеном такого подхода является структура данных, описывающая решение задачи. В работе предлагается подход к построению структур данных, обеспечивающих возможность одновременного использования в муравьином и генетическом алгоритмах.

В качестве структуры данных, несущих информацию об упаковке, используется последовательность номеров элементов (список), представляющая порядок их укладки [2, 6].

Интерпретацией решения, формируемого муравьем zk, является маршрут Mk на графе поиска решений, включающий все вершины графа, с разбиением его на части. Каждая часть Mkj∈Mk включает вершины, соответствующие элементам, помещаемым в блок Aj. Число частей служит оценкой решения. Трансформация маршрута в список тривиальна. Интерпретацией решения, формируемого генетическим алгоритмом, является хромосома, структура которой идентична структуре списка. Для получения решения хромосома декодируется с помощью СПУ.

Предлагается композитная архитектура многоагентной системы бионического поиска для решения задачи одномерной упаковки на основе интеграции муравьиного алгоритма и генетической эволюции. Структура муравьиного алгоритма служит опорной структурой гибридного алгоритма. Механизмы генетической эволюции используются для модификации популяции решений, формируемых на каждой итерации муравьиной колонией.

IV. Механизмы муравьиной колонии

Метаэвристика муравьиного алгоритма основывается на комбинации двух техник: общая схема строится на базовом методе, в которую включается встроенная процедура. Базовый метод заключается в реализации итерационной процедуры поиска лучшего решения на основе механизмов адаптивного поведения муравьиной колонии. Встроенная процедура – это конструктивный алгоритм построения муравьем некоторой конкретной интерпретации решения [12, 13]. Процесс поиска решений гибридным алгоритмом – итерационный (рис. 1). На первом этапе каждой итерации гибридного алгоритма коллективом муравьев $Z=\{z_k|k=1,2,\ldots,n\}$ формируется множество интерпретаций решений $M=\{M_k|k=1,2,\ldots,n\}$. Поиск решений осуществляется на полном графе поиска решений $G=(X,U)$. Вершины множества $X=\{x_i|i=1,2,\ldots, n\}$ соответствуют элементами списка S. Изначально на подготовительном этапе на всех ребрах графа G откладывается одинаковое (небольшое) количество феромона Φ/v, где $v=|U|$ (см. рис. 1). Параметр Φ задается априори. Интерпретацией решения, сформированного муравьем $z_k \in Z$, является маршрут M_k в графе G, включающий все вершины. Предложен конструктивный метод с линейной временной сложностью для решения отдельным муравьем задачи одномерной упаковки большой размерности.

Процесс построения муравьем z_k маршрута M_k в графе поиска решений G пошаговый, начиная со стартовой вершины St. Формирование маршрута M_k связано с процессом заполнения блоков элементами. В результате муравей строит маршрут, разбитый на части. Каждой части M_{kj} соответствует множество X_{kj} элементов, размещенных в одном блоке A_j с соблюдением ограничений $W_j \le C$.

На первом этапе каждой итерации гибридного алгоритма (см. рис. 1) каждый муравей z_k формирует свой собственный маршрут M_k с разбиением его на части. Каждая часть $M_{kj} \in M_k$ включает вершины, соответствующие элементам, помещаемым в блок A_{kj}. Процесс построения маршрута M_k пошаговый. На каждом шаге t агент z_k применяет вероятностное правило выбора следующей вершины для включения ее формируемый маршрут. Муравьи имеют собственную «память». Каждый муравей хранит в памяти список пройденных им вершин графа. Этот список называют *списком запретов (tabu list)* или просто *памятью* муравья. Выбирая вершину для следующего шага, муравей «помнит» об уже пройденных вершинах и не рассматривает их в качестве возможных для включения в маршрут. На каждом шаге список запретов пополняется новой вершиной. В памяти хранится номер j заполняемого блока и остаточная вместимость $\delta_{kj}(t)=C-W_{kj}(t)$ блока A_{kj} на шаге t, где $W_{kj}(t)$ суммарный вес элементов, назначенных в блок A_{kj}.

