

Б.К. Лебедев, О.Б. Лебедев, А.Е. Лебединский

**ГИБРИДНЫЙ БИОИНСПИРИРОВАННЫЙ АЛГОРИТМ РАЗМЕЩЕНИЯ
БАЗОВЫХ СТАНДАРТНЫХ БИБЛИОТЕЧНЫХ ЭЛЕМЕНТОВ ПРИ
ПРОЕКТИРОВАНИИ ТОПОЛОГИИ ПОЛУЗАКАЗНОЙ СБИС***

Предлагается гибридный алгоритм размещения базовых стандартных библиотечных элементов при проектировании топологии полужаказной СБИС методом одномерной упаковки, основанный на интеграции роевого и генетического алгоритмов. В работе в качестве интерпретации решения – структуры данных, несущей информацию об упаковке (размещении), используется последовательность элементов (приоритетный список), задающая порядок их упаковки. В качестве декодера используется стандартная процедура упаковки (СПУ). Интерпретацией решения задачи размещения, формируемого роевым и генетическим алгоритмами, является упорядоченный список, включающий все ячейки, с разбиением его на части. Каждая часть включает вершины, соответствующие элементам, помещаемым в линейку (блок). Число частей служит оценкой решения. Интерпретацией решения, формируемого алгоритмами, является хромосома, структура которой идентична структуре списка. Разбиение списка на части производится в результате декодирования. Описываются поисковые процедуры в пространстве решений, механизмы поведения модернизированного роя частиц и генетического поиска. В отличие от канонической парадигмы роевого алгоритма предлагается подход к построению модифицированной парадигмы роя частиц, обеспечивающей возможность одновременного использования хромосом с дискретными целочисленными значениями параметров в генетическом алгоритме и в алгоритме на основе роя частиц. В качестве аналога скорости при перемещении выступает оператор направленной мутации (ОНМ), суть которого заключается в изменения целочисленных значений генов в хромосоме. В отличие от канонической парадигмы роевого алгоритма предлагается подход к построению модифицированной парадигмы роя частиц, обеспечивающей возможность одновременного использования хромосом с дискретными целочисленными значениями параметров в генетическом алгоритме и в алгоритме на основе роя частиц. В качестве аналога скорости при перемещении частиц выступает оператор направленной мутации (ОНМ), суть которого заключается в изменения целочисленных значений генов в хромосоме. Это позволило снизить комбинаторную сложность задачи. Это позволило снизить комбинаторную сложность задачи. 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.

СБИС; размещение стандартных ячеек; одномерная упаковка; генетический поиск; рой частиц; аффинное пространство поиска; гибридизация; оптимизация.

B.K. Lebedev, O.B. Lebedev, A.E. Lebedinsky

**HYBRID BIOINSPIRED ALGORITHM FOR PLACING BASIC STANDARD
LIBRARY ELEMENTS WHEN DESIGNING A TOPOLOGY VLSI**

A hybrid algorithm is proposed for locating basic standard library elements in the design of topology of a semi-custom VLSI using the one-dimensional packaging method based on the integration of swarm and genetic algorithms. In the work, as a solution interpretation - a data structure that carries information about packaging (placement), a sequence of elements (priority list) is used, which determines the order of their packaging. As a decoder, the standard packaging procedure (SPP) is used. The interpretation of the placement problem, which is generated by the swarm

* Работа выполнена при финансовой поддержке РФФИ (проект № 17-07-00997 А).

and genetic algorithms, is an ordered list that includes all cells, with its division into parts. Each part includes vertices corresponding to elements placed in a ruler (block). The number of parts serves as an estimate of the solution. The interpretation of the solution generated by the algorithms is the chromosome, whose structure is identical to the list structure. The list is divided into parts as a result of decoding. We describe the search procedures in the decision space, the mechanisms of behavior of the modernized swarm of particles and genetic search. In contrast to the canonical paradigm of the swarm algorithm, an approach to constructing a modified particle swarm paradigm is proposed, which makes it possible to simultaneously use chromosomes with discrete integer parameter values in a genetic algorithm and in a particle swarm algorithm. The operator of a directed mutation (ODM), the essence of which consists in changing the integer values of genes in the chromosome, acts as an analogue of speed when moving. In contrast to the canonical paradigm of the swarm algorithm, an approach to constructing a modified particle swarm paradigm is proposed, which makes it possible to simultaneously use chromosomes with discrete integer parameter values in a genetic algorithm and in a particle swarm algorithm. As an analogue of speed when moving particles, there is a directed mutation operator (ODM), the essence of which is to change the integer values of genes in the chromosome. This reduced the combinatorial complexity of the problem. This reduced the combinatorial complexity of the problem. For the experiments were used well-known test problems presented in the library of OR-objects - (OR-Library Beasley). The temporal complexity of the algorithm, obtained experimentally, coincides with theoretical studies and for the considered test problems it is $O(n^2)$ - $O(n^3)$. The developed algorithm allowed us to obtain optimal solutions for all tasks of the set.

