

УДК 004.62

**В.Б. Лебедев****ФОРМИРОВАНИЕ КОНФИГУРАЦИИ РАСПРЕДЕЛЕННОЙ БАЗЫ ДАННЫХ**

*Рассматривается проблема формирования распределенной базы данных на основе совокупности локальных баз данных. Решаются задачи выбора конфигурации и определения состава библиотечных наборов каждой локальной базы данных. В связи с этим возникает задача покрытия контента распределенного хранилища данных минимальным набором локальных баз данных. Рассмотренная задача эквивалентна задаче о покрытии множествами и является NP-полной. Для компактного представления решения задачи покрытия используется матрица граничных требований. Излагается метод решения задачи покрытия на основе роевого интеллекта. Используется компактное символическое представление решения задачи покрытия. Это позволило организовать пространство решений, использующее неупорядоченные лингвистические шкалы, в рамках которого организован поисковый процесс, базирующийся на моделировании адаптивного поведения роя частиц. Предлагается подход к построению гибридной структуры поиска решения, основанной на сочетании генетического поиска с методами роевого интеллекта. Разработанный алгоритм покрытия имеют универсальный характер и может быть использован для решения широкого круга задач линейного целочисленного программирования. По сравнению с существующими алгоритмами достигнуто улучшение результатов.*

*Распределенная база данных; локальная база данных; хранилище данных; покрытие; адаптивное поведение; рой частиц; оптимизация.*

**V.B. Lebedev****CREATING A CONFIGURATION DISTRIBUTED DATABASES**

*In this paper we present the problem of the formation of a distributed database based on the set of local databases. Solved the problem of choosing the configuration and determine the composition of library collections each local database In this regard, there is the task of covering the content distributed data storage minimum set of local databases. The considered problem is equivalent to the set covering and is NP-complete. For a compact representation of the solution of the covering problem using a matrix boundary requirements. In this paper we present a method of solving the problem of coverage based on swarm intelligence. Used compact symbolic representation of the solution of the covering problem. It is possible to organize the space of solutions using unordered linguistic scale in which organized search process based on simulation of adaptive behavior of particle swarm. The paper proposes an approach to constructing a hybrid structure of finding a solution based on a combination of genetic search methods swarm intelligence. The developed algorithm cover are universal and can be used for a wide range of linear integer programming. Compared with existing algorithms to improve the results achieved.*

*Distributed database; the local database; data warehouse; coating; adaptive behavior; particle swarm; optimization.*

**Введение.** В настоящее время в связи с успешным развитием телекоммуникаций и ростом корпоративных связей между предприятиями актуальна проблема создания распределенных баз данных. Распределенная база данных является совокупностью локальных баз данных. Одна из первых и важнейших задач, решаемых на этапе разработки распределенных баз данных, – это выбор конфигурации и определение состава библиотечных наборов каждой локальной базы данных. Обычно на основе анализа производственной деятельности определяется состав библиотечных наборов в каждой локальной базе данных. С другой стороны, при создании распределенного хранилища данных в зависимости от его назначения

концептуально решается задача о составе и наполнении. Библиотечные наборы локальных баз данных часто пересекаются. В связи с этим возникает задача покрытия контента распределенного хранилища данных минимальным набором локальных баз данных. Рассмотренная задача эквивалентна задаче о покрытии множествами и является *NP*-полной. Покрытие во многом определяет качество топологии СБИС. В течение последних лет были предложены различные подходы к решению проблемы покрытия. Главным образом это алгоритмы, основанные на эвристических, обеспечивающих получение приемлемого результата за полиномиальное время [1]. Тем не менее возросшие сложности решаемых задач и требования к качеству решения делают актуальной разработку новых более эффективных методов. Приведенный в [1, 2, 3] обзор, сравнение и анализ разработанных алгоритмов покрытия (последовательных, итерационных и т.д.) показывает, что для создания эффективного алгоритма покрытия, отвечающего современным требованиям, необходимы новые технологии и подходы. В последние годы интенсивно разрабатывается научное направление, объединяющее математические методы, в которых заложены принципы природных механизмов принятия решений [4, 5]. Одним из новых направлений таких методов являются мультиагентные методы интеллектуальной оптимизации, базирующиеся на моделировании коллективного интеллекта [6, 7, 8]. Оптимизация с использованием роя частиц (Particle Swarm Optimization, PSO) – это метод поиска, который базируется на понятии популяции, и моделирует поведение птиц в стае и косяков рыб [9, 10]. Рой частиц рассматривается как многоагентная система, в которой каждый агент (частица) функционирует автономно по очень простым правилам. В таких случаях говорят о роевом интеллекте (Swarm intelligence).