Введем обозначения: $M_k(t)$ - маршрут, построенный муравьем z_k за t шагов; $X1_k(t) \in X$ - множество вершин, вошедших в состав маршрута $M_k(t)$; $X2_k(t)=X \backslash X1_k(t)$- множество не распределенных вершин. Если $X1_k(t)=X$, а $X2_k(t)=\varnothing$. то процесс построения маршрута завершен.

Если не все элементы упакованы, т.е. $X2_k(t) \ne \varnothing$, то возможны два варианта продолжения процесса построения маршрута. Пусть на шаге t формируется часть $M_{kj}(t)$, являющаяся хвостом $M_k(t)$; $e_k(t)$- последняя вершина части $M_{kj}(t)$ маршрута $M_k(t)$. Формируется множество вершин $X_k(t) \in X2_k(t)$ таких, что если $x_i \in X_k(t)$, то $\delta_{kj}(t) \ge w_i$ и соответствующий этой вершине элемент может быть зафиксирован в блоке A_j без переполнения, т.е. без превышения ограничения на суммарный вес размешенных в нем элементов. Если $X_k(t)=\varnothing$, то выбирается для заполнения следующий блок A_{j+1}, а в множество вершин $X_k(t)$ включаются вершины множества $X2_k(t)$, т.е. $X_k(t)=X2_k(t)$.

Агент просматривает все вершины $x_i \in X_k(t)$ и для каждой вершины рассчитывается параметр f_{ik} - суммарный уровень феромона на ребре графа G, связывающем x_i с вершиной $e_k(t)$. Вероятность P_{ik} включения вершины $x_i \in X_k(t)$ в часть $M_{kj}(t)$ маршрута $M_k(t)$ определяется следующим соотношением:

