



УДК 681.385

РАСПРЕДЕЛЕНИЕ РЕСУРСОВ НА ОСНОВЕ ГИБРИДНЫХ МОДЕЛЕЙ РОЕВОГО ИНТЕЛЛЕКТА

Б.К. Лебедев^а, О.Б. Лебедев^а, Е.М. Лебедева^а^а Южный федеральный университет, Ростов-на-Дону, 344006, Российская Федерация

Адрес для переписки: lebedev.ob@mail.ru

Информация о статье

Поступила в редакцию 03.10.17, принята к печати 31.10.17

doi: 10.17586/2226-1494-2017-17-6-1063-1073

Язык статьи – русский

Ссылка для цитирования: Лебедев Б.К., Лебедев О.Б., Лебедева Е.М. Распределение ресурсов на основе гибридных моделей роевого интеллекта // Научно-технический вестник информационных технологий, механики и оптики. 2017. Т. 17. № 6. С. 1063–1073. doi: 10.17586/2226-1494-2017-17-6-1063-1073

Аннотация

Предложена композитная архитектура многоагентной системы бионического поиска для решения общей распределительной задачи на основе интеграции роевого интеллекта и генетической эволюции. Рассмотрены три подхода к построению такой архитектуры. Связующим звеном такого подхода является единая структура данных, описывающая в виде хромосомы решение задачи. Рассмотренные в работе новые принципы и способы кодирования и декодирования хромосом для представления решения общей распределительной задачи исключают некорректные решения, отличаются простотой и линейными оценками временной и пространственной сложности. Предложена модифицированная парадигма метода роя частиц. Для организации перемещения роя частиц в гиперпространстве решений разработан оператор направленной мутации. Эксперименты показали, что качество решений у гибридного алгоритма на 10–15% лучше, чем у генетического и роевого алгоритмов. Общая оценка временной сложности при любом подходе к гибридизации не превышает оценки временной сложности генетического алгоритма и лежит в пределах $O(n^2)$ – $O(n^3)$.

Ключевые слова

распределительная задача, многоагентная система, роевой интеллект, генетическая эволюция, бионический поиск, гибридизация

Благодарности

Работа выполнена при финансовой поддержке РФФИ грант № 17-07-00997.

DISTRIBUTION OF RESOURCES BASED ON HYBRID MODELS OF SWARM INTELLIGENCE

B.K. Lebedev^a, O.B. Lebedev^a, E.M. Lebedeva^a^a South Federal University, Rostov-on-Don, 344006, Russian Federation

Corresponding author: lebedev.ob@mail.ru

Article info

Received 03.10.17, accepted 31.10.17

doi: 10.17586/2226-1494-2017-17-6-1063-1073

Article in Russian

For citation: Lebedev B.K., Lebedev O.B., Lebedeva E.M. Distribution of resources based on hybrid models of swarm intelligence. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2017, vol. 17, no. 6, pp. 1063–1073 (in Russian). doi: 10.17586/2226-1494-2017-17-6-1063-1073

Abstract

The composite architecture of the multi-agent bionic search system is proposed to solve the general distribution problem based on swarm intelligence and genetic evolution. Three approaches to the construction of such architecture are considered. The connecting link of this approach is a unified data structure that describes the solution of the problem in the form of a chromosome. The new principles and methods of coding and decoding of chromosomes for the representation of the general distribution problem considered in this paper exclude incorrect solutions, are distinguished by simplicity and linear estimates of temporal and spatial complexity. A modified paradigm of the particle swarm method is proposed. To organize the swarm movement of particles in hyperspace of solutions, a directed mutation operator has been developed. Experiments have shown that the quality of the solutions in the hybrid algorithm is 10 to 15% better than the genetic and swarm algorithms. The overall estimate of time complexity for any hybridization approach does not exceed the estimate of the time complexity of the genetic algorithm and lies within the range $O(n^2)$ – $O(n^3)$.

Keywords

general distribution problem, multi-agent system, swarm intelligence, genetic evolution, bionic search, hybridization

Acknowledgements

This research was supported by the grant from the Russian Foundation for Basic Research No.17-07-00997.

Введение

Распределительные задачи (P3) связаны с распределением ресурсов по работам, которые необходимо выполнить. Целью решения задачи является отыскание такого распределения ресурсов по работам, при котором либо минимизируются общие затраты, связанные с выполнением работ, либо максимизируется получаемый в результате общий доход. Задачи оптимального распределения ресурсов возникают в различных областях науки, техники и социальных сферах. Характер распределяемых ресурсов и смысл оптимальности может быть различным в зависимости от рассматриваемой прикладной области и конкретной задачи.

В силу того, что P3 принадлежат к классу NP-полных задач, создание эффективных методов их решения является актуальной проблемой теории расписаний. Для решения P3 российскими и зарубежными учеными разработано большое количество методов и алгоритмов, различающихся как областью применения, так и эксплуатационными свойствами. Методы решения P3 принято разделять на две большие группы – точные и приближенные. Оптимальное решение неоднородной P3 чрезвычайно сложно даже для современных вычислительных систем, особенно при использовании точных методов решения [1, 2]. В работе [3] в качестве метода точного решения используется наиболее универсальный и прогрессивный алгоритм Алексева и его модификации. Значительный класс задач распределения ресурсов решается с использованием методов линейного и динамического программирования [4–7]. Одним из наиболее известных и эффективных точных методов решения P3 являются алгоритмы [8, 9], построенные по принципам метода ветвей и границ (МВГ). Однако особенность МВГ заключается в том, что всегда найдется пример, решение которого «упрется» в полный перебор, и, несмотря на уменьшение комбинаторной сложности, потребуются весьма значительные затраты времени, особенно для задач повышенной размерности. Использование точных методов для решения неоднородной P3 при ее большой размерности за доступное время оказывается невозможным. Для проблем с более чем 15 рабочими местами и машинами такие методы оказываются непрактичными, поскольку они могут работать на современном оборудовании много лет подряд.

Приближенные методы не гарантируют нахождения глобального оптимума P3 и ориентированы на получение некоторого приемлемого или допустимого решения [10–12]. Несмотря на невысокую точность методов данного класса, главным их достоинством является высокая скорость решения, характеризующаяся полиномиальной или даже линейной зависимостью от порядка задачи. Для практических задач значимая потеря точности не всегда является приемлемой. В последние годы интенсивно разрабатывается научное направление, объединяющее математические методы, в которых заложены принципы природных механизмов принятия решений [13, 14].

