

**Kureichik Vladimir Viktorovich**

Taganrog Institute of Technology – Federal State-Owned Autonomy Educational Establishment of Higher Vocational Education “Southern Federal University”.

E-mail: vkur@tsure.ru.

44, Nekrasovskiy, Taganrog, 347928, Russia.

Phone: 88634383451.

The Department of Computer Aided Design; Head of the Department; Professor.

**Polupanov Aleksey Aleksandrovich**

Federal State-Owned Educational Establishment of Higher Vocational Education «Admiral F.F. Ushakov Marine State Academy».

E-mail: polualex@mail.ru.

195, ap. 47, Dzerzhinskogo Street, Novorossiysk, 353900, Russia.

Phone: +79282734490.

Department of Automation and Computing Technics; Associate Professor.

УДК 681.325

**Б.К. Лебедев, В.Б. Лебедев**

**РАЗМЕЩЕНИЕ НА ОСНОВЕ МЕТОДА ПЧЕЛИНОЙ КОЛОНИИ\***

*Рассматривается алгоритм решения задачи размещения, на основе моделирования поведения пчелиной колонии. Описаны модели поведения агентов разведчиков и агентов фуражиров, механизмы поиска и выбор позиций в заданной окрестности. Приведена общая структура оптимизационного процесса. Экспериментальные исследования проводились на IBM PC. По сравнению с существующими алгоритмами достигнуто улучшение результатов.*

*Роевой интеллект; пчелиная колония; адаптивное поведение; самоорганизация, размещение, оптимизация.*

**B.K. Lebedev, V.B. Lebedev**

**PLACEMENT ON THE BASIS OF THE BEER COLONY METHOD**

*In work the algorithm of the decision of a placing problem, on the basis of modeling of a beer colony behaviour is considered. Models of behaviour of scouts agents and foragers agents, search engines and a choice of positions in the set vicinity are described. The general structure of optimization process is resulted. Experimental researches were spent on IBM PC. In comparison with existing algorithms improvement of results is reached.*

*Swarm intelligence; a beer colony; adaptive behaviour; self-organizing; placement; optimization.*

**Введение.** Разработка методов и алгоритмов для решения задачи размещения осуществляется на протяжении многих лет, но, по-прежнему, является актуальной. Это связано, в первую очередь, с тем, что эта задача является NP-полной, и разработать универсальный алгоритм, позволяющий находить точное оптимальное решение за приемлемое время затруднительно. Появление новых более совершенных средств вычислительной техники, дающих мощные вычислительные ресурсы, а также повышение требований к проектируемым устройствам, является побудительной причиной разработки новых алгоритмов размещения. Одним из новых направлений таких методов являются мультиагентные методы интеллектуальной

\* Работа выполнена при поддержке: РФФИ (грант № 10-07-00055), г/б № 2.1.2.1652.

оптимизации [1], базирующиеся на моделировании коллективного интеллекта. Такие методы являются итеративными, эвристическими методами случайного поиска. Среди них особенно активно развиваются методы роевого интеллекта (Swarm Intelligence) [2-4], в которых совокупность сравнительно простых агентов конструирует стратегию своего поведения без наличия глобального управления.

Одним из новейших мультиагентных методов интеллектуальной оптимизации является метод пчелиной колонии [5-9]. Колонии социальных насекомых, таких как пчелы, имеют инстинктивные способности, известные как коллективный интеллект. Высокоорганизованное поведение позволяет колониям насекомых решать проблемы, выходящие за рамки возможностей отдельных членов колонии, путем коллективных действий и примитивного взаимодействия между членами группы. Такая колония характеризуется самоорганизацией, адаптивностью,

Основные механизмы поведения пчел заключаются в следующем. Сначала из улья вылетают в случайном направлении какое-то количество пчел-разведчиков, которые отыскивают источники, где есть нектар. Через какое-то время пчелы возвращаются в улей и особым образом сообщают остальным – где и сколько они нашли нектара. После этого на найденные источники отправляются другие пчелы, причем, чем больше на некотором источнике предполагается найти нектара, тем больше пчел летит в этом направлении. А разведчики опять улетают искать другие источники, после чего процесс повторяется. Иммитация самоорганизации пчелиной колонии составляет основу пчелиного алгоритма оптимизации – нового перспективного метода природных вычислений.