В работе излагается метод решения задачи покрытия на основе роевого интеллекта [10]. Используется компактное символическое представление решения задачи покрытия. Это позволило организовать пространство решений, использующее неупорядоченные лингвистические шкалы, в рамках которого организован поисковый процесс, базирующийся на моделировании адаптивного поведения роя частиц.

**Основные положения.** Исходными данными для решения задачи покрытия являются: состав проектируемого распределенного хранилища данных (совокупность библиотечных наборов) и составы библиотечных наборов в локальных типовых ячейках применяемого библиотечного набора. Необходимо найти такое распределение логических функций покрываемой схемы по отдельным ячейкам, при котором достигается экстремум целевой функции. Известные в литературе алгоритмы покрытия функциональной схемы ячейками из заданного набора оптимизируют следующие показатели: суммарную стоимость ячеек, покрывающих схему; общее число ячеек, необходимое для реализации схемы; число типов используемых ячеек; число межъячеечных связей, общее число элементов, входящих в покрывающий набор ячеек [1, 3, 6].

Пусть задана совокупность библиотечных наборов распределенного хранилища данных (РХД) состоящая из элементов  $s_i$  и множество  $E = \{e_i | i=1, 2, \dots, n\}$  типов библиотечных наборов. Количественный состав по типам библиотечных наборов опишем вектором  $B = \{b_i | i=1, 2, \dots, n\}$ , где  $b_i$  – число библиотечных наборов (БН) типа  $e_i$ , входящих в состав распределенного хранилища данных. Кроме того, задан набор покрывающих локальных баз данных (ЛБД)  $H = \{h_j | j=1, 2, \dots, m\}$ . Каждая локальная база данных имеет свой набор БН из  $E$ . Количественный состав ЛБД выразим с помощью матрицы  $A = \|a_{ij}\|_{n \times m}$ , где  $a_{ij}$  – число БН типа  $e_i$  в ЛБД типа  $h_j$ . Вектор  $C = \{c_j | j=1, 2, \dots, m\}$  конкретизирует для каждой ЛБД  $h_j$  её стоимость  $c_j$ . Схема считается покрытой ЛБД из набора  $H$ , если каждый БН функцио-

нальной схемы реализуется БН из состава выбранных ЛБД. Построим математическую модель. Введём целочисленную переменную  $x_j$ , определяющую число ЛБД типа  $h_j$ , входящих в покрывающий набор. Тогда задача покрытия формулируется следующим образом:

минимизировать

$$F = \sum_{j=1}^m x_j \cdot c_j, \quad (1)$$

при ограничениях  $F$

$$\begin{aligned} \sum_{j=1}^m a_{ij} \cdot x_j &\geq b_i, i=1,2,\dots,n; \\ x_j &\geq 0, j=1,2,\dots,m; \\ a_{ij} &\geq 0. \end{aligned}$$

Решение задачи представляет набор величин  $x_j$ ,  $j=1,2,\dots,m$ , при которых функция  $F$  (суммарная стоимость ячеек покрывающего набора) имеет минимальное значение. Если в качестве показателя  $c_j$  принять общее число БН, присутствующих в составе ЛБД  $t_j$ , т.е.

$$c_j = \sum_{i=1}^n a_{ij},$$

то  $F$  определяет общее число элементов покрывающего набора ячеек.

