Bova Victoria Victorovna – Southern Federal University; e-mail: vvbova@sfedu.ru; Taganrog, Russia; phone: +78634371651; the department of computer aided design; associate professor.

Kravchenko Yury Alekseevich – e-mail: yakravchenko@sfedu.ru; the department of computer aided design; associate professor.

Rodzin Sergey Ivanovich – e-mail: srodzin@sfedu.ru; phone: +78634371673; the department of software engineering; professor.

УДК 004.896 DOI 10.18522/2311-3103-2022-4-143-157

Б.К. Лебедев, О.Б. Лебедев, Е.О. Лебедева

ЭВОЛЮЦИОННЫЙ ПОПУЛЯЦИОННЫЙ МЕТОД РЕШЕНИЯ ТРАНСПОРТНОЙ ЗАДАЧИ*

Рассматривается эволюционный популяционный метод решения транспортной задачи на основе метаэвристики кристаллизации россыпи альтернатив. Исследуется закрытая (или сбалансированная) модель транспортной задачи: сумма груза у поставщиков равно общей сумме потребностей в пунктах назначения. Цель оптимизации – минимизация стоимости (достижение минимума затрат на перевозку) или расстояний и критерий времени (затрачивается минимум времени на перевозку). В основу метаэвристики кристаллизации россыпи альтернатив положена стратегия, основанная на запоминании и повторении прошлых успехов. Стратегия делает упор на «коллективную память», под которой подразумевается любой вид информации, которая отражает прошлую историю развития и хранится независимо от индивидуумов. В качестве кода решения транспортной задачи рассматривается упорядоченная последовательность D_k маршрутов. Объектами являются маршруты, альтернативами – множество позиций P в списке, где n_p – число позиций в списке $D_к$. Множество объектов $D_к$ соответствует множеству всех маршрутов. Множество альтернативных состояний P объекта соответствует множеству альтернативных вариантов размещения объекта списке $D_к$. Работа популяционного эволюционного алгоритма кристаллизации россыпи альтернатив опирается на коллективную эволюционную память, называемую россыпью альтернатив. Под россыпью альтернатив решения в работе называется структура данных, используемая в качестве коллективной эволюционной памяти, несущая информацию о решении, включающую сведения о реализованных альтернативах агентов в данном решении и о полезности решения. Разработан конструктивный алгоритм формирования опорного плана путем декодирования списка $D_к$. На каждом шаге t решается задача выбора очередного в последовательности $D_к$ маршрута и определения количества груза, перевозимого из пункта отправления A_i в пункт назначения B_j по этому маршруту. Разработанный алгоритм является популяционным, реализующим стратегию случайного направленного поиска. Каждый агент является кодом некоторого решения транспортной задачи. На первом этапе каждой итерации l конструктивным алгоритмом на базе интегральной россыпи альтернатив формируется n_k кодов решений D_k.Формирование каждого кода решения D_k выполняется последовательно по шагам путем последовательного выбора объекта и позиции. Для построенного кода решения D_k рассчитывается оценка решения ξ_k и оценка полезности δ_k. Формируется индивидуальная россыпь альтернатив R_k и переход к построению следующего кода решения. На втором этапе итерации производится суммирования интегральной россыпи альтернатив, сформированной на предыдущих итерациях от l до (l-1), со всеми индивидуальными россыпями альтернатив, сформированных на итерации l. На третьем этапе итерации l производится снижение всех интегральных оценок полезности $r^*_{\alpha\beta}$ интегральной россыпи альтернатив $R^*(l)$ на величину δ^*. Алгоритм решения транспортной задачи был реализован на языке С++ в среде Windows. Сравнение значений критерия, на тестовых примерах, с

* Работа выполнена при финансовой поддержке гранта РФФИ № 20–07–00260 А.

известным оптимумом показало, что у 90% примеров полученное решение было оптимальным, у 2% примеров решения были на 5% хуже, а у 8% примеров решения отличались менее, чем на 2%. Временная сложность алгоритма, полученная экспериментальным путем, лежит в пределах $O(n^2)$.

Транспортная задача; метаэвристика; кристаллизация россыпи альтернатив; оптимизация; популяционный алгоритм; коллективная память; агент; направленный поиск.