Основной причиной снижения популярности приближенных методов было появление метаэвристических подходов, таких как генетические алгоритмы (GA), Ant Colony Optimization (ACO), оптимизация роя частиц (PSO), табу поиск (TS), моделирование отжига (SA), алгоритм летучих мышей (Bat Algorithm, BA) и др. [13]. Такие методы являются итеративными, эвристическими методами случайного поиска. Среди них особенно активно развиваются методы генетического поиска [15–17] и роевого интеллекта (Swarm Intelligence) [18, 19], в которых совокупность простых агентов конструирует стратегию своего поведения без наличия глобального управления.

Эффективность генетических алгоритмов (ГА) во многом определяется как учетом специфики решаемой задачи, так и использованием новых и модифицированных процедур поиска. Повышения эффективности ГА можно добиться разработкой структуры хромосом, обладающих свойством гомологичности, что исключает возникновение нелегальных решений и упрощает выполнение генетических операторов, модифицирующих хромосомы. Достоинством ГА с такими хромосомами, прежде всего, является линейная оценка временной и пространственной сложности генетических процедур, выполняемых на каждой итерации, что дает возможность решать задачи большой размерности.

Для повышения эффективности в ГА используются структурирование хромосом и многохромосомные представления решения. Каждая часть структурированной хромосомы или каждая хромосома в случае многохромосомных представлений отвечает за определенный аспект решения. Это упрощает и делает более целенаправленным процесс генетического поиска. В процессе генетического поиска некоторые решения, содержащиеся в развивающейся популяции, бывают очень близки к оптимальным. Однако механизмы генетического поиска, реализующие случайные изменения, часто не находят ту цепочку изменений, которая приводит к оптимальному решению. Для этого нужны «осмысленные» изменения, направленные в сторону глобального оптимума. Такие свойства как раз присущи адаптивным поисковым процедурам на основе метода роя частиц. В настоящее время одним из основных путей повышения эф-

фактивности решения задач глобального поиска является разработка гибридных алгоритмов. В гибридных алгоритмах преимущества одного алгоритма могут компенсировать недостатки другого. В работе излагается метод решения распределительной задачи на основе интеграции генетической эволюции и роевого интеллекта [20, 21]. Предложена композитная архитектура многоагентной системы бионического поиска.

Постановка задачи

Если затраты (или доход), определяемые объемом x_{ij} ресурса i , выделенного на выполнение работы j , равны $x_{ij}c_{ij}$, то имеем линейную распределительную задачу.

Основные методы решения распределительных задач, в частности, линейного программирования (ЛП), построены на допущении, что объемы имеющихся в наличии ресурсов (b_i), требуемые объемы (a_j) и затраты (c_{ij}) точно известны. Общая распределительная задача ЛП – это распределительная задача, в которой работы и ресурсы (исполнители) выражаются в различных единицах измерения. Типичным примером такой задачи является организация выпуска разнородной продукции на оборудовании различных типов.

Если общий объем наличных ресурсов $\sum b_i$ ($i=1..m$) равен общей потребности в них $\sum a_j$ ($i=1..n$), то имеет место сбалансированная (закрытая) распределительная задача. Если же $\sum a_j \neq \sum b_i$, то задача называется несбалансированной (открытой). Задачи оптимального распределения ресурсов возникают в различных областях науки, техники и социальных сферах. Характер распределяемых ресурсов и смысл оптимальности могут быть различными в зависимости от рассматриваемой прикладной области и конкретной задачи.

Приведем простой пример такой задачи.

Пусть для изготовления набора продуктов $P=\{p_i|i=1,2,\dots,n\}$ в соответствии с планом используется набор станков $S=\{s_j|j=1,2,\dots,m\}$.

Введем: x_{ij} – планируемое количество продукта p_i , производимого на станке s_j ; c_{ij} – затраты на производство единицы продукта p_i на станке s_j ; a_j – суммарный ресурс станка s_j ; d_{ij} – расход ресурса при производстве единицы продукта p_i на станке s_j .

Целевая функция, отображающая суммарные затраты, имеет вид

$$F(x) = \sum_{i=1}^n \sum_{j=1}^m c_{ij} d_{ij} \Rightarrow \min. \quad (1)$$

Искомый набор x_{ij} должен удовлетворять следующим ограничениям:

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

$$\sum_{i=1}^n d_{ij} x_{ij} \leq a_j, j = 1, 2, \dots, m, \quad (3)$$

$$x_{ij} \geq 0, i = 1, 2, \dots, n, j = 1, 2, \dots, m. \quad (4)$$

При этом соотношения (2) обеспечивают выполнение плана заказов, а соотношения (3) и (4) учитывают ограничения на ресурс оборудования.

В результате решения распределительной задачи для каждого продукта p_i формируется вектор $X_i = \{x_{ij}|j=1,2,\dots,m\}$, задающий распределение объемов x_{ij} продукта p_i между станками набора $S = \{s_j|j=1,2,\dots,m\}$ с учетом минимизации критерия F согласно формуле (1). Если для некоторого решения $X = \{X_i|i=1,2,3,\dots,n\}$ задачи распределения не выполняется одно из ограничений (3), то такое распределение считается нелегальным.

Общая структура представления решений в алгоритме распределения ресурсов на основе роевого интеллекта и генетического поиска

В эвристических алгоритмах роевого интеллекта многомерное пространство поиска населяется роем частиц [5]. Каждая частица представляет некоторое решение, в нашем случае – решение распределительной задачи. Процесс поиска решений заключается в последовательном перемещении частиц в пространстве поиска. Позиция частицы i в пространстве решений в момент времени t (t имеет дискретные значения) определяется вектором $x_i(t)$. По аналогии с эволюционными стратегиями рой можно трактовать как популяцию, а частицу – как индивида (хромосому). Это дает возможность построения гибридной структуры поиска решения, основанной 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. Значение каждого гена откладывается на соответствующей оси пространства решений. В этом случае возникают некоторые требования к структуре хромосомы и значениям генов. Значения генов должны быть дискретными и независимыми друг от друга, т.е. хромосомы должны быть гомологичными.

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.