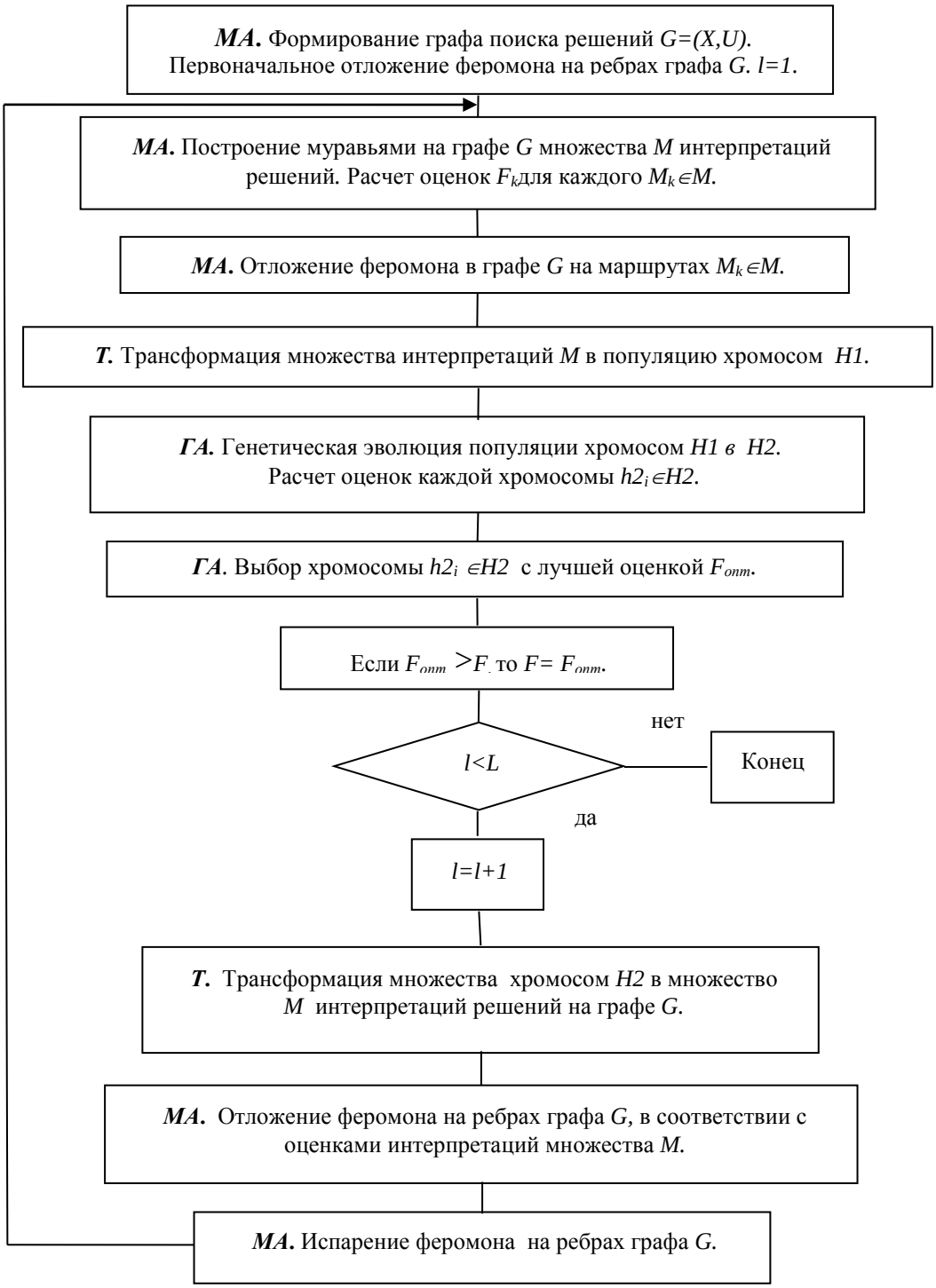

Рис. 1. Структурная схема гибридного алгоритма

$$P_{ik}=f_{ik}/\sum_i f_{ik} \qquad (1)$$

Агент z_k с вероятностью P_{ik} выбирает одну из вершин $x_i \in X_k(t)$, которая включается в конец маршрута $M_k(t)$. После этого фиксируется размещение элемента $x_i \in X_k(t)$ в блоке A_j. Если $X_k(t)$ оказывается пустым, то выбирается вершина не входящая в маршрут $M_k(t)$, а соответствующий ей элемент помещается в следующий блок A_{j+1}. На втором этапе итерации каждый муравей z_k откладывает феромон на рёбрах маршрута M_k графе G. В работе используется циклический (ant-cycle) метод муравьиных систем. В этом случае феромон откладывается агентами на ребрах графе G после полного формирования решения. Количество феромона τ_k, откладываемое муравьем z_k на каждом ребре построенного маршрута, определяется следующим образом:

$$\tau_k = \Phi / F_k, \qquad (2)$$

где Φ - общее количество феромона, откладываемое муравьем на ребрах построенного маршрута, F_k – целевая функция для решения, полученного муравьем z_k. Чем меньше F_k, тем больше феромона откладывается на ребрах построенного маршрута и, следовательно, тем больше вероятность выбора этих ребер при построении маршрутов на следующей итерации.

Далее, на этапе стыковки муравьиного и генетического алгоритмов сформированное муравьиной колонией множество интерпретаций *M* трансформируется в популяцию хромосом *H1*.

V. Механизмы генетического поиска

На третьем этапе гибридного алгоритма выполняется генетическая эволюция популяции хромосом *H1* в *H2*.

На каждой генерации вначале реализуются операторы кроссинговера, мутации, а затем расширенная популяция подвергается редукции с помощью селективного отбора, т.е. уменьшению до начального объема.

Основными генетическими операторами являются кроссинговер и мутация [11]. Структура кроссинговера в основном определяет эффективность ГА.

В работе предлагается модифицированный оператор кроссинговера, который выполняется следующим образом.

Пусть имеются две родительские хромосомы:

H_1=1,2,15,14,3,5,4,13,11,10,6,7,8,12,9;

H_2=4,3,1,14,13,10,9,2,7,5,11,15,6,8,12.