В работе, в соответствии с предложенными в [6,7] математическими моделями поведения пчёл, разработана модификация метода пчелиной колонии для решения задачи размещения, позволяющая находить качественные решения задач большой размерности.

**Постановка задачи.** Пусть дано множество элементов  $A = \{a_j | j=1, 2, \dots, n\}$  и множество позиций  $P = \{\pi_i | i=1, 2, \dots, c\}$  на коммутационном поле (КП). В качестве модели схемы используется гиперграф  $H = (X, E)$ , где  $X = \{x_i | i=1, 2, \dots, n\}$  – множество вершин, моделирующих элементы,  $E = \{e_j | e_j \subset X, j=1, 2, \dots, m\}$  – множество гиперребер, моделирующих цепи, связывающие элементы. Для размещения всех элементов необходимо выполнение условия  $c \geq n$ . Некоторое  $s$ -е решение задачи размещения элементов в позициях представляет собой перестановку  $V_s = \langle v_{s1}, \dots, v_{s2}, \dots, v_{sc} \rangle$ , где  $v_{si}$  задает номер элемента, который назначен в позицию  $\pi_i$ . В зависимости от выбранного критерия для оценки результатов размещения вводится целевая функция  $F(V_s)$ . Задача сводится к отысканию для каждого элемента таких позиций на КП кристалла СБИС, при которых оптимизируется выбранный показатель качества.

В качестве оценки  $l_j$  длины цепи  $t_j$ , моделируемой гиперребром  $e_j$ , используются: длина минимального связывающего дерева, построенного на множестве вершин  $e_j \subset X$ ; длина звездного графа, ребра которого инцидентны вершинам  $e_j \subset X$ , а корневая вершина помещена в центре “тяжести” множества вершин  $e_j$ ; длина полупериметра прямоугольника, описывающего множество вершин  $e_j$ ; суммарная длина ребер полного графа, построенного на множестве  $e_j$ .

С учетом этого, критерий оптимизации имеет вид:

$$F = \sum_{j=1}^m l_j .$$

Задача размещения состоит в отыскании оптимального значения функции  $F$  на множестве перестановок  $V$ . Для более полного учета связей между задачами размещения и трассировки используется критерий, основанный на оценках числа цепей, пересекающих заданные линии КП. Эти линии могут быть либо прямыми, пересекающими все КП, либо замкнутыми и ограничивающими некоторую область [1]. В настоящее время проводятся интенсивные исследования проблемы размещения, управляемого временными задержками. Разработка расширяемого, мощного и устойчивого алгоритма минимизации временной задержки схемы в процессе размещения является ключевой задачей в разработке топологии микросхем [10].

В работе основное внимание сосредоточено на построении поисковой процедуры размещения оптимизирующей выбранный показатель качества.

**Представления решений в алгоритме размещения на основе роевого интеллекта.** Основу поведения пчелиного роя составляет самоорганизация, обеспечивающая достижение общих целей роя на основе низкоуровневого взаимодействия. Основная идея парадигмы пчелиной колонии заключается в использовании двухуровневой стратегии поиска. На первом уровне с помощью пчел разведчиков формируется множество перспективных областей, на втором уровне с помощью пчел фуражиров осуществляется исследование окрестностей данных областей. Цель пчелиной колонии найти источник, содержащий максимальное количество нектара.

В алгоритме размещения каждое решение представляется в виде точки (позиции)  $x_s$  в многомерном пространстве, где каждая координата позиции  $x_i$  представляет собой один параметр функции  $F_s$ , которую надо оптимизировать. Каждая позиция является аналогией источника нектара. Количество нектара содержащегося в некотором источнике определяется координатами соответствующей позиции. Найденное количество нектара представляет собой значение целевой функции в этой точке.

В эвристических алгоритмах роевого интеллекта процесс поиска решений заключается в последовательном перемещении агентов  $b_k$  в многомерном пространстве поиска. Процесс поиска решений итерационный. На каждой итерации  $l$  каждый агент  $b_k$  перемещается в новую позицию  $x_k(l)$ . Значения целевой функции  $F$  определяются позициями, выбранными агентами.