VLSI; placement of standard cells; one-dimensional packaging; genetic search; particle swarm; affine search space; hybridization; optimization.

Введение. Структура полузаказной СБИС 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 входящих 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 при размещении решение 3-х задач.

Первая задача заключается в разбиении множества ячеек на подмножества. Каждое подмножество ячеек входит в состав одной линейки. Вторая задача связана с определением относительного расположения линеек друг относительно друга. Третья задача заключается в определении относительного расположения ячеек друг относительно друга в пределах каждой линейки. Основными критериями оптимизации являются площадь кристалла и оценка суммарной длины соединений. Если размер линеек (длина) задается априорно, то в этом случае 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.

Задача распределения множества элементов в минимально возможном количестве линеек одинаковой длины фактически является модификацией классической задачи 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 к 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 эвристики.
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 [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 M_a , который легко можно трансформировать в список M_c , и уже на базе M_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 размещения базовых стандартных библиотечных элементов методом одномерной упаковки на основе интеграции модифицированной модели адаптивного поведения роя частиц и генетической эволюции [15]. Предложена композитная архитектура многоагентной системы бионического поиска.

1. Постановка задачи. Проблема одномерной упаковки может быть сформулирована следующим образом. Дано множество элементов $A = \{a_i / i = 1, 2, \dots, n\}$. Вес элементов задается множеством $W = \{w_i / i = 1, 2, \dots, n\}$. Необходимо сформировать F узлов, т.е. множество A разбить на F непустых и непересекающихся подмножеств

A_j . $A = \cup A_j$, $A_j \neq \emptyset$, $(\forall i, j) [A_i \cap A_j = \emptyset]$. 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. Применительно к задаче размещения элементов полузаказной СБИС ячеечного типа необходимо определить взаимное расположение линеек относительно друг друга и элементов в пределах линеек.

В работе в качестве интерпретации решения – структуры данных, несущей информацию об упаковке (размещении), используется последовательность элементов (приоритетный список), задающая порядок их упаковки [1–3]. Приоритетный список – это кодированное решение, в терминах генетического алгоритма – «хромосома». Переход от приоритетного списка к решению производится с помощью декодера. Декодер – оператор, позволяющий перейти от косвенной (числовой) схемы кодирования решения задачи к прямой (графической) схеме, он формирует схему размещения элементов по приоритетному списку. Таким образом, с помощью декодера можно получить координаты элементов, а значит, и графическое представление решения задачи размещения, а также вычислить значение функции пригодности.

Рассмотрим, используемую в качестве декодера стандартную процедуру упаковки (СПУ) [2]. Дан список M элементов для упаковки. Предположим, что элементы для упаковки отсортированы некоторым образом. Первый элемент списка M упаковывается в блок A_1 . 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 $O(n^2)$. В результате выполнения СПУ множество ячеек разбивается на упорядоченное множество линеек, при этом ячейки внутри линеек упорядочены. Очевидно, что существует такая последовательность элементов в списке, при которой решение задачи упаковки (размещения) будет оптимальным. Таким образом, решение задачи размещения сводится к нахождению списка, используемого стандартной процедурой упаковки.

В работе рассматривается гибридный алгоритм, построенный на интеграции роевого и генетического алгоритмов. Построение алгоритма производилось в соответствии с классификацией Ванга [10] на основе низкоуровневой гибридизации 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 роевого и генетического алгоритмов это популяция лучших решений, сформированная в результате эволюционных преобразований на предыдущих итерациях алгоритмов.

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 роевом и генетическом алгоритмах.

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

Интерпретацией решения задачи размещения, формируемого роевым и генетическим алгоритмами, является упорядоченный список M_i , включающий все ячейки, с разбиением его на части. Каждая часть $M_{ij} \in M_i$ включает вершины, соответствующие элементам, помещаемым в линейку (блок) A_j . Число частей служит оценкой решения. Интерпретацией решения, формируемого алгоритмами, является хромосома, структура которой идентична структуре списка. Для получения решения хромосома декодируется с помощью СПУ. Отметим, что разбиение списка на части производится в результате декодирования. Предлагается композитная архитектура многоагентной системы бионического поиска для решения задачи размещения на основе интеграции алгоритма роя частиц и генетической эволюции. Структура генетического алгоритма служит опорной структурой гибридного алгоритма. Механизмы роевого алгоритма используются для модификации популяции решений, формируемых на каждой итерации генетического алгоритма.