Разработка структуры хромосомы производилась так, чтобы гены в одних и тех же локусах хромосом являлись гомологичными, так как это упрощает выполнение генетических операторов. Общее решение распределительной задачи представляется в виде совокупности векторов $X = \{X_i | i=1, 2, 3, \dots, n\}$, $X_i = \{x_{ij} | j=1, 2, \dots, m\}$. Хромосома H состоит из частей h_i , каждая из которых соответствует вектору $X_i = \{x_{ij} | j=1, 2, \dots, m\}$, и наоборот, $H = \{h_i | i=1, 2, \dots, n\}$. Каждая часть H_i представляет собой совокупность $(m-1)$ генов g_i , значения которых могут изменяться в пределах диапазона, определяемого параметром b_i :

$$h_i \in H, g_i \in H_i, 0 \leq g_i \leq b_i. \tag{5}$$

Структура хромосомы H , соответствующая совокупности векторов X , имеет вид, представленный на рис. 1. Для удобства введено обозначение $\varphi = m - 1$.

$$H = \bigcup_{i=1}^n h_i.$$

H											
h_1				h_2				h_3			
g_1	g_2	...	g_φ	$g_{\varphi+1}$	$g_{\varphi+2}$...	$g_{2\varphi}$...	$g_{(n-1),(\varphi+1)}$...	$g_{n\varphi}$

Рис. 1. Структура хромосомы

Процесс перехода от хромосомы H к совокупности векторов X осуществляется следующим образом. Сначала в каждой части h_i гены упорядочиваются по возрастанию их значений, т.е. если $g_i \in h_i$ и $g_{i+1} \in h_i$, то $g_{i+1} \geq g_i$. Значения соседних генов g_i и g_{i+1} в пределах каждой части h_i являются опорными точками на отрезке длиной b_i (от 0 до b_i), разбивающими его на интервалы. Длина интервала, определенного как $(g_{i+1} - g_i)$, $g_{i+1} \in h_i$ и $g_i \in h_i$, и есть величина соответствующего элемента x_{ij} вектора X_i .

Пусть для трех видов продукции имеется упорядоченная хромосома, состоящая из трех частей – $H = \{8, 12, 16, 25; 3, 5, 8, 9; 5, 7, 14, 20\}$, при этом $b_1=30, b_2=10, b_3=21, m=5$. Каждая часть хромосомы включает четыре гена ($m-1=4$). Принцип декодирования хромосомы представлен на рис. 2.

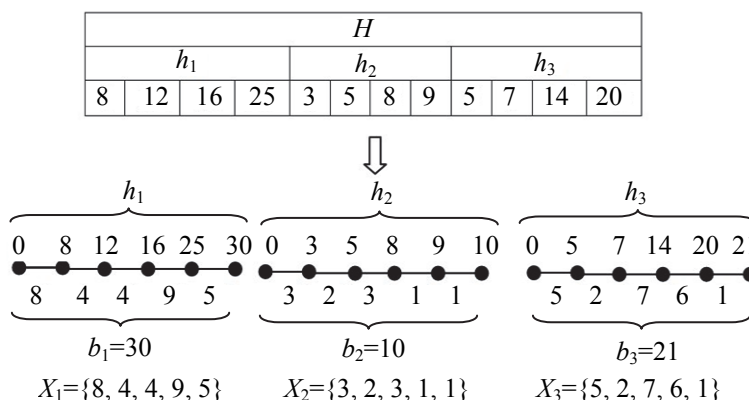


Рис. 2. Принцип декодирования хромосомы

Представленной хромосоме соответствуют три вектора $X_1 = \{8, 4, 4, 9, 5\}$, $X_2 = \{3, 2, 3, 1, 1\}$, $X_3 = \{5, 2, 7, 6, 1\}$, задающие распределение продукции по 5 станкам.

В работе предлагается два способа кодирования и декодирования хромосом.

При первом способе представления решения задачи распределения значения генов определяются только ограничениями типа (5). Процесс перехода от хромосомы H к совокупности векторов X предусматривает предварительное упорядочивание генов в каждой части h_i по возрастанию их значений.

При втором способе представления решения изначально гены в пределах каждой части h_i упорядочены по возрастанию их значений.

Временная сложность декодирования t_d хромосом второго вида, т.е. переход от H к набору X , имеет линейную зависимость $O(L)$, где L – длина хромосомы, $L = m \cdot n$. Для хромосом первого вида предварительно в каждой части $h_i \in H$ осуществляется сортировка, временная сложность которой определяется как $O(m \cdot \log m)$. Тогда временная сложность декодирования всей хромосомы первого типа определяется как $O(m \cdot n \cdot \log m) = O(L \cdot \log m)$.

Для организации генетического поиска формируется исходная популяция особей $\Pi = \{H^u | u = 1, 2, \dots, \theta\}$, где θ – размер популяции. Популяция Π представляет собой репродукционную

группу – совокупность индивидуальностей, любые из которых могут размножаться, выступая в роли родителей.

В работе используется принцип случайного формирования исходной популяции. Для этого в пределах каждой хромосомы H^k в каждой части h_i^u каждый ген $g_i \in h_i^u$ принимает случайное значение в диапазоне $(0 - b_i)$, т.е. от 0 до b_i .

Механизмы генетического поиска

На каждой генерации вначале реализуются операторы кроссинговера, мутации, а затем расширенная популяция подвергается редукции с помощью селективного отбора, т.е. уменьшению до начального объема. Для хромосом первого вида используются два типа операторов кроссинговера – K_1 и K_2 . Кроссинговер K_1 выполняется следующим образом. Пусть имеются две родительские хромосомы H^1 и H^2 . Последовательно, начиная с первого, просматриваются их локусы, и с вероятностью P_{k1} осуществляется обмен генами. В результате получаются две новые хромосомы – H^{*1} и H^{*2} .

Кроссинговер второго типа K_2 выполняется следующим образом. Последовательно, начиная с первой, просматриваются пары областей (h_i^1, h_i^2) ($i=1, 2, \dots, n$) родительской пары хромосом H^1 и H^2 , и с вероятностью P_{k2} происходит обмен текущей пары областей h_i^1 и h_i^2 . В результате появляется новая пара хромосом H^{*1} и H^{*2} .

Мутация для хромосом первого вида реализуется следующим образом. Последовательно просматриваются локусы хромосом, и с вероятностью P_m ген $g_i \in h_i$ в текущем локусе мутирует. В результате мутации ген $g_i \in h_i$ приобретает новое случайное значение в диапазоне $0 \leq g_i \leq b_i$.