Разработка поведенческой модели самоорганизации колонии пчёл, заключаются в разработке методов и механизмов:

- ◆ формирования роя агентов разведчиков и роя агентов фуражиров;
- ◆ поиска перспективных позиций разведчиками;
- ◆ выбора базовых позиций среди перспективных для исследования их окрестностей;
- ◆ передачи информации между разведчиками и фуражирами;
- ◆ выбора фуражирами базовых позиций;
- ◆ выбора фуражирами позиций в окрестности базовой позиций;
- ◆ общей структуры оптимизационного процесса.

Первая задача при разработке алгоритма на основе парадигмы пчелиной колонии заключается в формировании пространства поиска.

Представим множество позиций КП в виде ортогональной решетки. Позиции пронумерованы. Пусть  $\pi_i$  – позиция КП. Каждая  $\pi_i$  имеет свои координаты – номер строки и номер столбца решетки. Информация о координатах позиций хранится в виде двумерной матрицы  $K = \|k_{ij}\|_{c \times 2}$ , где  $k_{i1}$  и  $k_{i2}$  – координаты позиции  $\pi_i$  в решетке,  $k_{i1}$  – номер строки,  $k_{i2}$  – номер столбца. Позиция  $\pi_i = (k_{i1}, k_{i2})$  расположена на пересече-

чении строки  $k_{i1}$  и столбца  $k_{i2}$ . Будем в дальнейшем пару координат  $(k_{i1}, k_{i2})$  рассматривать в качестве эквивалента позиции  $\pi_i$ . То есть  $\pi_i = (k_{i1}, k_{i2})$ . Некоторое  $s$ -е решение задачи размещения элементов в позициях КП представляет собой перестановку  $V_s = \langle v_{s1}, \dots, v_{si}, \dots, v_{sc} \rangle$ , где  $v_{si}$  задает номер элемента, который назначен в позицию  $\pi_i$ .

Теперь рассмотрим пространство решений. В работе в отличие от стандартной парадигмы вместо метрической (числовой) шкалы используется *неупорядоченная лингвистическая шкала*. Пространство решений содержит одну ось. В нашем случае позиция пространства решений  $x_s$  соответствует решению, задаваемому перестановкой  $V_s$ . Точками отсчета на оси являются значения лингвистической переменной  $x_s$  неупорядоченные друг относительно друга. Значением  $x_s$  является некоторая комбинация взаимного расположения элементов в векторе  $V_s$ . Отметим, что точки отсчета  $x_s$  на оси определяются после нахождения агентами роя решений  $V_s$  и откладываются в порядке нахождения этих решений.

Ключевой операцией алгоритма является исследование перспективных позиций и их окрестностей в пространстве решений. Рассмотрим принципы формирования окрестности позиции в пространстве решений, опираясь на понятие окрестности позиции в коммутационном поле, то есть окрестности позиции в ортогональной решетке.

Назовем  $\delta$ -окрестностью позиции  $\pi_i$  множество позиций  $\Pi_i$  таких, что координаты любой из позиций множества  $\Pi_i$  отличаются от координат позиции  $\pi_i$  не более, чем на величину  $\delta$ . Пусть  $\delta=1$ , тогда  $\delta$ -окрестностью позиции  $\pi_i = (k_{i1}, k_{i2})$ , является множество позиций  $\Pi_i = \{(k_{i1}+1, k_{i2}), (k_{i1}-1, k_{i2}), (k_{i1}, k_{i2}+1), (k_{i1}, k_{i2}-1), (k_{i1}+1, k_{i2}+1), (k_{i1}+1, k_{i2}-1), (k_{i1}-1, k_{i2}+1), (k_{i1}-1, k_{i2}-1)\}$ . Назовем две позиции соседними, если одна из них лежит в окрестности другой. На рис. 1 представлены позиции КП. Позиция  $\pi_i$  залита черным цветом. Множество заштрихованных вершин входят в  $\delta$ -окрестность  $\Pi_i$  позиции  $\pi_i$  для  $\delta = 1$ .