Первый и наиболее простой подход к гибридизации заключается в следующем. С начала поиск решения осуществляется генетическим алгоритмом. Затем на основе популяции, полученной на последней итерации генетического поиска, формируется популяция для роевого алгоритма. В формируемую популяцию включаются лучшие, но отличные друг от друга хромосомы. При необходимости полученная популяция доукомплектовывается новыми индивидами. После этого дальнейший поиск решения осуществляется роевым алгоритмом.

При втором подходе метод роя частиц используется в процессе генетического поиска и играет роль аналогичную генетическим операторам. В этом случае на каждой итерации генетического алгоритма синтез новых хромосом с одной стороны осуществляется с помощью кроссинговера и мутации, а с другой стороны, с помощью операторов направленной мутации (ОНЦ) методами роя частиц.

Связующим звеном такого подхода является структура данных, описывающая в виде хромосомы решение задачи [17]. Предлагается подход к построению модифицированной парадигмы роя частиц, обеспечивающей возможность одновременного использования хромосом с целочисленными значениями параметров в генетическом алгоритме и в алгоритме на основе роя частиц.

2. Механизмы генетического поиска. На первом этапе гибридного алгоритма выполняется генетическая эволюция популяции.

Начальная популяция генерируется случайным способом. Хромосома, описывающая решение, имеет следующую структуру: $H_i(t) = \{g_{il}(t) | l = 1, 2, \dots, n\}$. Значение гена $g_{il}(t)$ равно значению соответствующего элемента списка M_i .

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. 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 список первым включается ген хромосомы H_2 , а затем ген хромосомы H_1 .

При третьем способе в 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_m применительно к одной хромосоме имеют линейную зависимость и оценки временной сложности имеют вид: $t_k=O(L_h)$, $t_m=O(L_h)$, 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) вначале 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 выполняются заключительные процедуры алгоритма роя частиц.

3. Механизмы роя частиц. Метод роя частиц (Particle Swarm Optimization, PSO) является методом стохастической оптимизации в чем-то схожим с эволюционными алгоритмами. Этот метод моделирует не эволюцию, а ройное и стайное поведение животных [6]. В отличие от популяционных методов PSO работает с одной статической популяцией, члены которой постепенно улучшаются с появлением информации о пространстве поиска. Данный метод представляет собой вид направленной мутации (*directed mutation*). Решения в PSO мутируют в направлении наилучших найденных решений. Частицы никогда не умирают (т.к. нет селекции).

В процессе поиска методом роя частиц каждая частица перемещается в новую позицию [14]. Новая позиция в канонической парадигме роя частиц определяется как:

$$x_i(t+1) = x_i(t) + v_i(t+1), \quad (1)$$

где $v_i(t+1)$ скорость перемещения частицы из позиции $x_i(t)$ в позицию $x_i(t+1)$. Начальное состояние определяется, как $x_i(0)$, $v_i(0)$ [1].

Введем обозначения:

◆ $x_i(t)$ – текущая позиция частицы, $f_i(t)$ значение целевой функции частицы p_i в позиции $x_i(t)$;

◆ $x_i^*(t)$ – лучшая позиция частицы p_i , которую она посещала с начала первой итерации, а $f_i^*(t)$ – значение целевой функции частицы p_i в этой позиции (лучшее значение с момента старта жизненного цикла частицы p_i);

◆ $x^*(t)$ – позиция частицы с лучшим значением целевой функции $f^*(t)$ среди частиц роя в момент времени t .

Для отдельного измерения j пространства поиска формула (1) примет вид:

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1), \quad (2)$$

где $x_{ij}(t)$ – позиция частицы p_i в измерении j , $v_{ij}(t+1)$ – скорость частицы p_i измерении j .

Тогда скорость частицы p_i на шаге $(t+1)$ в измерении j вычисляется как:

$$v_{ij}(t+1) = w \cdot v_{ij}(t) + k_1 \cdot \text{rnd}(0,1) \cdot (x_{ij}^*(t) - x_{ij}(t)) + k_2 \cdot \text{rnd}(0,1) \cdot (x_j(t) - x_{ij}(t)), \quad (3)$$

где $\text{rnd}(0,1)$ – случайное число на интервале $(0,1)$, (w, k_1, k_2) – некоторые коэффициенты.

Скорость $v_i(t+1)$ рассматривается как средство изменения решения.

Каноническая парадигма роя частиц предусматривает использование вещественных значений параметров в многомерных, вещественных, метрических пространствах. Однако в большинстве генетических алгоритмов гены в хромосомах имеют дискретные, целочисленные значения. В свою очередь хромосомы являются некоторыми интерпретациями решений, которые трансформируются в решения путем декодирования хромосом.