На первом этапе формируется общий список H_0 путем “склеивания” хромосом. Последовательно просматриваются локусы хромосом H_1 и H_2. При рассмотрении текущего локуса в формируемый список H_0 первым включается ген хромосомы H_1, а затем ген хромосомы H_2.

При втором способе в формируемый список первым включается ген хромосомы H_2, а затем ген хромосомы H_1.

При третьем способе в формируемый список H_0 ген хромосомы H_1 включается первым с вероятностью P = 0,5.

Для представленных выше хромосом H_1 и H_2, сформированный на первом этапе путем “склеивания” хромосом по первому способу, список H_0 имеет вид:

H_0=1,4,2,3,15,1,14,14,3,13,5,10,4,9,13,2,11,7,10,5,6,11,7,15,8,6,12,8,9,12.

В списке H_0 содержатся по два элемента с одинаковым значением.

На втором этапе осуществляется расщепление списка H_0 на два списка путем последовательного просмотра его элементов, начиная с первого. В список H_1 включаются те элементы списка H_0, которые встречаются при просмотре впервые. В список H_2 включаются оставшиеся элементы.

H_1=1,4,2,3,15,14,13,5,10,9,11,7, 6,8,12.

H_2=1,14,3,4,13,2,10,5,11,7,15,6 ,8,9,12.

Каждый полученный список определяет новую хромосому. Таким образом, в результате использования модифицированного оператора кроссинговера из двух родительских хромосом тремя способами формируется шесть хромосом потомков.

В работе используются два оператора мутации. Пусть выбрана хромосома H_i, в соответствии с которой стандартной процедурой осуществлена упаковка, т.е. имеется некоторое решение задачи упаковки n элементов в F узлов. n_i –число элементов, уложенных в i-ый блок.

Первый оператор осуществляет перемещение случайного элемента из одного случайного блока в другой случайный блок.

С помощью второго оператора новое решение создается путем взаимного обмена двух (случайных) элементов из двух (случайных) блоков.

Как видно из алгоритмов, реализующих процедуры кроссинговера и мутации, временная сложность операторов кроссинговера t_k и мутации $t_м$ применительно к одной хромосоме имеют линейную зависимость и оценки временной сложности имеют вид: $t_k=O(L_h)$, $t_м=O(L_h)$, где L_h – длина хромосомы. Для получения решения хромосома декодируется с помощью стандартной процедуры упаковки и определяется ее оценка.

На каждой генерации (число генераций равно T) сначала реализуются операторы кроссинговера и мутации. Выбор родительской пары осуществляется одним из способов, задаваемых при настройке механизмов генетического поиска. Это может быть "принцип рулетки", выбор на основе рейтинга и т.д.

Затем расширенная популяция подвергается редукции с помощью селективного отбора, то есть уменьшению до начального объема. Селекция популяции после выполнения операций кроссинговера и мутации осуществляется с использованием принципа “рулетки”. В общем случае временные затраты в пределах поколения имеют оценку $O(n)$.

После завершения заданного числа итераций генетического алгоритма выбирается лучшее решение, полученное на протяжении всех выполненных итераций.

Если все итерации гибридного алгоритма выполнены, то - конец работы алгоритма, в противном случае в начале следующей итерации гибридного алгоритма для стыковки процедур генетического и муравьиного алгоритмов популяция хромосом *H2* трансформируется в множество *M* интерпретаций

решений (маршрутов) на графе *G*. Далее выполняются заключительные процедуры базового метода муравьиного алгоритма. Производится отложение феромона на ребрах графа *G*, в соответствии с оценками интерпретаций множества *M*. Затем осуществляется общее испарение феромона на ребрах полного графа *G* в соответствии с формулой:

$$f_i = f_i(1-\rho), \qquad (3)$$

где f_i – общее количество феромона, отложенного на ребре u_i графа *G*, ρ – коэффициент обновления.