Для хромосом второго вида кроссинговер K_1 выполняется следующим образом. Пусть имеются две родительские хромосомы H^1 и H^2 . Последовательно просматриваются локусы хромосом, и с вероятностью P_{k1} осуществляется обмен генами в текущем локусе при условии соблюдения ограничения на кроссинговер. Напомним, что в хромосомах второго вида в каждой части h_i^1 и h_i^2 значения генов упорядочены по возрастанию. Суть ограничения состоит в следующем. Если текущий локус не является ни первым, ни последним в частях h_i^1 и h_i^2 , то обмен генами $g_{1l} \in h_{1l}^1$ и $g_{2l} \in h_{2l}^2$ может быть произведен при соблюдении ограничений, имеющих следующий вид:

$$\begin{aligned} &(g_{2l-1} \leq g_{1l} \leq g_{2l+1}) \text{ и } (g_{1l-1} \leq g_{2l} \leq g_{1l+1}); \\ &(g_{1l} \leq g_{2l+1}) \text{ и } (g_{2l} \leq g_{1l+1}), \text{ если локус } l \text{ является первым в частях } H^1 \text{ и } H^2; \\ &(g_{2l-1} \leq g_{1l}) \text{ и } (g_{1l-1} \leq g_{2l}), \text{ если локус } l \text{ является последним в частях } H^1 \text{ и } H^2. \end{aligned} \quad (6)$$

Если выполняются одно из ограничений (6), то после обмена генами g_{1l} и g_{2l} в новых хромосомах гены будут упорядочены.

Оператор мутации для хромосом второго вида отличается от рассматриваемого выше различными диапазонами возможных значений мутируемого гена, которые регламентируются ограничениями на диапазон.

Если в ходе последовательного просмотра локусов в текущем локусе l выпадает с вероятностью P_m событие «мутация», то ген $g_l \in h_l$ мутирует. При этом он приобретает случайное значение в диапазоне:

$$\begin{aligned} &0 \leq g_l \leq g_{l+1}, \text{ если локус } l - \text{ первый в части } h_i; \\ &g_{l-1} \leq g_l \leq g_{l+1}, \text{ если локус } l \text{ не является ни первым, ни последним в части } h_i; \\ &g_{l-1} \leq g_l \leq b_i, \text{ если локус } l - \text{ последний в части } h_i. \end{aligned} \quad (7)$$

Выполнение ограничений на кроссинговер и на диапазон мутации при выполнении операций кроссинговера и мутации обеспечивает сохранение основного свойства хромосом второго типа, заключающегося в том, что в пределах каждой части H_i гены упорядочены по возрастанию их значений.

Как видно из алгоритмов, реализующих процедуры кроссинговера и мутации, временная сложность операторов кроссинговера t_k и мутации t_m применительно к одной хромосоме имеет линейную зависимость, и оценки временной сложности имеют вид $t_k = O(L)$, $t_m = O(L)$, где L – длина хромосомы.

Временные затраты в пределах поколения складываются из затрат на операторы: кроссинговера t_k , мутации t_m , расчета оценки t_ϕ , декодирования t_d и селекции t_c . Как уже указывалось выше, временные затраты – t_k , t_m и t_ϕ имеют линейную зависимость от L , где $L = n \cdot m$, n – число типов продуктов, m – число станков. Временные затраты оператора селекции t_c на основе «принципа рулетки» имеют линейную зависимость от объема популяции M . Отсюда временные затраты в пределах поколения для хромосом первого вида имеют оценку $O(L \cdot M)$, а для хромосом второго вида – $O(L \cdot \log m \cdot M)$.

Следует отметить, что представленное хромосомой H решение X задачи распределения обеспечивает выполнение плана заказов, так как структура хромосомы сформирована в соответствии с соотношениями (2). Однако хромосома H не учитывает ограничения на ресурсы оборудования, задаваемые соотношениями (3) и (4). Другими словами, хромосома H , формируемая в соответствии с изложенными выше правилами, учитывает ограничения на выполнение плана заказов, но не учитывает ограничения на ресурсы оборудования. В связи с этим каждая хромосома, создаваемая либо на этапе формирования исходной популяции, либо в результате выполнения операторов кроссинговера или мутации, проверяется на выполнение ограничений на ресурсы оборудования, задаваемые соотношениями (3). Если представленное

хромосомой H решение X задачи распределения не соответствует ограничениям на ресурсы, то такая хромосома исключается из рассмотрения.

Очевидно, что число нелегальных хромосом напрямую зависит от соотношения между суммарным объемом V_p ресурсов, необходимым для производства планируемого количества продукции, и суммарным объемом V_s ресурсов оборудования. Исследования показали, что при равенстве этих объемов в процессе работы алгоритма число создаваемых нелегальных хромосом колеблется в пределах 5–7% от числа хромосом в популяции. Дополнительная проверка хромосом на легальность увеличивает время решения в среднем на 5%. При превышении V_s над V_p на 5% число нелегальных хромосом составило менее 1%.

Механизмы решения задачи распределения ресурсов 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 всеми частицами роя, может взаимодействовать со всеми частицами и тяготеет к лучшему решению роя. Процесс поиска решений – итерационный. На каждой итерации каждая частица перемещается в пространстве параметров. Эти частицы никогда не умирают (так как нет селекции), а перемещаются с помощью направленной мутации в новую позицию. В каждый момент времени (на каждой итерации) частицы имеют в этом пространстве некоторое положение $x_i(t)$ и «вектор скорости» $v_i(t)$. Для каждого положения частицы вычисляется соответствующее значение целевой функции. При определении следующего положения частицы учитывается информация о наилучшей частице из числа ее «соседей», а также информация о данной частице на той итерации, когда этой частице соответствовало наилучшее значение целевой функции. На основе этой информации, по определенным правилам, с помощью «вектора скорости» $v_i(t)$ частица меняет свое положение в пространстве поиска и модифицирует «скорость» [14].

В каноническом методе роя частиц [15], использующем непрерывные метрические шкалы, новая позиция частицы i определяется с помощью аналитических выражений как

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

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

$$x_{ik}(t+1) = x_{ik}(t) + v_{ik}(t+1),$$

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

Введем обозначения:

- $f_i(t)$ – текущее значение целевой функции частицы i в позиции $x_i(t)$;
- $f^*(t)$ – лучшее значение целевой функции среди частиц роя в момент времени t , а $x^*(t)$ – позиция с этим значением;
- $x_i^*(t)$ – лучшая позиция частицы i , которую она посещала с начала первой итерации, а $f_i^*(t)$ – значение целевой функции в этой позиции (лучшее значение частицы i).

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 непрерывной метрической (числовой) шкалы используется *дискретная*.