Если в качестве позиции используется хромосома, то число параметров, определяющих положение частицы в пространстве решений (позицию) должно быть равно числу генов в хромосоме. Значение каждого гена откладывается на соответствующей оси пространства решений. В этом случае возникают некоторые требования к структуре хромосомы и значениям генов. В связи с этим актуальной является разработка модернизированной структуры пространства поиска, структуры данных для представления решений и позиций, модернизированных механизмов перемещения частиц в пространстве поиска. В работе предлагается подход к построению модифицированной парадигмы роя частиц, обеспечивающей возможность одновременного использования хромосом с дискретными целочисленными значениями параметров в генетическом алгоритме и в алгоритме на основе роя частиц.

Пусть имеется линейное векторное пространство (ЛВП), элементами которого являются n -мерные точки. Каждым любым двум точкам p и q этого пространства однозначным образом сопоставим единственную упорядоченную пару этих точек, которую в дальнейшем будем называть геометрическим вектором (вектором). $p, q \in V(p, q)$ – геометрический вектор (упорядоченная пара).

Совокупность всех точек ЛВП, пополненная геометрическими векторами, называют точечно-векторным или аффинным пространством. Аффинное пространство является n -мерным, если соответствующее ЛВП так же является n -мерным.

Аффинно-релаксационная модель (АРМ) роя частиц – это граф, вершины которого соответствуют позициям роя частиц, а дуги соответствуют аффинным связям между позициями (точками) в аффинном пространстве. Аффинность – мера близости двух агентов (частиц). На каждой итерации каждый агент p_i переходит в аффинном пространстве в новое состояние (позицию), при котором вес аффинной связи между агентом p_i и базовым (лучшим) агентом p_* уменьшается. Переход агента p_i в новую позицию $x_i(t+1)$ из $x_i(t)$ осуществляется с помощью релаксационной процедуры.

Специальная релаксационная процедура перехода зависит от вида структуры данных (хромосомы): вектор, матрица, дерево и их совокупности, являющейся интерпретацией решений.

Лучшие частицы (и их позиции) с точки зрения целевой функции объявляются «центром притяжения». Векторы перемещения всех частиц в аффинном пространстве устремляются к этим центрам.

Переход возможен с учетом степени близости к одному базовому элементу либо к группе соседних элементов и с учетом вероятности перехода в новое состояние.

Каждую хромосому, описывающую i -е решение популяции, будем обозначать как $H_i(t) = \{g_{il} | l=1, 2, \dots, n\}$. Позиция $x_i(t)$ соответствует решению, задаваемому хромосомой $H_i(t)$, т.е. $x_i(t) = H_i(t) = \{g_{il} | l=1, 2, \dots, n\}$. Число осей в пространстве решений равно числу n генов в хромосоме $H_i(t)$. Точками отсчета на каждой оси l являются дискретные целочисленные значения генов.

В отличие от канонического метода роя частиц, в нашем случае скорость $v_i(t+1)$ не может быть представленной в виде аналитического выражения с целочисленными дискретными значениями переменных. В качестве аналога скорости $v_i(t+1)$ выступает оператор направленной мутации (ОНМ), суть которого заключается в изменения целочисленных значений генов в хромосоме $H_i(t)$. В отличие от канонической парадигмы роевого алгоритма предлагается подход к построению модифицированной парадигмы роя частиц, обеспечивающей возможность одновременного использования хромосом с дискретными целочисленными значениями параметров в генетическом алгоритме и в алгоритме на основе роя частиц. В качестве аналога скорости $v_i(t+1)$ выступает оператор направленной мутации (ОНМ), суть которого заключается в изменения целочисленных значений генов в хромосоме $H_i(t)$. Это позволило снизить комбинаторную сложность задачи.

В работе предлагается подход к построению структур и принципов кодирования гомологичных хромосом (имеющих одинаковый набор генов), обеспечивающих их гомологичность, и возможность одновременного использования в генетическом алгоритме и в алгоритме на основе роя частиц, полученную после применения ОНМ.

По аналогии с каноническим методом роя частиц позицию $x_i^*(t)$ будем называть когнитивным центром притяжения, а позицию $x^*(t)$ – социальным центром притяжения. Когнитивный центр выступает в роли индивидуальной памяти о наиболее оптимальных позициях данной частицы. Благодаря социальному центру частица имеет возможность передвигаться в оптимальные позиции, найденные соседними частицами.

В качестве оценки степени близости между двумя позициями $x_i(t)$ и $x_z(t)$ будем использовать величину аффинной связи d_{iz} (расстояния между хромосомами $H_i(t)$ и $H_z(t)$).

Целью перемещения хромосомы $H_i(t)$ в направлении хромосомы $H_z(t)$ является сокращении расстояния между ними.