Далее осуществляется переход на первую процедуру муравьиного алгоритма, реализующую построение муравьями на графе *G* множества *M* интерпретаций решений.

VI. Экспериментальные исследования

Целью исследования программы было определение качество решения, которое определяется величиной разницы (отклонением) между количеством блоков в известном лучшем решении (либо глобальном оптимуме) и количеством блоков в решении, достигнутом гибридным алгоритмом. При исследовании сходимости алгоритмов для каждого эксперимента запоминался номер генерации, после которой не наблюдалось улучшения оценки. В каждой серии из 50 испытаний определялись среднее значение числа генераций, после которого не наблюдалось улучшения оценки. Для каждой серии испытаний определялось лучшее решение. На основе обработки экспериментальных исследований для алгоритмов была построена средняя зависимость качества решений от числа итераций. Установлено что начальное количество феромона Q должно быть в 14 раз больше среднего количества значения ферромона τk(l), откладываемого агентами на каждой итерации. Коэффициент обновления ρ=0,95.

Исследования показали, что число итераций, при которых алгоритм находил лучшее решение, лежит в пределах 111-131. Алгоритм сходится в среднем на 127 итерации (рис. 2).

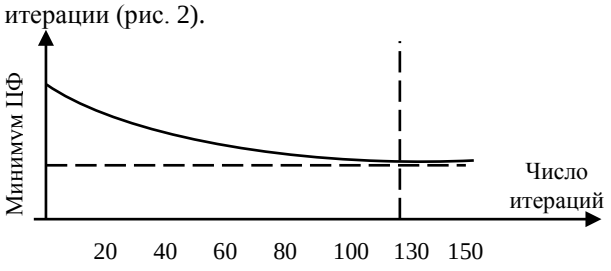

Рис.2. Зависимость качества гибридного алгоритма от числа итераций

Для проведения объективных экспериментов были использованы доступные в библиотеке OR-объектов (OR-Library Beasley) контрольные задачи класса U, на которых был протестирована разработанная программа. Класс задач состоит из элементов, веса которых равномерно распределены на интервале (20, 100). Для всех тестовых задач класса U оптимальные решения были получены при помощи точного алгоритма, разработанного Valerio de Carvalho [16]. Данный алгоритм построен по методу ветвей и границ и обладает значительной трудоемкостью. Для решения каждой из тестовых задач разработанная программа запускалась 10 раз. Каждый запуск являлся независимым. В таблице 1 в столбце «Задача» приведено кодовое название контрольной (тестовой) задачи, «GO» - количество блоков в глобально оптимальном решении, «HB» - количество блоков в лучшем решении, полученном при помощи гибридного алгоритма упаковки, «O» - отклонение полученного решения от глобально оптимального решения.

Таблица 1

Результаты, полученные на контрольных задачах наборов U120 и U250

Задача	GO	HB	O	Задача	GO	HB	O
U120-1	48	48	0	U250-1	99	99	0
U120-2	49	49	0	U250-2	100	100	0
U120-3	46	46	0	U250-3	102	102	0
U120-4	49	49	0	U250-4	100	100	0
U120-5	50	50	0	U250-5	101	101	0
U120-6	48	48	0	U250-6	101	101	0
U120-7	48	48	0	U250-7	102	102	0
U120-8	49	49	0	U250-8	103	103	0
U120-9	50	50	0	U250-9	105	105	0
U120-10	46	46	0	U250-10	101	101	0
U120-11	52	52	0	U250-11	105	105	0
U120-12	49	49	0	U250-12	101	101	0
U120-13	48	48	0	U250-13	105	105	0
U120-14	49	49	0	U250-14	102	102	0
U120-15	50	50	0	U250-15	100	100	0
U120-16	48	48	0	U250-16	105	105	0
U120-17	52	52	0	U250-17	97	97	0
U120-18	52	52	0	U250-18	100	100	0
U120-19	49	49	0	U250-19	100	100	0
U120-20	49	49	0	U250-20	102	102	0