**Формирование пространства решений задачи покрытия.** Для удобства изложения будем осуществлять процесс формирования пространства решений на примере. Пусть покрываемое РХД включает БН трех типов:  $E=\{e_1, e_2, e_3\}$ . Количественный состав РХД задается вектором  $B=\{10, 30, 20\}$ . Следовательно, имеем 10 БН первого типа, 30 – второго и 20 – третьего. Набор покрывающих ЛБД состоит из пяти типов:  $H=\{h_1, h_2, h_3, h_4, h_5\}$ . Матрица  $A$ , описывающая количественный состав ячеек, представлена на рис. 1.

| Типы<br>БН | Типы ЛБД |       |       |       |       |
|------------|----------|-------|-------|-------|-------|
|            | $h_1$    | $h_2$ | $h_3$ | $h_4$ | $h_5$ |
| $e_1$      | 2        | 3     | 1     | 2     | 2     |
| $e_2$      | 2        | 2     | 3     | 2     | 1     |
| $e_3$      | 3        | 1     | 1     | 2     | 2     |

Рис. 1. Матрица  $A$ , описывающая количественный состав ячеек

Введём матрицу  $P$ , которая отражает граничные требования по количественному составу БН, покрываемых ЛБД каждого типа,  $P = \|p_{ij}\|_{n \times m}$ , где  $p_{ij}$  – минимальное число БН типа  $e_i$ , которое обязательно должно быть покрыто ЛБД типа  $h_j$ ,  $p_{ij} \geq 0$ ,  $p_{ij}$  – целое число. При этом для реализации полного покрытия всех БН в соответствии с требованиями матрицы  $P$  необходимо выполнение ограничений:

$$\sum_{j=1}^m p_{ij} = b_i, i=1,2,\dots,n. \quad (2)$$

Для вышеприведённого примера один из возможных вариантов матрицы  $P$  имеет вид, представленный на рис. 2.

Матрица  $P$  однозначно соответствует покрывающему набору ячеек, который формируется следующим образом. Сначала строится матрица  $D = \lceil \lceil d_{ij} \rceil \rceil_{n \times m}$ . Элемент матрицы  $d_{ij}$  – целое число, которое определяется как большее целое от  $p_{ij}/a_{ij}$  и фактически равно *минимальному* числу ЛБД типа  $h_j$ , необходимых для покрытия  $p_{ij}$  БН типа  $e_i$ . Затем в пределах каждого  $j$ -го столбца матрицы  $D$  находится максимальное число  $d_{jmax}$  ( $\forall i$ ) [ $d_{jmax} \geq d_{ij}$ ]. Оно является минимальным числом  $x_j = d_{jmax}$  ЛБД типа  $h_j$ , гарантированно обеспечивающих покрытие  $p_{1j}$  БН типа  $e_1$ ,  $p_{2j}$  БН типа  $e_2$ , ...,  $p_{nj}$  БН типа  $e_n$ , в соответствии с требованиями матрицы  $P$ . Причём кроме БН типа  $e_i$ , для которых  $d_{ij} = d_{jmax}$ , остальные типы элементов могут быть покрыты с избытком.

Для рассматриваемого примера матрица  $D$  и покрывающий набор  $X$  имеют вид, представленный на рис. 3.

Общее число ЛБД  $N_j$  в покрывающем наборе и общее число  $N_i$  БН типа  $e_i$ , входящее в состав покрывающего набора ЛБД определяются по формулам:  $N_j = \sum x_j$ ;  $N_i = \sum x_j \cdot a_{ij}$ .

Для нашего примера:  $N_j = \sum x_j = 25$ ;  $N_1 = 2 \cdot 2 + 6 \cdot 3 + 5 \cdot 1 + 3 \cdot 2 + 9 \cdot 2 = 51$ ;  $N_2 = 2 \cdot 2 + 6 \cdot 2 + 5 \cdot 3 + 3 \cdot 2 + 9 \cdot 1 = 46$ ;  $N_3 = 2 \cdot 3 + 6 \cdot 1 + 5 \cdot 1 + 3 \cdot 2 + 9 \cdot 2 = 41$ . Общее число БН, образующих покрывающий набор,  $N_s = N_1 + N_2 + N_3 = 138$ . Итак, матрице  $P$ , количество БН которой удовлетворяют ограничениям (2), соответствует одно решение. Различные решения получаются путём комбинирования значениями  $p_{ij}$  матрицы  $P$ , удовлетворяющими ограничениям (2).