Для учета одновременного тяготения частицы p_i к лучшей к позиции $x^*(t)$, среди частиц роя в момент времени t , и к лучшей позиции $x^*_i(t)$ частицы p_i , которую она посещала с начала первой итерации, формируется виртуальный центр (позиция) притяжения $x^c_i(t)$ частицы p_i . Формирование *виртуальной позиции* $x^c_i(t)$ осуществляется путем применения процедуры *виртуального* перемещения из позиции $x^*_i(t)$ в виртуальную позицию $x^c_i(t)$ по направлению к позиции $x^*(t)$. После определения центра притяжения $x^c_i(t)$ частица p_i перемещается в направлении *виртуальной позиции* $x^c_i(t)$ из позиции $x_i(t)$ в позицию $x_i(t+1)$. После перемещения частицы p_i в новую позицию $x_i(t+1)$ виртуальная позиция $x^c_i(t)$ исключается.

Локальная цель перемещения частицы p_i – достижение ею позиции с наилучшим значением целевой функции. Глобальная цель роя частиц – формирование оптимального решения задачи.

Суть *процедуры перемещения*, реализуемой ОНМ в алгоритме роя частиц, заключается в изменении взаимного расположения элементов в списке (генов в хромосоме). Частица p_i перемещается из позиции $H_i(t)$ в направлении позиции в новую позицию $H_i(t+1)$ с новым взаимным расположением элементов в списке.

Введем характеристику, отражающую степень различия между позициями $H_i(t)$ и $H_z(t)$. Для этого произведем сравнение взаимного расположения элементов у всех возможных пар генов в сравниваемых позициях $H_i(t)$ и $H_z(t)$. Пусть $S_{iz}(t)$ – число пар, у которых взаимное расположение элементов в сравниваемых позициях $H_i(t)$ и $H_z(t)$ не совпадает.

Чем больше $d_{iz}(t)$, тем больше различие между $H_i(t)$ и $H_z(t)$, и наоборот, чем меньше $d_{iz}(t)$, тем различие меньше. Тяготение частицы p_i в позиции $H_i(t)$ к лучшей позиции $H_z(t)$, выражается в стремлении уменьшения различия между $H_i(t)$ и $H_z(t)$, то есть уменьшения показателя $d_{iz}(t)$. Будем считать, чем меньше различие между $H_i(t)$ и $H_z(t)$, тем ближе позиция $H_i(t)$, к лучшей позиции $H_z(t)$.

Модификация позиции $H_i(t)$, то есть переход к позиции $H_i(t+1)$, производится путём выборочных групповых парных перестановок соседних элементов в позиции $H_i(t)$. Модификация позиции $H_i(t)$ выполняется за два такта.

На первом такте формируется множество D_1 непересекающихся пар элементов в позиции $H_i(t)$ таких у которых индекс l -нечетное число: $P_1 = \{(g_{il}(t), g_{i,l+1}(t)) | l=1,3,5, \dots\}$. Подсчитывается число пар $d_{iz}(t)$ в множестве D_1 , у которых взаимное расположение элементов пары в позиции $H_i(t)$ не совпадает с взаимным расположением этих элементов в позиции $H_z(t)$. Принимается решение о перестановке элементов каждой такой пары в позиции $H_i(t)$, вычисляются показатели $d_{ik}(t)$ и осуществляется модификация $H_i(t)$.

На втором такте формируется множество D_2 непересекающихся пар элементов в позиции $H_i(t)$ у которых индекс l -четное число: $D_2 = \{(g_{il}(t), g_{i,l+1}(t)) | l=2,4,6, \dots\}$. Подсчитывается число пар $d_{iz}(t)$ множества D_2 , у которых взаимное расположение элементов пары в позиции $H_i(t)$ не совпадает с взаимным расположением этих элементов в позиции $H_z(t)$. Принимается решение о перестановке элементов каждой такой пары в позиции $H_i(t)$, вычисляются показатели $S_{ik}(t+1)$ и формируется позиция $H_i(t+1)$.

Перестановка выполняется с вероятностью $P = a \cdot d_{iz}(t) / n$, где n – число пар множества D .

Отметим, что изменение взаимного расположения пары соседних элементов в позиции не приводит к изменению относительного расположения каждого элемента пары с остальными элементами позиции, а также к изменению взаимного расположения остальных элементов друг относительно друга.

Пример работы процедуры перемещения.

Пусть позиции $H_i(t)$ и $H_z(t)$ имеют вид:

$$H_i(t)=\{1,3,2,10,8\}, H_z(t)=\{1,10,2,3,8\}, S_{iz}(t)=2.$$

На первом такте шага t формируется множество пар $D_1=(1,3), (2,10)$. Взаимное расположение элементов пары $(1,3)$ в $H_i(t)$ и $H_z(t)$ совпадают, пары $(2,10)$ – нет. Отсюда $d_{iz}(t_1)=1$. Переставляются в $H_i(t)$ местами элементы пары $(2,10)$. $H_i(t_1)=\{1,3,10,2,8\}$.