Как видно из таблицы 1 использование разработанной программы позволило получить оптимальные решения для всех задач набора. Следует отметить, что время, затраченное на решение каждой из тестовых задач, указанных в таблицах выше, не превышало 20 секунд. Максимальная эффективность адаптивного поиска обеспечивается при значениях

управляющих параметров: $Q=140$, $\tau_k=10$, $\rho=0.95$, найденных в процессе экспериментальных исследований. Q - априорно задаваемое количество феромона, τ_k – общее количество феромона, откладываемое k-м муравьем, ρ-коэффициент обновления. Увеличение размера популяции больше чем 100 не дает улучшения результатов, а только увеличивает время сходимости алгоритма, а также время его работы.

Временная сложность алгоритма при фиксированных значениях размера популяции и числа итераций лежит в пределах $O(n)$. Общая оценка временной сложности при любом подходе к гибридизации в пределах $O(n^2)$- $O(n^3)$.

VII. Заключение

Предложены новые механизмы решения задачи упаковки, использующие математические методы, в которых заложены принципы природных механизмов принятия решений. Интегрируемые алгоритмы базируются на использовании коллективной эволюционной памяти, имеющую различные принципы ее организации. В связи с этим интеграция метаэвристик муравьиного алгоритма и генетической эволюции обеспечивает более широкий обзор пространства поиска и более высокую вероятность локализации глобального экстремума задачи. Связующим звеном такого подхода является структура данных, описывающая решение задачи. В работе предлагается подход к построению структур данных, обеспечивающих возможность одновременного использования в муравьином и генетическом алгоритмах. В отличие от канонической парадигмы муравьиного алгоритма муравьем на вершинах графа поиска решений $G=(X,U)$ строится маршрут с разбиением на части. Это позволило снизить комбинаторную сложность задачи.

Экспериментальные исследования проводились на IBM PC. Временная сложность алгоритма (ВСА), полученная экспериментальным путем, практически совпадает с теоретическими исследованиями и для рассмотренных тестовых задач составляет (ВСА ≈ $O(n^2)$ - $O(n^3)$).

Для проведения объективных экспериментов были использованы известные тестовые задачи, представленные в литературе и Интернет. Задачи, на которых был протестирован разработанный алгоритм, доступны в библиотеке OR-объектов (OR-Library Beasley). Для составления достоверных выводов был проведен не один, а серия опытов-экспериментов. Разработанный гибридный алгоритм позволил получить оптимальные решения для всех задач набора.

Поддержка

Работа выполнена при финансовой поддержке гранта РФФИ № 17-07-00997 а

Литература

[1] Bischoff E. E. and Wäscher G. Cutting and packing. European Journal of Operational Research, 84, 1995. P. 503–505.

[2] Потарусов Р.В., Курейчик В.М. Проблема одномерной упаковки элементов // Известия ТРТУ. Изд-во ТРТУ, 2006. №8. С. 88 – 93.

[3] Ross P., Marin-Blazquez J.G., Schulenburg, S., and Hart E. Learning a Procedure That Can Solve Hard Bin-Packing Problems: A New GA-Based Approach to Hyper-heurstics// Proceeding of the Genetic and Evolutionary Computation Conference, GECCO 2003, Chicargo, Illinois, USA, 2003, 1295-1306.

[4] Gupta J. N. and Ho J. C. A New Heuristic Algorithm for the One-dimensional Bin-packing Problem, Production Planning & Control. 10, 1999, 598-603.

[5] Levine J. and F. Ducatelle. Ant Colony Optimization and Local Search for Bin Packing and Cutting Stock Problems. Centre for Intelligent Systems and their Applications, School of Informatics, University of Edinburgh, 2003.

[6] Fleszar, K. and Hindi, K. S. New Heuristics for One-dimensional Bin-packing, Computers & Operations Research. 29, 2002, 821-839.

[7] Alvim Adriana C. F., Glover Fred S., Ribeiro Celso C., and Aloise Dario J. Local search for the bin packing problem // Journal of Heuristics, 10, 2004, 205–229.