Для того чтобы избежать при описании популяции (роя) путаницы, будем в дальнейшем каждую хромосому, описывающую i -ое решение популяции обозначать как $H^k(t)$. Причем каждая $H^k(t)$ имеет структуру, описанную выше. В нашем случае позиция $x_k(t)$ соответствует решению X^k задачи распределения, задаваемому хромосомой $H^k(t)$, т.е. $x_k(t) = H^k(t)$. Аналогично $x_k^*(t) = H^{k*}(t)$, $x^*(t) = H^*(t)$. Число осей в пространстве решений равно числу генов в хромосоме $H_i(t)$. Точками отсчета на каждой оси k являются значения генов.

В работе 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 аналога скорости $v_k(t+1)$ выступает оператор направленной мутации (ОНМ), суть которого заключается в изменении значения генов в хромосоме $H^k(t)$. Перемещение частицы $x_k(t)$ в новую позицию означает переход от хромосомы $H^k(t)$ к новой – $H^k(t+1)$ с новыми значениями генов g_{kl} , полученными после применения ОНМ.

По аналогии с каноническим методом роя частиц позицию $x^*(t)$ 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. Формирование $x_k^c(t)$ осуществляется путем применения процедуры перемещения к позициям $x(t)$ и $x_k^*(t)$. После определения центра притяжения $x_k^c(t)$ частица $x_k(t)$ с помощью процедуры перемещения перемещается в направлении $x_k^c(t)$ из позиции $x_k(t)$ в позицию $x_k(t+1)$.

В качестве оценки степени близости между двумя позициями $x_k(t)$ и $x_z(t)$ будем использовать величину R_{kz} расстояния между хромосомами $H^k(t)$ и $H^z(t)$:

$$R_{kz} = \sum_{l=1}^n |g_{kl} - g_{zl}|.$$

Целью перемещения хромосомы $H_z(t)$ в направлении хромосомы $H_k(t)$ является сокращение расстояния между ними.

Суть процедуры перемещения, реализуемой ОНМ, заключается в изменении разности между значениями каждой пары генов (g_{kl}, g_{zl}) двух хромосом, $l=1, 2, \dots, nm$.

Перемещение частицы из позиции $H^z(t)$ в позицию $H^z(t+1)$ под воздействием притяжения к позиции $H^k(t)$ осуществляется путем применения ОНМ к $H^z(t)$ следующим образом. Последовательно просматриваются (начиная с первого) локусы хромосом $H^k(t)$ и $H^z(t)$, и сравниваются соответствующие им гены. Если в ходе последовательного просмотра локусов в текущем локусе l выпадает с вероятностью p событие «мутация», то ген $g_{zl}(t) \in H^z(t)$ мутирует.

Мутации выполняются по следующим правилам:

если $(g_{kl}(t) - g_{zl}(t)) = 0$, то $g_{zl}(t+1) = g_{zl}(t)$;

если $(g_{kl}(t) - g_{zl}(t)) < 0$, то $g_{zl}(t+1) = g_{zl}(t) - 1$;

если $(g_{kl}(t) - g_{zl}(t)) > 0$, то $g_{zl}(t+1) = g_{zl}(t) + 1$.

Параметр p зависит от расстояния $R_{kz}(t)$ между позициями $H^k(t)$ и $H^z(t)$, и определяется следующим образом:

$$p = \alpha \cdot R_{kz}(t) / nm,$$

где α – коэффициент, nm – длина хромосомы. Таким образом, чем больше расстояние $R_{kz}(t)$ между позициями $H^k(t)$ и $H^z(t)$, тем больше вероятность, что значение $g_{zl}(t)$ будет изменено.

Пример. Пусть $H^k(t) = \langle 5, 8, 17, 20 \rangle$, $H^z(t) = \langle 4, 10, 15, 22 \rangle$. В соответствии с выражением (7) $R_{kz}(t) = |5 - 4| + |8 - 10| + |17 - 15| + |20 - 22| = 7$.

Примем $p = 1$, тогда в соответствии с правилами направленной мутации

$$H^z(t+1) = \langle 5, 9, 16, 21 \rangle, \quad R_{kz}(t+1) = |5 - 5| + |8 - 9| + |17 - 16| + |20 - 21| = 3.$$

Таким образом, расстояния между позициями H^k и H^z сократилось с 7 до 3.

Локальная цель перемещения частицы H^k – достижение ею позиции с наилучшим значением целевой функции. Глобальная цель роя частиц – формирование решения задачи распределения – минимизация общего количества затрат при изготовлении набора продуктов $P = \{p_i | i = 1, 2, \dots, n\}$ в соответствии с планом на наборе станков $S = \{s_j | j = 1, 2, \dots, m\}$.

Гибридизация роевого интеллекта с генетическим поиском

В гибридных алгоритмах преимущества одного алгоритма могут компенсировать недостатки другого. В связи с этим в настоящее время одним из основных путей повышения эффективности решения задач глобального поиска является разработка гибридных алгоритмов. Существуют различные способы гибридизации поисковых алгоритмов оптимизации. В соответствии с классификацией Ванга [22] выделяют три категории гибридных алгоритмов: вложенные алгоритмы, алгоритмы типа препроцессор/постпроцессор, коалгоритмы.

Предлагается композитная архитектура многоагентной системы бионического поиска для решения общей распределительной задачи на основе роевого интеллекта и генетической эволюции. Рассмотрены три подхода к построению такой архитектуры [20, 21].

Первый и наиболее простой подход к гибридизации заключается в последовательном использовании двух алгоритмов. Сначала 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 популяция хромосом рассматривается как рой частиц. Дальнейший поиск решения осуществляется роевым алгоритмом.

При втором подходе используется высокоуровневая гибридизация вложением, предполагающая сращивание гибридизируемых алгоритмов. Отличительным признаком высокоуровневой гибридизации является то, что объединяемые алгоритмы сохраняют значительную автономию. Метод роя частиц используется в процессе генетического поиска и играет роль, аналогичную генетическим операторам. В этом случае на каждой итерации генетического алгоритма на первом этапе с помощью кроссинговера и мутации осуществляется синтез новых хромосом, а на втором этапе с помощью оператора направленной мутации осуществляется модификация хромосом, что соответствует перемещению роя частиц в новые

позиции. Далее обновленная популяция используется в общей структуре процедуры генетического поиска.

Третий подход является объединением первого и второго подходов. Первые итерации поиска реализует гибридный алгоритм в соответствии со вторым подходом, а завершающие итерации – метод роя частиц.