На втором такте формируется множество пар $D_2=(3,10), (2,8)$. Взаимное расположение элементов пары $(2,8)$ в $H_i(t_1)$ и $H_z(t)$ совпадают, пары $(3,10)$ – нет. Отсюда $d_{iz}(t_2)=1$. Переставляются в $H_i(t_1)$ местами элементы пары $(3,10)$. $H_i(t+1)=\{1,10,3,2,8\}$. $d_{iz}(t+1)=1$.

4. Экспериментальные исследования. Целью исследования разработанного алгоритма было определение качества 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 феромона $\tau_k(l)$, откладываемого агентами на каждой итерации. Коэффициент обновления $\rho=0,95$.

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).

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 [17–20]. Для решения каждой из тестовых задач разработанная программа запускалась 10 раз. Каждый запуск являлся независимым. В табл. 1 в столбце «Задача» приведено кодовое название контрольной (тестовой) задачи, «GO» – количество блоков в глобально оптимальном решении, «НВ» – количество блоков в лучшем решении, полученном при помощи гибридного алгоритма упаковки, «O» – отклонение полученного решения от глобально оптимального решения.

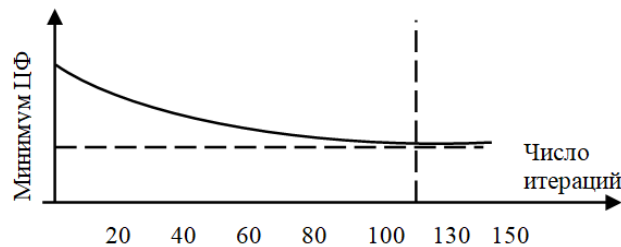


Рис. 2. Зависимость качества гибридного алгоритма от числа итераций

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 табл. 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 $O(n^2) - O(n^3)$.

Заключение. Предложены новые механизмы решения задачи размещения базовых стандартных библиотечных элементов методом одномерной упаковки на основе интеграции модифицированных моделей адаптивного поведения роя частиц и генетической эволюции, использующие математические методы, в которых заложены принципы природных механизмов принятия решений. Интегрируемые алгоритмы базируются на использовании коллективной эволюционной памяти, имеющую общие принципы ее организации. В связи с этим интеграция метаэвристик алгоритма роя частиц 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 роевом и генетическом алгоритмах. В отличие от канонической парадигмы роевого алгоритма предлагается подход к построению модифицированной парадигмы

роя частиц, обеспечивающей возможность одновременного использования хромосом с дискретными целочисленными значениями параметров в генетическом алгоритме и в алгоритме на основе роя частиц. В качестве аналога скорости $v_i(t+1)$ выступает оператор направленной мутации (ОНМ), суть которого заключается в изменении целочисленных значений генов в хромосоме $H_i(t)$. Это позволило снизить комбинаторную сложность задачи.

Временная сложность алгоритма (ВСА), полученная экспериментальным путем, практически совпадает с теоретическими исследованиями и для рассмотренных тестовых задач составляет (ВСА $\approx O(n^2) - O(n^3)$).

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 была проведена серия опытов-экспериментов. Разработанный гибридный алгоритм позволил получить оптимальные решения для всех задач набора.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Месягутов М.А., Мухачева Э.А., Белов Г.Н., Шайтхауэр Г.* Упаковка одномерных контейнеров с продолженным выбором идентичных предметов: точный метод поиска оптимального решения // Автоматика и телемеханика. – 2011. – № 1. – С. 154-173.
2. *Потарусов Р.В., Курейчик В.М.* Проблема одномерной упаковки элементов // Известия 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-heuristics // Proceeding of the Genetic and Evolutionary Computation Conference, GECCO 2003, Chicargo, Illinois, USA. – 2003. – P. 1295-1306.
4. *Gupta J.N. and Ho J.C.* A New Heuristic Algorithm for the One-dimensional Bin-packing Problem // Production Planning & Control. – 1999. – No. 10. – P. 598-603.
5. *Levine J. and Ducatelle F.* 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. – P. 389-401.
6. *Fleszar K. and Hindi K.S.* New Heuristics for One-dimensional Bin-packing // Computers & Operations Research. – 2002. – No. 29. – P. 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. – 2004. – No. 10. – P. 205-229.
8. *Курейчик В.М., Лебедев Б.К., Лебедев О.Б.* Поисковая адаптация: теория и практика. – М.: Физматлит, 2006. – 288 с.
9. *Engelbrecht A.P.* Fundamentals of Computational Swarm Intelligence. John Wiley & Sons, Chichester, UK, 2005. – 235 p.
10. *Карпенко А.П.* Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой: учеб. пособие. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2014. – 448 с.
11. *Курейчик В.М., Лебедев Б.К., Лебедев О.Б.* Решение задачи размещения на основе эволюционного моделирования // Известия РАН. Теория и системы управления. – 2007. – № 4. – С. 78-90.
12. *Clerc M.* Particle Swarm Optimization. – ISTE, London, UK, 2006. – 248 p.
13. *Лебедев О.Б.* Модели адаптивного поведения муравьиной колонии в задачах проектирования. – Таганрог: Изд-во ЮФУ, 2013. – 199 с.
14. *Lebedev B.K., Lebedev O.B., Lebedeva E.O.* One-Dimensional swarm algorithm packaging // International Conference Information Technologies in Business and Industry 2018. Journal of Physics: Conference Series. Vol. 1015. Enterprise Information systems. – 2018.
15. *Лебедев Б.К., Лебедев О.Б., Лебедев В.Б.* Гибридизация роевого интеллекта и генетической эволюции на примере размещения // Программные продукты, системы и алгоритмы. – <http://swsys-web.ru/hybridization-of-swarm-intelligence-and-genetic-evolution.html>. DOI: 10.15827/2311-6749.25.280.