[8] Курейчик В.М., Лебедев Б.К., Лебедев О.Б. Поисковая адаптация: теория и практика. М.: Физматлит, 2006.

[9] Engelbrecht A. P. Fundamentals of Computational Swarm Intelligence. John Wiley & Sons, Chichester, UK, 2005.

[10] Карпенко А.П. Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой: учебное пособие – М: Издательство МГТУ им. Н.Э. Баумана, 2014.

[11] Курейчик В.М., Лебедев Б.К., Лебедев О.Б. Решение задачи размещения на основе эволюционного моделирования. //Известия Академии наук. Теория и системы управления. 2007. №4. с. 78-90.

[12] M. Dorigo and T. Stützle. Ant Colony Optimization. MIT Press, Cambridge, MA, 2004.

[13] Лебедев О.Б. Модели адаптивного поведения муравьиной колонии в задачах проектирования. - Таганрог: Изд-во ЮФУ, 2013.

[14] Лебедев Б.К., Лебедев О.Б. Муравьиные алгоритмы разбиения, использующие представления задачи, отличные от канонического //Вестник Ростовского государственного университета путей сообщения – Ростов-на-Дону: Изд-во РГУПС, 2016, №3 (63). – С. 42-47.

[15] Лебедев Б.К., Лебедев О.Б., Лебедев В.Б. Гибридизация роевого интеллекта и генетической эволюции на примере размещения. ЭЛЕКТРОННЫЙ НАУЧНЫЙ ЖУРНАЛ: ПРОГРАММНЫЕ ПРОДУКТЫ, СИСТЕМЫ И АЛГОРИТМЫ. DOI: 10.15827/2311-6749.25.280. http://swsys-web.ru/hybridization-of-swarm-intelligence-and-genetic-evolution.html

[16] Valerio de Carvalho, J. M. Exact Solution of Bin-packing Problems Using Column Generation and Branch-and-bound //Annals of Operations Research. 86, 1999, 629-659.

Hybrid Bioinspired Algorithm for Forming Standard Cell Lines in the Design of the VLSI Topology

B.K. Lebedev, O.B. Lebedev

Southern Federal University, Таганрог, lebmius45@mail.ru

***Abstract* — A hybrid one-dimensional package algorithm based on the integration of the ant algorithm and genetic search is proposed to solve the problem of forming standard cell lines in the design of the VLSI topology. Integrable algorithms are based on the use of collective evolutionary memory, which has different principles of its organization. In this regard, the integration of metaheuristics of the ant algorithm and genetic evolution provides a broader overview of the search space and a higher probability of localizing the global extremum of the problem. The connecting link of this approach is the data structure describing the solution of the problem. The paper proposes an approach to the construction of data structures that provide the possibility of simultaneous use in ant and genetic algorithms. Interpretation of the decision formed by the ant is a route on the solution search graph. Transformation of the route into the list is trivial. Interpretation of the solution formed by the genetic algorithm is a chromosome whose structure is identical to the list structure. To obtain a solution, the chromosome is decoded using a standard packing procedure.**

The composite architecture of the multi-agent bionic search system is proposed to solve the problem of one-dimensional packaging based on the integration of the ant algorithm and genetic evolution. The structure of the ant algorithm serves as the supporting structure of the hybrid algorithm. Mechanisms of genetic evolution are used to modify the population of solutions formed at each iteration by an ant colony. Search procedures in the space of solutions, methods of deposition and evaporation of pheromone, mechanisms of genetic search are described. Unlike the canonical paradigm of the ant algorithm, an ant on the vertices of the search graph $G = (X, U)$ constructs a route with a partition into parts. This made it possible to reduce the combinatorial complexity of the problem. To conduct objective experiments, we used well-known test problems presented in the literature and the Internet. The tasks on which the developed algorithm was tested are available in the OR-objects library (OR-Library Beasley). To draw up reliable conclusions, not one, but a series of testing experiments was carried out. The developed hybrid algorithm allowed to obtain optimal solutions for all set tasks. The temporal complexity of the algorithm, obtained experimentally, practically coincides with the theoretical studies and for the considered test problems is $O(n^2)$- $O(n^3)$.