Оценка временной сложности операторов роя частиц не превышает оценки временной сложности генетических операторов. Оценка временной сложности генетического алгоритма не превышает оценки временной сложности алгоритма роя частиц. В связи с этим общая оценка временной сложности при любом подходе к гибридизации не превышает оценки временной сложности генетического алгоритма и лежит в пределах $O(n^2)$ – $O(n^3)$.

Экспериментальные исследования

Для написания программы был использован язык C++ в среде Microsoft Visual Studio 2010 для операционной системы (ОС) Windows, так как главный упор делался на скорость работы приложения.

Тестирование проводилось на ЭВМ с процессором Intel Core 2 Duo T6600 2200 МГц под управлением ОС Windows 7.

Были проведены исследования рассмотренных алгоритмов на оптимальность и масштабируемость.

Целью первой части исследований являлось определение эффективности интеграции роевого и генетического алгоритмов. Для этого сначала проводились по отдельности исследования роевого и генетического алгоритмов, разработанных авторами. А затем исследовался гибридный алгоритм.

Исследованию подвергались примеры задач распределения, для наборов продуктов размером от 20 до 100 и наборов оборудования размером от 5 до 15. Каждый пример тестировался на 20 независимых прогонах генетическим, роевым и гибридным алгоритмом по отдельности.

При исследовании сходимости алгоритмов для каждого эксперимента запоминался номер генерации, после которой не наблюдалось улучшения оценки. В каждой серии из 20 испытаний определялось 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.

В результате проведенных исследований установлено, что качество решений у гибридного алгоритма на 10–15% лучше качества решений генетического и роевого алгоритмов по отдельности. В среднем программа входит в локальный оптимум на 130 итерации (рис. 3).

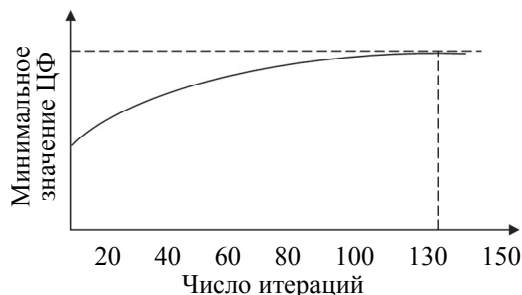


Рис. 3. Зависимость качества гибридного алгоритма от числа итераций.
ЦФ – целевая функция

Эксперименты показали, что увеличение популяции M больше 100 нецелесообразно, так как это не приводит к заметному изменению качества.

Исследования показали, что при равенстве между суммарным объемом V_p ресурсов, необходимых для производства планируемого количества продукции, и объемом V_s ресурсов оборудования, число создаваемых в процессе работы алгоритма нелегальных хромосом колеблется в пределах 5–7% от общего числа хромосом в популяции. Дополнительная проверка хромосом на легальность увеличивает время решения в среднем на 5%. При превышении V_s над V_p на 5% число нелегальных хромосом составило менее 1%.

Вероятность получения глобального оптимума составила 0,94. В среднем запуск программы обеспечивает нахождение решения, отличающегося от оптимального менее чем на 2%.

Временная сложность алгоритма при фиксированных значениях M и количества генераций T лежит в пределах $O(n)$. Общая оценка временной сложности при любом подходе к гибридизации не превышает оценки временной сложности генетического алгоритма и лежит в пределах $O(n^2)$ – $O(n^3)$.

Сравнительный анализ с другими алгоритмами решения РЗ производился на стандартных тестовых примерах (бенчмарках).

В таблице приведены результаты сравнительных экспериментов и их анализ. Предлагаемый гибридный алгоритм GA-PSO сравнивается с PSO [19] и IBA (Improved Bat Algorithm) [23, 24] на benchmarks

(FT06, FT10 и FT20) и (от LA01 до LA07). Эти тесты доступны на веб-сайте OR-Library [21] и <http://mscmga.ms.ic.ac.uk>.

№	Probl	Size $n \times m$	Opt	PSO			IBA			GA-PSO		
				best	Aver	RPE %	best	Aver	RPE %	best	Aver	RPE %
1	Ft06	6×6	55	55	58,9	0	55	56,93	0	55	56,3	0
2	Ft10	10×10	930	1007	1086	8,2795	991	1012,6	6,5591	975	976,57	4,8387
3	Ft20	20×5	1165	1242	1296	6,6094	1177	1188,5	1,0300	1181	1183,7	1,3733
4	La01	10×5	666	681	705	2,2522	666	666	0	666	675,9	0
5	La02	10×5	655	694	729,7	5,9541	668	677,5	1,9847	662	676,966	1,0526
6	La03	10×5	597	633	657,5	6,0301	613	622,6	2,6800	611	613,466	2,3450
7	La04	10×5	590	611	648,1	3,5593	611	611	3,5593	605	607,3	2,5423
8	La05	10×5	593	593	601,1	0	593	593	0	593	593,8	0
9	La06	15×5	926	926	940,2	0	926	926	0	926	927,5	0
10	La07	15×5	890	890	940,1	0	890	890	0	890	890,933	0

Таблица. Результаты сравнительного анализа

Содержимое таблицы включает имя каждой тестовой задачи (Probl), масштаб проблемы (Size), значение самого лучшего известного решения для каждой задачи (Opt), значение наилучшего решения, найденного с использованием сравниваемых алгоритмов (best), среднее значение решения (Aver), процент отклонения по отношению к наиболее известному решению (RPE %). $RPE = (\text{best} - \text{Opt}) / \text{Opt} \times 100$, (RPE) Relative Percentage Error.

Значения параметров Opt, best, Aver отражают суммарные затраты в условных единицах.

Для задач (La05, La06 и La07) все три алгоритма дают наилучшие известные значения ($\text{best} = \text{Opt}$), среднее значение решений Aver у GA-PSO лучше, чем среднее значение у PSO, но незначительно хуже, чем у IBA. Для задачи FT20 GA-PSO по сравнению с PSO дает лучшие результаты по всем показателям; по сравнению с IBA незначительно худший результат по показателю best, но лучший результат по показателю Aver. Для задач (FT10, La02, La03 и La04) GA-PSO дает результаты по всем показателям (best, Aver, RPE) лучше, чем PSO и IBA.

Результаты показывают, что предлагаемый подход дает улучшение качества решения по отношению к другим алгоритмам и является приемлемой альтернативой для решения распределительных задач.