B.K. Lebedev, O.B. Lebedev, E.O. Lebedeva

EVOLUTIONARY POPULATION METHOD FOR SOLVING THE TRANSPORT PROBLEM

The paper considers an evolutionary population method for solving a transport problem based on the metaheuristics of crystallization of a placer of alternatives. We study a closed (or balanced) model of the transport problem: the amount of cargo from suppliers is equal to the total amount of needs at destinations. The goal of optimization is to minimize the cost (achieving a minimum of transportation costs) or distances and the criterion of time (a minimum of time is spent on transportation). The metaheuristics of the crystallization of a placer of alternatives is based on a strategy based on remembering and repeating past successes. The strategy emphasizes «collective memory», which refers to any kind of information that reflects the past history of development and is stored independently of individuals. An ordered sequence D_k of routes is considered as a code for solving the transport problem. The objects are routes, the alternatives are the set of positions P in the list, where n_p is the number of positions in the list D_k. The set of objects D_k corresponds to the set of all routes. The set of alternative states P of the object corresponds to the set of alternative options for placing the object in the list D_k. The operation of the population evolutionary algorithm for the crystallization of a placer of alternatives is based on a collective evolutionary memory called a placer of alternatives. A scattering of solution alternatives is a data structure used as a collective evolutionary memory that carries information about the solution, including information about the realized alternatives of agents in this solution and about the usefulness of the solution. A constructive algorithm for the formation of a reference plan by decoding the list D_k has been developed. At each step t, the problem of choosing the next route in the sequence D_k and determining the amount of cargo transported from the point of departure A_i to the point of destination B_j along this route is solved. The developed algorithm is population-based, implementing the strategy of random directed search. Each agent is a code for some solution of the transport problem. At the first stage of each iteration l, a constructive algorithm based on the integral placer of alternatives generates n_k decision codes D_k. The formation of each decision code D_k is performed sequentially in steps by sequentially selecting an object and position. For the constructed solution code D_k, the solution estimate ξ_k and the utility estimate δ_k are calculated. An individual scattering of alternatives R_k is formed and a transition to the construction of the next solution code is formed. At the second stage of the iteration, the integral placer of alternatives formed at previous iterations from l to (l-1) is summed with all individual placers of alternatives formed at iteration l. At the third stage of iteration l, all integral utility estimates $r^*_{\alpha\beta}$ of the integral placer of alternatives $R^*(l)$ are reduced by δ^*. The algorithm for solving the transport problem was implemented in C++ in the Windows environment. Comparison of the values of the criterion, on test examples, with a known optimum showed that in 90% of the examples the solution obtained was optimal, in 2% of the examples the solutions were 5% worse, and in 8% of the examples the solutions differed by less than 2%. The time complexity of the algorithm, obtained experimentally, lies within $O(n^2)$.

Transport problem; metaheuristics; crystallization of a placer of alternatives; optimization; population algorithm; collective memory; agent; directed search.

Введение. Транспортная задача (классическая) – задача об оптимальном плане перевозок однородного продукта из однородных пунктов наличия в однородные пункты потребления на однородных транспортных средствах (предопределённом количестве) со статичными данными и линеарном подходе (это основные условия задачи) [1, 2].

Распространенность в приложениях задач транспортного типа оправдывает неослабевающее внимание к ним. Транспортная задача является по теории_сложности вычислений *NP*-трудной и входит в класс сложности_*NP*. Транспортные задачи относятся к задачам линейного программирования и могут решаться симплекс-методом.

Когда суммарный объём предложений (грузов, имеющихся в пунктах отправления) не равен общему объёму спроса на товары (грузы), запрашиваемые пунктами потребления, транспортная задача называется несбалансированной.

Для того чтобы решить транспортную задачу используется множество методов [4, 5]. В зависимости от особенностей задачи используются метод потенциалов, дифференциальных рент (для поиска оптимального плана), метод северо-западного угла, метод наименьшего элемента (для поиска опорного плана) и т.д. Одним из подходов к получению качественного решения задачи предполагает итерационное улучшение плана перевозок. Метод содержит три последовательных этапа [4, 5]:

1. Формирование опорного плана.
2. Проверка опорного плана на оптимальность.
3. Переход к новому опорному плану, если предыдущий не оптимален.

На первом этапе решается задача нахождения опорного плана. Опорным решением транспортной задачи называется любое допустимое решение, для которого векторы условий, соответствующие положительным координатам, линейно независимы.

После построения опорного плана перевозок нужно применить один из алгоритмов его улучшения, приближения к оптимальному результату.

Альтернативным походом к решению этих задач является применение методов, инспирированными искусственными или естественными системами, способными за приемлемое время найти оптимум в пространстве поиска решений большой размерности [6–8].