***Keywords* — VLSI; placement of standard cells; one-dimensional packing; ant algorithm; genetic search; hybridization; optimization.**

REFERENCES

[1] Bischoff E. E. and Wäscher G. Cutting and packing. European Journal of Operational Research, 84, 1995. P. 503–505.

[2] Potarousov R.V., Kureichik V.M. Problema odnomernoy upakovki elementov (The problem of one-dimensional packing of elements) // Proceedings of TRTU. Publishing House of TRTU, 2006. №8. P. 88 - 93.

[3] Ross P., Marin-Blazquez J.G., Schulenburg, S., and Hart E. Learning a Procedure That Can Solve Hard Bin-Packing Problems: A New GA-Based Approach to Hyper-heurstics// Proceeding of the Genetic and Evolutionary Computation Conference, GECCO 2003, Chicargo, Illinois, USA, 2003, 1295-1306.

[4] Gupta J. N. and Ho J. C. A New Heuristic Algorithm for the One-dimensional Bin-packing Problem, Production Planning & Control. 10, 1999, 598-603.

[5] Levine J. and F. Ducatelle. Ant Colony Optimization and Local Search for Bin Packing and Cutting Stock Problems. Centre for Intelligent Systems and their Applications, School of Informatics, University of Edinburgh, 2003.

[6] Fleszar K. and Hindi K. S. New Heuristics for One-dimensional Bin-packing, Computers & Operations Research. 29, 2002, 821-839.

[7] Alvim Adriana C. F., Glover Fred S., Ribeiro Celso C., and Aloise Dario J. Local search for the bin packing problem // Journal of Heuristics, 10, 2004, 205–229.

[8] Kurejchik V.M., Lebedev B.K., Lebedev O.B. Poiskovaya adaptatsiya: teoriya i praktika. (Search adaptation: theory and practice). Moscow: Fizmatlit,2006.

[9] Engelbrecht A. P. Fundamentals of Computational Swarm Intelligence. John Wiley & Sons, Chichester, UK, 2005.

[10] Karpenko A.P. Sovremennyye algoritmy poiskovoy optimizatsii. Algoritmy, vdokhnovlennyye prirodoy: uchebnoye posobiye (Modern algorithms of search optimization. Algorithms inspired by nature: a tutorial) - M: Izdatelstvo MSTU im. N.E. Baumana, 2014.

[12] M. Dorigo and T. Stützle. Ant Colony Optimization. MIT Press, Cambridge, MA, 2004.

[13] Lebedev O.B. Modeli adaptivnogo povedeniya murav'inoy kolonii v zadachakh proyektirovaniya (Models of adaptive behavior of ants colony in design problems). -Taganrog: Publishing house of SFedU, 2013.

[14] Lebedev B.K., Lebedev O.B. Murav'inyye algoritmy razbiyeniya, ispol'zuyushchiye predstavleniya zadachi, otlichnyye ot kanonicheskogo (Ant partition algorithms using representations of a problem other than the canonical one) // Bulletin of the Rostov State Transport University - Rostov-on-Don: RSTU, 2016, No. 3 (63). - P. 42-47.

[15] Lebedev B.K., Lebedev O.B., Lebedev V.B. Gibridizatsiya royevogo intellekta i geneticheskoy evolyutsii na primere razmeshcheniya (Hybridization of swarm intelligence and genetic evolution using the example of placement). ELEKTRONNYY NAUCHNYY ZHURNAL: PROGRAMMNYY PRODUKTY, SISTEMY I ALGORITMY. DOI: 10.15827/2311-6749.25.280. http://swsys-web.ru/hybridization-of-swarm-intelligence-and-genetic-evolution.html

[16] Valerio de Carvalho, J. M. Exact Solution of Bin-packing Problems Using Column Generation and Branch-and-bound //Annals of Operations Research. 86, 1999, 629-659.