Пусть множество элементов  $A = \{a_j | j=1, 2, \dots, n\}$  размещено на КП в соответствии с решением  $V_s$ . После любой взаимной перестановки пары элементов  $(a_i, a_m)$ , один из которых расположен в позиции  $\pi_i$ , а другой в одной из позиций  $\pi_r$ , принадлежащей окрестности  $\Pi_i$ , будет получено новое решение. Пусть паре элементов  $(a_i, a_m)$ , размещенных в соседних позициях  $(\pi_i, \pi_r)$ , соответствуют элементы  $(v_{si}, v_{sr})$  вектора  $V_s$ . Перестановке на КП пары элементов  $(a_i, a_m)$  соответствует перестановка пары элементов  $(v_{si}, v_{sr})$  в векторе  $V_s$ . В результате чего образуется вектор  $V_w$ . Будем считать, что полученное таким способом решение – вектор  $V_w$  лежит в окрестности решения  $V_s$ . В общем случае окрестность решения  $V_s$  составляют решения, полученные путем парной перестановки элементов, расположенных в двух соседних позициях КП. Отметим, что перестановка пары соседних элементов в векторе  $V_s$  не всегда приводит к получению решения  $V_w$ , лежащего в окрестности решения  $V_s$ .

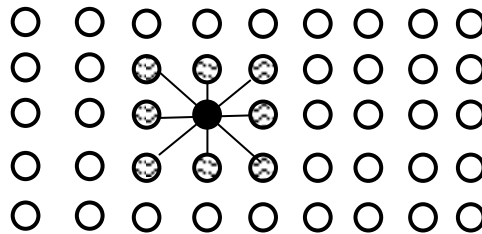


Рис. 1. Множество позиций КП в виде ортогональной решетки;  $\delta$ -окрестность позиции  $\pi_i$

**Организация поисковых процедур на основе моделировании адаптивного поведения пчелиной колонии.** На первом шаге задаются основные параметры метода пчелиной колонии: количество агентов  $n_b$ , максимальное количество итераций  $L$ , начальное количество агентов-разведчиков  $n_r$ , ограничение максимального количества агентов-разведчиков, пороговое значение размера окрестности  $\delta$  и др.

В начале процесса поиска все агенты расположены в улье, т.е. вне пространства поиска.

На первой итерации ( $l=1$ ) агенты-разведчики в количестве  $n_r$  случайным образом размещаются в пространстве поиска. Эта операция заключается в генерации случайным образом множества отличающихся друг от друга решений  $V(1) = \{V_s(1) | s=1, 2, \dots, n_r\}$ , которым соответствует множество позиций  $X(1) = \{x_s(1) | s=1, 2, \dots, n_r\}$ . Для каждого решения  $V_s(1)$  вычисляется значение целевой функции  $F_s(1)$ . Выбирается  $n_b$  базовых (лучших) решений, у которых значения целевой функции не меньше, чем значение целевой функции у любого не выбранного решения. Формируется множество базовых (лучших) решений  $V^{\delta}(1)$  и соответствующее ему множество базовых (лучших) позиций  $X^{\delta}(1)$ .

Предлагается три подхода к определению числа агентов фуражиров, направляемых в окрестности базовых позиций, выбранных агентами разведчиками. При первом подходе агенты фуражиры распределяются по базовым позициям равномерно. При втором подходе агенты фуражиры распределяются по базовым позициям пропорционально значению целевой функции позиции. При третьем подходе реализуется вероятностный выбор. Вероятность  $p(x_s)$  выбора агентом фуражиром базовой позиции  $x_s^{\delta}(1) \in X^{\delta}(1)$  пропорциональна значению целевой функции  $F_s(1)$  в этой позиции и определяется как

$$p(x_s^{\delta}(1)) = F_s^{\delta}(1) / \sum_s (F_s^{\delta}(1)).$$

После выбора агентом фуражиром  $b_z$  базовой позиции  $x_s^{\delta}(1) \in X^{\delta}(1)$ , реализуется вероятностный выбор позиции  $x_z(1)$ , расположенной в окрестности базовой позиции  $x_s^{\delta}(1)$ . Вероятностный выбор позиции  $x_z(1)$  осуществляется следующим образом. На КП случайным образом выбирается позиция  $\pi_i$ , а затем в  $\delta$ -окрестности позиции  $\pi_i$  случайным образом выбирается позиция  $\pi_r$ . После перестановки пары элементов  $(v_{s\pi}, v_{s\pi_r})$  вектора  $V_s(1)$ , соответствующих паре  $(\pi_i, \pi_r)$ , образуется новое решение  $V_z(1)$  и соответствующая ему позиция  $x_z(1)$  в которую и помещается фуражир  $b_z$ .