$$P = \begin{array}{c|ccccc|c} \begin{array}{c} \text{Типы} \\ \text{БН} \end{array} & \begin{array}{ccccc} \text{Типы ЛБД} \\ \hline h_1 & h_2 & h_3 & h_4 & h_5 \end{array} & \begin{array}{c} b_i = \sum p_{ij} \end{array} \\ \hline e_1 & 2 & 3 & 3 & 1 & 1 & b_1 = 10 \\ e_2 & 4 & 4 & 8 & 5 & 9 & b_2 = 30 \\ e_3 & 2 & 6 & 5 & 1 & 6 & b_3 = 20 \end{array}$$

Рис. 2. Матрица  $P$ , соответствующая покрывающему набору ЛБД

$$D = \begin{array}{c|ccccc} \begin{array}{c} \text{Типы} \\ \text{БН} \end{array} & \begin{array}{ccccc} \text{Типы ЛБД} \\ \hline h_1 & h_2 & h_3 & h_4 & h_5 \end{array} \\ \hline e_1 & 1 & 1 & 3 & 1 & 1 \\ e_2 & 2 & 2 & 3 & 3 & 9 \\ e_3 & 1 & 6 & 5 & 1 & 3 \end{array}$$

$$X = \begin{array}{c|ccccc} \sum x_j & x_1 & x_2 & x_3 & x_4 & x_5 \\ \hline 25 & 2 & 6 & 5 & 3 & 9 \end{array}$$

Рис. 3. Матрица  $D$  и покрывающий набор  $X$

Таким образом, матрица  $P$  является символьным представлением решения задачи покрытия множествами. В работе множество решений представляется множеством матриц  $\Pi = \{P_k | k=1, 2, \dots, m\}$ . Поиск решения сводится к поиску такой матрицы  $P_k$ , т.е. к поиску совокупности таких значений элементов матрицы  $P_k$ , которые оптимизируют показатель качества (критерий).

**Механизмы покрытия на основе роевого интеллекта.** В методе оптимизации роем частиц агентами являются частицы в пространстве параметров задачи оптимизации. Основу поведения роя частиц составляет самоорганизация, обеспечивающая достижение общих целей роя на основе низкоуровневого взаимодействия. Каждая частица связана со всем роем, может взаимодействовать со всем роем и она тяготеет к лучшему решению роя. Процесс поиска решений итерационный. На каждой итерации каждая частица перемещается в новую позицию. В каждый момент времени (на каждой итерации) частицы имеют в этом пространстве некоторое положение и вектор скорости. Для каждого положения частицы вычисляется соответствующее значение целевой функции. При определении следующего положения частицы учитывается информация о наилучшей частице из числа «соседей» данной частицы, а также информация о данной частице на той итерации, когда этой частице соответствовало наилучшее значение целевой функции, и на этой основе по определенным правилам частица меняет свое положение и скорость в пространстве поиска [10]. В работе позиция  $x_k(t)$   $k$ -й частицы в момент времени  $t$  соответствует решению, задаваемому матрицей  $P_k(t)$ . Перемещение  $k$ -й частицы в дискретные моменты времени соответствует изменению матрицы  $P_k(t)$ .

В каноническом методе роя частиц [10], использующем метрические шкалы, новая позиция частицы  $k$  определяется как

$$x_k(t+1) = x_k(t) + v_k(t+1),$$

где  $v_k(t+1)$  – скорость перемещения частицы  $k$  из позиции  $x_k(t)$  в позицию  $x_k(t+1)$ . Начальное состояние определяется как  $x_k(0)$ ,  $v_k(0)$ . Приведенная формула представлена в векторной форме. Для отдельного измерения  $i$  пространства поиска формула примет вид

$$x_{ki}(t+1) = x_{ki}(t) + v_{ki}(t+1), \quad (3)$$

где  $x_{ki}(t)$  – позиция частицы  $k$  в измерении  $i$ ,  $v_{ki}(t+1)$  – скорость частицы  $k$  в измерении  $i$ .