Заключение

На основе сравнительного анализа существующих подходов и методов для решения общей распределительной задачи (ОРЗ) использованы мультиагентные методы интеллектуальной оптимизации, базирующиеся на моделировании адаптивного поведения биологических систем. Предложен подход к решению ОРЗ методами гибридного роевого интеллекта. Композитная архитектура многоагентной системы бионического поиска для решения ОРЗ базируется на основе интеграции роевого интеллекта и генетической эволюции. Связующим звеном такого подхода является структура данных, описывающая в виде хромосомы решение задачи. Рассмотренные в работе новые принципы и способы кодирования и декодирования хромосом для представления решения ОРЗ исключают некорректные решения, отличаются простотой и линейными оценками временной и пространственной сложности, что упрощает использование генетических операторов и позволяет использовать модификации генетических операторов, близких к естественным.

Многохромосомные представления решений позволили создать иерархические структуры генетических операторов, что дает возможность организовать целенаправленный поиск.

Предложена модифицированная парадигма метода роя частиц. В отличие от стандартной парадигмы метода роя частиц вместо непрерывной метрической (числовой) шкалы используется дискретная. В качестве скорости $v_k(t+1)$ выступает оператор направленной мутации, суть которого заключается в изменении значения генов в хромосоме. Перемещение частицы $x_k(t)$ в новую позицию означает переход соответствующей хромосомы посредством предложенного оператора направленной мутации в новое состояние с новыми значениями генов.

Гибридный подход, способы и методы представления ОРЗ в виде эволюционных процессов, основанных на интеграции моделей адаптивного поведения биологических систем, композитных архитектур нахождения решений, позволяют работать с задачами большой размерности и получать качественные результаты за приемлемое время.

Литература

References

1. Золотарев А.А. Методы оптимизации распределительных процессов. М.: Инфра-Инженерия, 2014. 160 с.
2. Brucker P. *Scheduling Algorithms*. 5th ed. Springer, 2007. 379 p.
3. Нейдорф Р.А., Кобак В.Г., Красный Д.Г. Точное решение неоднородной распределительной задачи модификацией алгоритма Алексева // Известия вузов. Северо-Кавказский регион. Серия: Технические науки. 2008. №1. С. 17–21.
4. Серая О.В. Распределительная задача линейного программирования // Системы обработки информации. 2013. № 2 (109). С. 167–170.
5. Акулич И.Л. Математическое программирование в примерах и задачах. 3-е изд. СПб.: Лань, 2011. 352 с.
6. Костюкевич В.М., Давыдов Г.А., Хотина И.Г. Оптимальное распределение ресурсов с использованием динамического программирования // Resources and Technology. 2006. Т. 7. С. 49–51.
7. Lorigeon T., Billaut J.C., Bouquar J.L. A dynamic programming algorithm for scheduling jobs in a two-machine open shop with an availability constraint // Journal of the Operational Research Society. 2002. V. 53. N 11. P. 1239–1246. doi: 10.1057/palgrave.jors.2601421
8. Прилуцкий М.Х., Кумагина Е.А. Метод ветвей и границ решения задачи многоресурсного сетевого планирования // Системы управления и информационные технологии. 2014. № 2(56). С. 48–51.
9. Artigues C., Feillet D. A branch and bound method for the job shop problem with sequence dependent setup times // Annals of Operations Research. 2008. V. 159. N 1. P. 135–159. doi: 10.1007/s10479-007-0283-0
10. Пантелеев А.В., Скавинская Д.В., Алёшина Е.А. Метаэвристические алгоритмы поиска оптимального программного управления. М.: ИНФРА-М, 2017. 396 с.
11. Богданов С.И., Петров А.В. Эффективные процессы распределения товаров: концепции, модели, методы реализации. Екатеринбург: УрГЭУ, 2008. 234 с.
12. Золотарев А.А., Венцов Н.Н., Агибалов О.И., Деева А.С. Оптимизация распределительных процессов на основе аналитических методов и эвристических алгоритмов // Вестник науки и образования Северо-Запада России. 2016. Т. 2. № 1. С. 74–82.
13. Карпенко А.П. Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой. М.: МГТУ им. Н.Э. Баумана, 2014. 448 с.
14. Курейчик В.М., Лебедев Б.К., Лебедев О.Б. Поисковая адаптация. Теория и практика. М.: Физматлит, 2006. 272 с.
15. Verma S., Jain M., Choudhary D. Solving the job-shop scheduling problem by using genetic algorithm // International Journal of Computational Science and Mathematics. 2011. V. 3. N 1. P. 93–98.
16. Moon I., Lee S., Bae H. Genetic algorithms for job shop scheduling problems with alternative routings // International Journal of Production Research. 2008. V. 46. N 10. P. 2695–2705. doi: 10.1080/00207540701244820
17. Liang S., Cheng X., Liang Y. Solving job shop scheduling problem using genetic algorithm with penalty function // International Journal of Intelligent Information Processing. 2010. V. 1. N 2. P. 65–77. doi: 10.4156/ijiip.vol1.issue2.7
18. Nickabadi A., Ebadzadeh M.M., Safabakhsh R. A novel particle swarm optimization algorithm with adaptive inertia weight // Applied Soft Computing Journal. 2011. V. 11. N 4. P. 3658–3670. doi: 10.1016/j.asoc.2011.01.037
19. Sha D.Y., Hsu C.Y. A hybrid particle swarm optimization for job shop scheduling problem // Computers and Industrial Engineering. 2006. V. 51. N 4. P. 791–808. doi: 10.1016/j.cie.2006.09.002
20. Лебедев Б.К., Лебедев В.Б. Покрытие на основе метода роя частиц // Материалы Международной конференции по нейрокибернетике. 2011. Т. 2. С. 417–425.
21. Лебедев Б.К., Лебедев В.Б. Планирование на основе роевого интеллекта и генетической эволюции // Известия ЮФУ. Технические науки. 2009. №4. С. 25–33.
22. Blum Ch., Aguilera M.J.B., Roli A., Sampels M. Hybrid
1. Zolotarev A.A. *Methods for Optimizing Distribution Processes*. Moscow, Infra-Inzheneriya Publ., 2014, 160 p. (In Russian)
2. Brucker P. *Scheduling Algorithms*. 5th ed. Springer, 2007, 379 p.
3. Neydorf R.A., Kobak V.G., Krasniy D.G. The exact decision non-uniform of problems by modification of algorithm Alekseev. *University News. North-Caucasian Region. Technical Sciences Series*, 2008, no. 1, pp. 17–21. (In Russian)
4. Seraya O.V. Distribution problem of linear programming. *Sistemy Obrabotki Informatsii*, 2013, no. 2, pp. 167–170. (In Russian)
5. Akulich I.L. *Mathematical Programming in Examples and Tasks*. 3rd ed. St. Petersburg, Lan' Publ., 2011, 352 p. (In Russian)
6. Kostyukevich V.M., Davydov G.A., Khotina I.G. Resources optimal distribution on the basis of dynamic programming. *Resources and Technology*, 2006, vol. 7, pp. 49–51. (In Russian)
7. Lorigeon T., Billaut J.C., Bouquar J.L. A dynamic programming algorithm for scheduling jobs in a two-machine open shop with an availability constraint. *Journal of the Operational Research Society*, 2002, vol. 53, no. 11, pp. 1239–1246. doi: 10.1057/palgrave.jors.2601421
8. Prilutskii M.Kh., Kumagina E.A. Method of branches and boundaries for solving the problem of multi-resource network planning. *Sistemy Upravleniya i Informatsionnye Tekhnologii*, 2014, no. 2, pp. 48–51. (In Russian)
9. Artigues C., Feillet D. A branch and bound method for the job shop problem with sequence dependent setup times. *Annals of Operations Research*, 2008, vol. 159, no. 1, pp. 135–159. doi: 10.1007/s10479-007-0283-0
10. Panteleev A.V., Skavinskaya D.V., Aleshina E.A. *Metaheuristic Algorithms of Search of Optimum Program Control*. Moscow, Infra-M Publ., 2017, 396 p. (In Russian)
11. Bogdanov S.I. *Effective Processes of Product Distribution: Concepts, Models, Implementation Methods*. Ekaterinburg, UrSEU Publ., 2008, 234 p. (In Russian)
12. Zolotarev A.A., Ventsov N.N., Agibalov O.I., Deeva A.S. Distribution process optimization based on analytical methods and heuristic algorithms. *Journal of Science and Education of North-West Russia*, 2016, vol. 2, no. 1, pp. 74–82. (In Russian)
13. Karpenko A.P. *Modern Algorithms of Search Optimization. Algorithms Inspired by Nature*. Moscow, Bauman MSTU Publ., 2014, 448 p. (In Russian)
14. Kureichik V.M., Lebedev B.K., Lebedev O.B. *Search Adaptation. Theory and Practice*. Moscow, Fizmatlit Publ., 2006, 272 p. (In Russian)
15. Verma S., Jain M., Choudhary D. Solving the job-shop scheduling problem by using genetic algorithm. *International Journal of Computational Science and Mathematics*, 2011, vol. 3, no. 1, pp. 93–98.
16. Moon I., Lee S., Bae H. Genetic algorithms for job shop scheduling problems with alternative routings. *International Journal of Production Research*, 2008, vol. 46, no. 10, pp. 2695–2705. doi: 10.1080/00207540701244820
17. Liang S., Cheng X., Liang Y. Solving job shop scheduling problem using genetic algorithm with penalty function. *International Journal of Intelligent Information Processing*, 2010, vol. 1, no. 2, pp. 65–77. doi: 10.4156/ijiip.vol1.issue2.7
18. Nickabadi A., Ebadzadeh M.M., Safabakhsh R. A novel particle swarm optimization algorithm with adaptive inertia weight. *Applied Soft Computing Journal*, 2011, vol. 11, no. 4, pp. 3658–3670. doi: 10.1016/j.asoc.2011.01.037
19. Sha D.Y., Hsu C.Y. A hybrid particle swarm optimization for job shop scheduling problem. *Computers and Industrial Engineering*, 2006, vol. 51, no. 4, pp. 791–808. doi: 10.1016/j.cie.2006.09.002
20. Lebedev B.K., Lebedev V.B. Coverage based on the particle swarm method. *Proc. Int. Conf. on Neurocybernetics*, 2011, vol. 2, pp. 417–425. (In Russian)
21. Lebedev B.K., Lebedev V.B. Planning on a basis swarm