Обозначим множество позиций, выбранных агентами фуражирами в окрестности позиции  $x_s(l)$  как  $O_s(l)$ . Назовем множество позиций  $O_s(l) \cup x_s^{\delta}(l)$  областью  $D_s(l)$ . В каждой области  $D_s(l)$  выбирается лучшая позиция  $x_s^*(l)$  с лучшим решением  $V_s^*(l)$ . Назовем решение  $V_s^*(l)$  оценкой области  $D_s(l)$ . Среди  $V_s^*(l)$  выбирается лучшее решение  $V^*(l)$ , найденное на данной итерации совместно роем разведчиков и роем фуражиров. Лучшее решение  $V^*(l)$  сохраняется, а затем происходит переход к следующей итерации.

Отметим, что в рассматриваемой парадигме пчелиной колонии не важно, знать, каким агентом выбрана позиция в пространстве поиска. Важно знать число агентов-разведчиков и число агентов фуражиров, а также – какие именно позиции выбраны агентами-разведчиками и какие – агентами фуражирами.

В начале второй и на последующих итерациях, прежде всего, формируется множество базовых позиций  $X^{\delta}(l)$  ( $l=2, 3, \dots, L$ ), составленное из двух частей  $X^{\delta 1}(l)$  и  $X^{\delta 2}(l)$ ,  $X^{\delta 1}(l) \cup X^{\delta 2}(l) = X^{\delta}(l)$ . В первую часть  $X^{\delta 1}(l)$  включаются  $n_{b1}$  лучших позиций, среди позиций  $x_s^*$ , найденных агентами в каждой из областей, сформированных на предыдущей итерации. Вторая часть  $X^{\delta 2}(l)$  формируется также, как и на первой ите-

рации. Отличие заключается в числе  $n_{r,l}$  агентов разведчиков, выбирающих случайным способом новые позиции.  $n_{r,l} < n_r$ . В множество  $X^{\delta^2}(l)$  включается  $n_{\delta^2}$  лучших позиций из  $n_{r,l}$  новых позиций, найденных агентами разведчиками на  $l$ -ой итерации.  $n_{\delta^1} + n_{\delta^2} = n_{\delta}$ . Далее выполняются действия, аналогичные действиям, рассмотренным на первой итерации. Рассчитывается число агентов фуражиров, направляемых в окрестности каждой базовой позиции. Каждым агентом фуражиром  $b_z$  выбирается базовая позиция  $x_s(l)$  и позиция  $x_z(l)$ , расположенная в окрестности этой базовой позиции. В каждой области  $D_s(l)$  выбирается лучшая позиция  $x_s^*(l)$  с лучшим решением  $V_s^*(l)$ . Среди  $V_s^*(l)$  выбирается лучшее решение  $V^*(l)$ . Если  $V^*(l)$  лучше  $V^*(l-1)$ , то оно сохраняется, а затем происходит переход к следующей итерации.

Схема работы роевого алгоритма размещения включает следующие шаги:

1. Задаются основные параметры метода пчелиной колонии:

$L$  – максимальное количество итераций;

$n_r$  – начальное количество агентов-разведчиков;

$n_{\delta}$  – количество базовых позиций;

$\delta$  – пороговое значение размера окрестности;

$n_f$  – начальное количество агентов фуражиров;

$n_{\delta^1}$  – количество базовых позиций, формируемых из лучших позиций  $x_s^*(l)$ , найденных роем на  $l$ -ой итерации;

$n_{r,l}$  – количество агентов-разведчиков, выбирающих случайным способом новые позиции;

$n_{\delta^2}$  – количество базовых позиций, формируемых из лучших новых позиций, найденных агентами разведчиками на  $l$ -ой итерации.

2. Генерация случайным образом множества отличающихся друг от друга решений  $V(I) = \{V_s(I) | s=1, 2, \dots, n_r\}$ , которым соответствует множество позиций  $X(I) = \{x_s(I) | s=1, 2, \dots, n_r\}$ .

3. Для каждого решения  $V_s(I)$  вычисляется значение целевой функции  $F_s(I)$ .

4. Формируется множество базовых решений  $V^{\delta}(I) \subset V(I)$  с лучшими значениями целевых функций  $F_s(I)$  и соответствующее множество базовых позиций  $X^{\delta}(I) \subset X(I)$ .  $|V^{\delta}(I)| = |X^{\delta}(I)| = n_{\delta}$ .