Введем обозначения:

- ◆  $f_k(t)$  – текущее значение целевой функции частицы  $k$  в позиции  $x_k(t)$  (решение  $P_k(t)$ );
- ◆  $f^*(t)$  – лучшее значение целевой функции среди частиц роя в момент времени  $t$ , а  $x^*(t)$  – позиция с этим значением (решение  $P(t)$ );
- ◆  $x_k^*(t)$  – лучшая позиция частицы  $k$ , которую она посещала с начала первой итерации (решение  $P_k^*(t)$ ), а  $f_k^*(t)$  – значение целевой функции в этой позиции лучшее значение частицы  $k$ .

Лучшие частицы, с точки зрения целевой функции, объявляются «центром притяжения». Векторы скоростей всех частиц устремляются к этим центрам.

В работе в отличие от стандартной парадигмы метода роя частиц вместо метрической (числовой) шкалы используется *неупорядоченная лингвистическая шкала*.

В нашем случае позиция  $x_k(t)$  соответствует решению, задаваемому матрицей  $P_k(t)$ . Число  $n$  осей в пространстве решений равно числу строк  $P_{ki}(t)$  в матрице  $P_k(t)$ . Точками отсчета на каждой оси  $i$  являются значения лингвистической переменной  $x_{ki}(t)$  неупорядоченные друг относительно друга. Значением  $x_{ki}(t)$  является набор значений элементов  $p_{kij}(t)$  в  $i$ -й строке  $P_{ki}(t)$  матрицы  $P_k(t)$  для которых выполняется ограничение (2). Позиция  $x_k(t)$ , т.е. матрица  $P_k(t)$  является решением, а скорость  $v_k(t+1)$  рассматривается как средство изменения решения. В отличие от канонического метода роя частиц, в нашем случае скорость  $v_k(t+1)$  не может быть представленной в виде аналитического выражения. В качестве скорости  $v_k(t+1)$  выступает *процедура перемещения*, суть которой заключается в изменении значения элементов в строках  $P_{ki}(t)$  матрицы  $P_k(t)$  с соблюдением ограничений (2). Частица  $k$  перемещается в новую позицию с новыми значениями лингвистических переменных  $x_{ki}(t)$ .

Пусть «центром притяжения» частицы  $k$ , размещенной в позиции  $x_k(t)$  (решение  $P_k(t)$ ), является позиция  $x_c(t)$  (решение  $P_c(t)$ ).

Рассмотрим работу процедуры перемещения, приводящую к модификации позиции  $x_{ki}(t)$  частицы  $k$  в измерении  $i$ . Введем характеристику, отражающую степень различия между строками  $P_{ki}(t)$  и  $P_{ci}(t)$ . Для этого для каждой пары элементов  $p_{kij}(t)$  и  $p_{cij}(t)$  рассчитывается величина  $\delta_{kcij}(t) = p_{kij}(t) - p_{cij}(t)$ . Подсчитывается число  $\alpha_{kci}(t)$  положительных и число  $\beta_{kci}(t)$  отрицательных значений величины  $\delta_{kcij}(t)$ .

Если  $\alpha_{kci}(t) < \beta_{kci}(t)$ , то значения всех  $\alpha_{kci}(t)$  элементов  $p_{kij}(t)$ , для которых величина  $\delta_{kcij}(t)$  положительна, уменьшаются на единицу, а значения случайно выбранных  $\alpha_{kci}(t)$  элементов  $p_{kij}(t)$ , для которых  $\delta_{kcij}(t)$  отрицательна, увеличиваются на единицу.

Если  $\alpha_{kci}(t) > \beta_{kci}(t)$ , то для значения всех  $\beta_{kci}(t)$  элементов  $p_{kij}(t)$ , для которых величина  $\delta_{kcij}(t)$  отрицательна, увеличиваются на единицу, а значения случайно выбранных  $\beta_{kci}(t)$  элементов  $p_{kij}(t)$ , для которых  $\delta_{kcij}(t)$  положительна, уменьшаются на единицу. При таком подходе суммарное значение элементов строки  $P_{ki}(t+1)$  не изменится. Локальная цель перемещения частицы  $k$  – достижение ею наилучшей позиции. Глобальная цель роя частиц – формирование решения задачи покрытия с наилучшим значением целевой функции.