16. Valerio de Carvalho J.M. Exact Solution of Bin-packing Problems Using Column Generation and Branch-and-bound // *Annals of Operations Research*. – 1999. – No. 86. – P. 629-659.
17. Агасиев Т.А., Карпенко А.П. Современные техники глобальной оптимизации // *Информационные технологии*. – 2018. – № 6. – С. 370-386.
18. Норенков И.П., Уваров М.Ю. Поддержка принятия решений на основе паттернов проектирования // *Наука и образование: Научное издание*. – 2011. – № 9. – С. 25-32.
19. Божко А.Н. Методы структурного анализа сложных изделий в интегрированных CFD/CAM-системах // *Информационные технологии*. – 2018. – № 8. – С. 499-506.
20. Столяров А.И., Донецкая Ю.В., Гатчин Ю.А. Особенности САПР при проектировании комплекса // *Вестник Иркутского государственного технического университета*. – 2018. – № 7 (138). – С. 88-95.

REFERENCES

1. Mesyagutov M.A., Mukhacheva E.A., Belov G.N., Shaytkhauer G. Upakovka odnomernykh konteynerov s prodolzhenym vyborom identichnykh predmetov: tochnyy metod poiska optimal'nogo resheniya [Packaging of one-dimensional containers with continued selection of identical items: the exact method of finding the optimal solution], *Avtomatika i telemekhanika* [Automation and Remote Control], 2011, No. 1, pp. 154-173.
2. Potarusov R.V., Kureychik V.M. Problema odnomernoy upakovki elementov [The problem of one-dimensional packaging of elements], *Izvestiya TRTU* [Izvestiya TSURE], 2006, No. 8, pp. 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-heuristics, *Proceeding of the Genetic and Evolutionary Computation Conference, GECCO 2003, Chicargo, Illinois, USA*, 2003, pp. 1295-1306.
4. Gupta J.N. and Ho J.C. A New Heuristic Algorithm for the One-dimensional Bin-packing Problem, *Production Planning & Control*, 1999, No. 10, pp. 598-603.
5. Levine J. and Ducatelle F. 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, pp. 389-401.
6. Fleszar K. and Hindi K.S. New Heuristics for One-dimensional Bin-packing, *Computers & Operations Research*, 2002, No. 29, pp. 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*, 2004, No. 10, pp. 205-229.
8. Kureychik V.M., Lebedev B.K., Lebedev O.B. Poiskovaya adaptatsiya: teoriya i praktika [Search adaptation: theory and practice]. Moscow: Fizmatlit, 2006, 288 p.
9. Engelbrecht A.P. Fundamentals of Computational Swarm Intelligence. John Wiley & Sons, Chichester, UK, 2005, 235 p.
10. Karpenko A.P. Sovremennyye algoritmy poiskovoy optimizatsii. Algoritmy, vdokhnovlennyye prirodoy: ucheb. posobie [Modern search engine optimization algorithms. Algorithms inspired by nature: textbook]. Moscow: Izd-vo MGTU im. N.E. Bauman, 2014, 448 p.
11. Kureychik V.M., Lebedev B.K., Lebedev O.B. Reshenie zadachi razmeshcheniya na osnove evolyutsionnogo modelirovaniya [The solution of the problem of placement based on evolutionary modeling], *Izvestiya RAN. Teoriya i sistemy upravleniya* [News of the RAS. Theory and Management Systems], 2007, No. 4, pp. 78-90.
12. Clerc M. Particle Swarm Optimization. ISTE, London, UK, 2006, 248 p.
13. Lebedev O.B. Modeli adaptivnogo povedeniya murav'inoi kolonii v zadachakh proektirovaniya. Taganrog: Izd-vo YuFU, 2013, 199 p.
14. Lebedev B.K., Lebedev O.B., Lebedeva E.O. One-Dimensional swarm algorithm packaging, *International Conference Information Technologies in Business and Industry 2018. Journal of Physics: Conference Series. Vol. 1015. Enterprise Information systems*, 2018.
15. Lebedev B.K., Lebedev O.B., Lebedev V.B. Gibrizatsiya roevogo intellekta i geneticheskoy evolyutsii na primere razmeshcheniya [Hybridization of swarm intelligence and genetic evolution on the example of placement], *Programmye produkty, sistemy i algoritmy* [Software products, systems and algorithms]. Available at: <http://swsys-web.ru/hybridization-of-swarm-intelligence-and-genetic-evolution.html>. DOI: 10.15827/2311-6749.25.280.
16. Valerio de Carvalho J.M. Exact Solution of Bin-packing Problems Using Column Generation and Branch-and-bound, *Annals of Operations Research*, 1999, No. 86, pp. 629-659.