- Metaheuristics: An Emerging Approach to Optimization. Springer, 2008. 299 p. doi: 10.1007/978-3-540-78295-7
23. Hegazy Z., Mahmoud El-S., Naglaa R.S., Heba S. A novel improved bat algorithm for job shop scheduling problem // International Journal of Computer Applications. 2017. V. 164. N 5. P. 24–30. doi: 10.5120/ijca2017913627
24. Beasley J.E. Distributing test problems by electronic mail // Journal of the Operational Research Society. 1990. V. 41. N 11. P. 1069–1072. doi: 10.1057/jors.1990.166
- intelligence and genetic evolution. *Izvestiya SFedU. Engineering Sciences*, 2009, no. 4, pp. 25–33. (In Russian)
22. Blum Ch., Aguilera M.J.B., Roli A., Sampels M. *Hybrid Metaheuristics: An Emerging Approach to Optimization*. Springer, 2008, 299 p. doi: 10.1007/978-3-540-78295-7
23. Hegazy Z., Mahmoud El-S., Naglaa R.S., Heba S. A novel improved bat algorithm for job shop scheduling problem. *International Journal of Computer Applications*, 2017, vol. 164, no. 5, pp. 24–30. doi: 10.5120/ijca2017913627
24. Beasley J.E. Distributing test problems by electronic mail. *Journal of the Operational Research Society*, 1990, vol. 41, no. 11, pp. 1069–1072. doi: 10.1057/jors.1990.166

Авторы

Лебедев Борис Константинович – доктор технических наук, профессор, профессор, Южный федеральный университет, Ростов-на-Дону, 344006, Российская Федерация, lebedev.b.k@gmail.com

Лебедев Олег Борисович – кандидат технических наук, доцент, доцент, Южный федеральный университет, Ростов-на-Дону, 344006, Российская Федерация, lebedev.ob@mail.ru

Лебедева Елена Михайловна – аспирант, Южный федеральный университет, Ростов-на-Дону, 344006, Российская Федерация, lebedeva.el.m@mail.ru

Authors

Boris K. Lebedev – D.Sc., Full Professor, South Federal University, Rostov-on-Don, 344006, Russian Federation, lebedev.b.k@gmail.com

Oleg B. Lebedev – PhD, Associate Professor, Associate Professor, South Federal University, Rostov-on-Don, 344006, Russian Federation, lebedev.ob@mail.ru

Elena M. Lebedeva – postgraduate, South Federal University, Rostov-on-Don, 344006, Russian Federation, lebedeva.el.m@mail.ru