**Пример** работы процедуры перемещения. Пусть  $P_{ki}(t) = \langle 2, 4, 3, 5, 3, 6, 3 \rangle$ ,  $P_{ci}(t) = \langle 1, 6, 3, 2, 3, 7, 4 \rangle$ . Рассчитываем значения  $\delta_{kcij}(t) = p_{kij}(t) - p_{cij}(t)$ .  $\delta_{kci}(t) = \langle +1, -2, 0, +3, 0, -1, -1 \rangle$ . Число  $\alpha_{kci}(t) = 2$ , а число  $\beta_{kci}(t) = 3$ . Имеем  $\alpha_{kci}(t) < \beta_{kci}(t)$ . Тогда в соответствии с вышеприведенным правилом значения 1-го и 4-го элементов строки  $P_{ki}(t)$  уменьшаем на единицу, а значения случайно выбранных 6-го и 7-го элементов строки  $P_{ki}(t)$  увеличиваем на единицу. В результате получаем  $P_{ki}(t+1) = \langle 1, 4, 3, 4, 3, 7, 4 \rangle$ .

По аналогии с каноническим методом роя частиц позицию  $x_k^*(t)$  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.

Для учета одновременного тяготения частицы  $k$  к позициям  $x^*(t)$  и  $x_k^*(t)$  формируется центр притяжения  $x_k^c(t)$  этой частицы. Среди позиций  $x^*(t)$  и  $x_k^*(t)$  определяется худшая –  $x_k^b(t)$  и лучшая –  $x_k^g(t)$ . Формирование  $x_k^c(t)$  осуществляется путем применения модифицированной процедуры перемещения к частице в худшей позиции  $x_k^b(t)$  для перемещения из нее в направлении к лучшей  $x_k^g(t)$ . После определения центра притяжения  $x_k^c(t)$  частица  $k$  с помощью процедуры перемещения перемещается в направлении  $x_k^c(t)$  из позиции  $x_k(t)$  в позицию  $x_k(t+1)$ .

Схема работы роевого алгоритма покрытия включает следующие шаги:

1. В соответствии с постановкой задачи покрытия и исходными данными формируется структура матрицы  $P_k$ . Формируется пространство решений, каждая ось которого соответствует строке  $P_k$ .

2. Создается исходная «случайная» популяция частиц,  $t=0$ . Для каждой частицы случайным образом задается начальная позиция  $x_k(0)$ . С этой целью формируется матрица  $P_k(0)$ . В каждую  $i$ -ю строку матрицы заносятся случайные значения элементов  $p_{kij}(0)$ , с соблюдением ограничения (2).

3. Шаги 4–10 итерационно повторяются заданное число раз.

4. Рассчитывается целевая функция  $f_k(t)$  для каждой частицы.

5. Определяются лучшая позиция роя  $x^*(t)$  на шаге  $t$  и значение целевой функции  $f^*(t)$  в этой позиции.

6. Для каждой частицы определяются лучшие позиции  $x_k^*(t)$ , которую она посещала с начала первой итерации, и значение целевой функции  $f_k^*(t)$  в этой позиции.
7. Для каждой частицы определяется центр притяжения  $x_k^c(t)$  этой частицы.
8. Рассчитываются новые позиции частиц в пространстве решений. С помощью процедуры перемещения каждая частица перемещается из позиции  $x_k(t)$  в позицию  $x_k(t+1)$ .
9.  $t=t+1$ .
10. Запоминается лучшее решение, найденное роем к моменту  $t$ .

По аналогии с эволюционными стратегиями, рой можно трактовать как популяцию, а частицу как индивида (хромосому). Это дает возможность построения гибридной структуры поиска решения, основанную на сочетании генетического поиска с методами роевого интеллекта. Связующим звеном такого подхода является структура данных, описывающая в виде хромосомы решение задачи. Если в качестве частицы используется хромосома, то число параметров, определяющих положение частицы в пространстве решений должно быть равно числу генов в хромосоме. В работе предлагается подход к построению структур и принципов кодирования хромосом, обеспечивающих их гомологичность и возможность одновременного использования в генетическом алгоритме, и в алгоритме на основе роя частиц [6]. Первый и наиболее простой подход к гибридизации заключается в следующем. С начала поиск решения осуществляется генетическим алгоритмом. 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.