17. Agasiev T.A., Karpenko A.P. Sovremennye tekhniki global'noy optimizatsii [Modern technology of global optimization], *Informatsionnye tekhnologii* [Information technologies], 2018, No. 6, pp. 370-386.
18. Norenkov I.P., Uvarov M.Yu. Podderzhka prinyatiya resheniy na osnove patternov proektirovaniya [Decision support based on design patterns], *Nauka i obrazovanie: Nauchnoe izdanie* [Science and Education: Scientific publication], 2011, No. 9, pp. 25-32.
19. Bozhko A.N. Metody strukturnogo analiza slozhnykh izdeliy v integrirovannykh CFD/CAM-sistemakh [Structural analysis methods for complex products in integrated CAD/CAM-systems], *Informatsionnye tekhnologii* [Information Technologies], 2018, No. 8, pp. 499-506.
20. Stolyarov A.I., Donetskaya Yu.V., Gatchin Yu.A. Osobennosti SAPR pri proektirovanii kompleksa [Features of CAD in the design of the complex], *Vestnik Irkutskogo gosudarstvennogo tekhnicheskogo universiteta* [Bulletin of the Irkutsk State Technical University], 2018, No. 7 (138), pp. 88-95.

Статью рекомендовал к опубликованию д.т.н. профессор А.Г. Коробейников.

Лебедев Борис Константинович – Южный федеральный университет; e-mail: lebedev.b.k@gmail.com; 347928, г. Таганрог, пер. Некрасовский, 44; тел.: 89282897933; кафедра систем автоматизированного проектирования; профессор.

Лебедев Олег Борисович – e-mail: lebedev.ob@mail.ru; тел.: 89085135512; кафедра систем автоматизированного проектирования; доцент.

Лебединский Александр Евгеньевич – e-mail: lebedev.ob@mail.ru; тел.: 89034359392; кафедра систем автоматизированного проектирования; аспирант.

Lebedev Boris Konstantinovich – Southern Federal University; e-mail: lebedev.b.k@gmail.com; 44, Nekrasovskiy, Taganrog, 347928, Russia; phone: +7928-2897933; the department of computer aided design; professor.

Lebedev Oleg Borisovich – e-mail: lebedev.ob@mail.ru; phone: +79085135512; the department of computer aided design; associate professor.

Lebedinsky Alexander Evgenievich – e-mail: lbedevakate@mail.ru; phone: +79289591426; the department of computer aided design; graduate student.

УДК 004.891.3

DOI 10.23683/2311-3103-2019-3-110-121

И.А. Тарасова

ЗАДАНИЕ ФУНКЦИЙ ПРИНАДЛЕЖНОСТИ ТЕРМОВ ЛИНГВИСТИЧЕСКИХ ПЕРЕМЕННЫХ В ЗАДАЧЕ ОПРЕДЕЛЕНИЯ ДОЗИРОВОК МЕДИКАМЕНТОВ ПРИ ЛЕЧЕНИИ ПРЕЭКЛАМПСИИ БЕРЕМЕННЫХ ЖЕНЩИН

Проблема преэклампсии беременных женщин является одной из актуальных в современном акушерстве, так как это заболевание является наиболее распространенным и тяжелым осложнением беременности, а проблема лечения тяжелых форм преэклампсии – одна из самых трудных в акушерской анестезиологии и реаниматологии. Высокая частота материнской и перинатальной заболеваемости и смертности основана на отсутствии точных знаний о патогенезе заболевания, который зависит от множества предрасполагающих факторов, достоверных диагностических критериев, что ведет к неадекватной терапии и различным осложнениям, зависящим от своевременности и метода родоразрешения, объема анестезиологической и реанимационной помощи. Учитывая высокую распространенность преэклампсии в России, а также отсутствие тенденции к снижению, целью работы является повышение эффективности проведения лечебной терапии при гипертензивных осложнениях беременности. Анализ процесса диагностики и введения медикаментов при лечении преэклампсии беременных женщин, который характеризуется нели-