5.  $l=1$ . ( $l$  – номер итерации).

6.  $z=1$ . ( $z$  – порядковый номер агента фуражира).

7. Выбор с вероятностью  $p(x_s^{\delta}(l)) = F_s^{\delta}(l) / \sum_s (F_s^{\delta}(l))$  базовой позиции  $x_s^{\delta}(l) \in X^{\delta}(l)$ .

8. Вероятностный выбор позиции  $x_z(l)$ , расположенной в окрестности базовой позиции  $x_s^{\delta}(l)$ , с соответствующим решением  $V_z(l)$ .

9. Если позиция  $x_z(l)$  совпадает с ранее выбранными позициями, то переход к 8, иначе переход к 10.

10. Позиция  $x_z(l)$  включается в множество  $O_s(l)$ .

11. Расчет значения целевой функции  $F_z(l)$  решения  $V_z(l)$ .

12. Если  $z < n_f$ , то  $z = z+1$  и переход к 6, иначе переход к 13.

13. Формирование для каждой базовой позиции  $x_s^{\delta}(l)$  области

$$D_s(l) = O_s(l) \cup X_s^{\delta}(l).$$

14. В каждой области  $D_s(l)$  выбирается лучшая позиция  $x_s^*(l)$  с лучшим решением  $V_s^*(l)$ .

15. Среди  $V_s^*(l)$  выбирается лучшее решение  $V^*(l)$ .

16. Если  $V^*(l)$  лучше  $V^*(l-1)$ , то оно сохраняется, иначе  $V^*(l) = V^*(l-1)$ .

17. Если  $l < L$ , то  $l = l+1$  и переход к 18, иначе переход к 22.

18. В первую часть  $X^{\delta^1}(l)$  включают  $n_{\delta^1}$  лучших позиций, среди позиций  $x_s^*(l-1)$ , найденных агентами в каждой из  $D_s(l-1)$  областей, сформированных на предыдущей итерации.

19. Генерация случайным образом множества отличающихся друг от друга решений  $V(l) = \{V_s(l) | s=1,2,\dots,n_{r1}\}$ , которым соответствует множество позиций  $X(l) = \{x_s(l) | s=1,2,\dots,n_{r1}\}$ .  $|V(l)| = n_{r1}$ .

20. Включение в множество  $X^{\delta 2}(l)$   $n_{\delta 2}$  лучших позиций из множества  $X(l)$  новых позиций, найденных агентами разведчиками на  $l$ -ой итерации.  $n_{\delta 1} + n_{\delta 2} = n_{\delta}$ .

21. Формирование множества базовых позиций.  $X^{\delta}(l) = X^{\delta 1}(l) \cup X^{\delta 2}(l)$ . Переход к 6.

22. Конец работы алгоритма. Решение  $V^*(l)$  – лучшее решение, найденное роением агентов.

Временная сложность этого алгоритма зависит от времени жизни колонии  $l$  (число итераций), количества позиций  $c$  и числа агентов  $m$ , и определяется как  $O(l * c^2 * m)$ .

**Заключение.** На основе сравнительного анализа существующих подходов и методов для решения задачи размещения использованы мультиагентные методы интеллектуальной оптимизации, базирующиеся на моделировании адаптивного поведения пчелиной колонии. В работе задача размещения представлена в виде набора компонент пчелиного алгоритма. Описаны модели поведения агентов разведчиков и агентов фуражиров, механизмы поиска и выбора позиций в заданной окрестности, усиливающие сходимость алгоритма.

В отличие от стандартной парадигмы роевого метода вместо метрической (числовой) шкалы используются *неупорядоченная лингвистическая шкала измерений*. Повышения эффективности роевого алгоритма можно добиться путем адаптивного управления параметрами процедуры размещения, такими как размер  $\delta$ -окрестности позиции  $\pi_i$ , соотношение между числом агентов фуражиров и агентов разведчиков и т.д.

Экспериментальные исследования проводились на IBM PC. Для анализа точности получаемых решений был синтезирован ряд примеров с априори известным оптимальным значением целевой функции. Исследованию подвергались примеры, содержащие до 1000 элементов. Представленный роевой алгоритм размещения находит решения для задач большой размерности, не уступающие по качеству, а иногда и превосходящие своих аналогов с меньшими временными затратами. Вероятность получения оптимального решения составила 0.9. Сравнение с известными алгоритмами показало, что при меньшем времени работы у полученных с помощью пчелиного алгоритма решений значения целевой функции лучше (меньше) в среднем на 6 %. В среднем запуск программы обеспечивает нахождения решения, отличающегося от оптимального менее, чем на 1 %.

#### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Лебедев Б.К., Лебедев О.Б. Методы размещения. Монография. – Таганрог: Изд-во ТРТУ, 2006.
2. Engelbrecht A.P. Fundamentals of Computational Swarm Intelligence. John Wiley & Sons, Chichester, UK, 2005.
3. Курейчик В.М., Лебедев Б.К., Лебедев О.Б. Поисковая адаптация: теория и практика. – М.: Физматлит, 2006.
4. Курейчик В.М., Лебедев Б.К., Лебедев О.Б. Адаптация в задачах проектирования топологии. // Проблемы разработки перспективных микро- и нанoeлектронных систем. – 2010. Сборник научных трудов / Под ред. А.Л. Стемповского. – М.: ИППМ РАН, 2010. – С. 170-177.
5. Lučić P., Teodorović D. Computing with Bees: Attacking Complex Transportation Engineering Problems // International Journal on Artificial Intelligence Tools. – 2003. – № 12. – P. 375-394.

6. *Teodorović D., Dell'Orco M.* Bee Colony Optimization – a Cooperative Learning Approach to Complex Transportation Problems // *Advanced OR and AI Methods in Transportation: Proceedings of 16th Mini-EURO Conference and 10th Meeting of EWGT (13-16 September 2005)*. – Poznan: Publishing House of the Polish Operational and System Research, 2005. – P. 51-60.
7. *Quijano N., Passino K.M.* Honey Bee Social Foraging Algorithms for Resource Allocation: Theory and Application. – Columbus: Publishing house of the Ohio State University, 2007. – 39 p.
8. *Clerc M.* Particle Swarm Optimization. ISTE, London, UK, 2006.
9. *Курейчик В.В., Полуанова Е.Е.* Эволюционная оптимизация на основе алгоритма колонии пчел // *Известия ЮФУ. Технические науки*. – 2009. – № 12 (101). – С. 41-46.
10. *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.

**Лебедев Борис Константинович**

Технологический институт федерального государственного автономного образовательного учреждения высшего профессионального образования «Южный федеральный университет» в г. Таганроге.

E-mail: lbk@tsure.ru.

347928, г. Таганрог, пер. Некрасовский, 44.

Тел.: 88634371743.

Кафедра систем автоматизированного проектирования; профессор.

**Лебедев Владимир Борисович**

Кафедра системного анализа и телекоммуникаций; доцент.

**Lebedev Boris Konstantinovich**

Taganrog Institute of Technology – Federal State-Owned Autonomy Educational Establishment of Higher Vocational Education “Southern Federal University”.

E-mail: lbk@tsure.ru.

44, Nekrasovsky, Taganrog, 347928, Russia.

Phone: +78634371743.

Department of Computer Aided Design; Professor.

**Lebedev Vladimir Borisovich**

Department of System Analysis and Telecommunications; Associate Professor.

УДК 681.3.001.63

**В.Н. Кучуганов, М.А. Наборщиков**

**РАСПРЕДЕЛЕНИЕ РАБОТ С ПОМОЩЬЮ ГЕНЕТИЧЕСКОГО  
АЛГОРИТМА**

*Предложен эволюционный метод проектирования технологических схем распределения труда. Укрупненно порядок действий предопределен технологическим процессом. Конкретный план работ формируется исходя из реального состояния ресурсов, в первую очередь, невозобновляемых: оборудование, люди, время. Алгоритм рассмотрен на примере работы цеха по пошиву мужской и женской защитной одежды.*

*Планирование производственное; процесс технологический; ресурс невозобновляемый; алгоритм генетический; локус; место рабочее.*

**V.N. Kuchuganov, M.A. Naborschikov**

**WORK DISTRIBUTION BY MEANS OF A GENETIC ALGORITHM**

*In the paper an evolutionary method for the design of work distribution flowcharts is proposed. In general terms the workflow is predetermined by the manufacturing process. The concrete working schedule is generated basing on the as-is state of the resources, above all of nonrenewa-*