Для анализа точности получаемых решений был синтезирован ряд примеров с априори известным оптимальным значением целевой функции. Исследованию подвергались примеры, содержащие до 1 000 элементов. В среднем 70 итераций достаточно для нахождения лучшего решения. Вероятность получения оптимального решения составила 0,9, а оценки локально оптимальных решений, отличались от глобального оптимума в среднем на 1 %. Сравнение с известными алгоритмами [2, 3] показало, что при меньшем времени работы у полученных с помощью гибридного алгоритма решений значения целевой функции лучше (меньше) в среднем на 6 %.

**Выводы.** На основе сравнительного анализа существующих подходов и методов для решения задачи покрытия использованы мультиагентные методы интеллектуальной оптимизации, базирующиеся на моделировании коллективного интеллекта, позволяющие работать с задачами покрытия большой размерности и получать качественные результаты за приемлемое время.

В отличие от стандартной парадигмы метода роя частиц, вместо метрической (числовой) шкалы используется *неупорядоченная лингвистическая шкала*. Впервые разработан метод перемещения частиц в пространстве поиска, связанный с матричным представлением позиций (решений). Отметим, что процедура перемещения фактически является оператором (назовем его роевым) с помощью которого изменяется текущее решение. Это дает возможность построения гибридной структуры поиска решения, основанную на сочетании генетического поиска [6] с методами роевого интеллекта. Связующим звеном такого подхода является структура данных, описывающая в виде хромосомы решение задачи.

Разработанный алгоритм покрытия имеет универсальный характер и может быть использован для решения широкого круга задач линейного целочисленного программирования.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Деньдобренко Б.П., Малика А.С.* Автоматизация проектирования радиоэлектронной аппаратуры. – М.: Высш. шк., 2002.
2. *Coudert O.* “On solving covering problems”, in proceedings of 30th ACM // IEEE Design automation conference. – 1996. – P. 197-202.
3. *Cordone R., Ferrandi F., Sciuto D., Calvo R.W.* An Efficient Heuristic Approach to Solve the Unate Covering Problem // IEEE Transactions on computer-aided design of integrated circuits and systems. – December 2001. – Vol. 120, № 12. – P. 1377-1387.
4. *Лебедев О.Б.* Покрытие методом муравьиной колонии // Двенадцатая национальная конференция по искусственному интеллекту с международным участием КИИ-2010. Труды конференции. Т. 2. – М.: Физматлит, 2010. – С. 423-431.
5. *Курейчик В.М., Лебедев Б.К., Лебедев О.Б.* Поисковая адаптация: теория и практика. – М.: Физматлит, 2006.
6. *Курейчик В.М., Лебедев Б.К., Лебедев О.Б.* Решение задачи покрытия на основе эволюционного моделирования // Теория и системы управления. – 2009. – № 1. – С. 101-116.
7. *Лебедев Б.К., Лебедев В.Б.* Размещение на основе метода пчелиной колонии // Известия ЮФУ. Технические науки. – 2010. – № 12 (113). – С. 12-19.
8. *Лебедев Б.К., Лебедев О.Б.* Моделирование адаптивного поведения муравьиной колонии при поиске решений, интерпретируемых деревьями // Известия ЮФУ. Технические науки. – 2012. – № 7 (132). – С. 27-34.
9. *Poli R.* Analysis of the publications on the applications of particle swarm optimization // Journal of Artificial Evolution and Applications, Article ID 685175. – 2008. – P. 10.
10. *Clerc M.* Particle Swarm Optimization. ISTE, London, UK, 2006.

Статью рекомендовал к опубликованию д.т.н., профессор В.Н. Иванченко.

**Лебедев Владимир Борисович** – Южный федеральный университет; e-mail: lbk@tsure.ru; 347928, г. Таганрог, пер. Некрасовский, 44; тел.: 88634371651; кафедра системного анализа и телекоммуникаций; к.т.н., доцент.

**Lebedev Vladimir Borisovich** – Southern Federal University; e-mail: lbk@tsure.ru; 44, Nekrasovsky, Taganrog, 347928, Russia; phone: 88634371651; the department of system analysis and telecommunications; associate professor.