Среди них особенно активно развиваются поведенческие методы роевого интеллекта (Swarm Intelligence) [6, 9]. Базовыми идеями поведенческих методов являются: децентрализованность, взаимодействие агентов, коллективная адаптация, простота поведения агентов. Являясь по своей сути итерационными, алгоритмы на основе моделирования естественных процессов отличаются от обычных итерационных процедур «слепого поиска». В основе работы этих алгоритмов лежат метаэвристики поиска в пространстве состояний (решений). Основными свойствами алгоритмов на основе методов метаэвристической оптимизации являются:

♦ стохастичность, то есть случайность поиска, за счёт чего исключается возможность зацикливания в локальном оптимуме;

♦ мультиагентность;

♦ скорость нахождения оптимального решения выше, чем у традиционных методов [9].

В работе наряду с метаэвристиками, на которых построены роевые алгоритмы [10–12], используется метаэвристика, имеющая тенденцию к использованию альтернатив (вариантов компонентов) из наилучших найденных решений [13, 14]. В процессе эволюционной коллективной адаптации методами дискриминантного анализа формируются оценки приспособленности альтернатив. Приспособленность альтернатив рассматривается как вероятность ее использования в формируемом решении. Совокупность данных об альтернативах и их оценках составляет россыпь альтернатив [15,16]. Дискриминантный анализ альтернатив в процессе эволюционной коллективной адаптации назван по аналогии с процессами вычленения объектов (формирования кристаллов) кристаллизацией. Другими словами, в процессе эволюционной коллективной адаптации производится вычленение из множества вариантов наиболее приспособленных альтернатив. Отсюда название метода оптимизации – метод кристаллизации россыпи альтернатив (КРА, Crystallization of alternatives field (CAF)) [13]. В работе рассматривается эволюционный популяционный метод решения транспортной задачи на основе метаэвристики кристаллизации россыпи альтернатив [13, 14].

1. Математическая постановка транспортной задачи. Общая постановка транспортной задачи заключается в определении оптимального плана перевозок некоторого однородного груза из пунктов отправления $A=\{A_i|i=1, 2, ..., m\}$ в пункты назначения $B=\{B_j|j=1, 2, ..., n\}$. Критерием оптимальности является минимальная стоимость перевозки или минимальное время доставки груза.

Рассмотрим транспортную задачу, где в качестве критерия оптимальности взята минимальная стоимость перевозок всего груза [2, 4].

Задается список $D=\{d_{ij}|i=1,\ldots,m,\ j=1,\ldots n\}$ маршрутов между множеством пунктов отправления A и множеством пунктов назначения B. Обозначим через c_{ij} тарифы перевозки единицы груза из пункта отправления i в пункт назначения j. Обозначим через a_i запасы груза i-м пункте отправления, а через b_j потребности груза j-м пункте назначения, а через x_{ij} количество единиц груза переводимого из пункта отправления i в пункт назначения j.

Тогда математическая модель транспортной задачи состоит в определении минимального значения функции [1, 2]

$$F=\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}\, x_{ij}, \tag{1}$$

при следующих условиях:

$$\sum_{i=1}^{m} x_{ij}=b_j \quad (j=1,\ldots,n), \tag{2}$$

$$\sum_{j=1}^{n} x_{ij}=a_i \quad (i=1,\ldots,m), \tag{3}$$

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

Поскольку удовлетворяются условия (1)−(4), то обеспечивается доставка необходимого количества груза в каждый из пунктов назначения, вывоз груза из всех пунктов отправления, а также исключаются обратные перевозки.

В работе исследуется закрытая (или сбалансированная) модель транспортной задачи: сумма груза у поставщиков равно общей сумме потребностей в пунктах назначения:

$$\sum_{i=1}^{m} a_i=\sum_{j=1}^{n} b_j \tag{5}$$

В задаче требуется найти такой план перевозок, т.е. такие $x_{ij}\geq 0$, чтобы суммарная стоимость перевозок была минимальной, и выполнялись все ограничения.

Исходные данные транспортной задачи представлены в виде табл. 1.

Таблица 1

Исходные данные транспортной задачи

Пункты отправления	Пункты назначения				Запасы
	B_1	B_2	B_3	B_3	
A_1	c_{11} x_{11}	c_{12} x_{12}	c_{13} x_{13}	c_{14} x_{14}	a_1
A_2	c_{21} x_{21}	c_{22} x_{22}	c_{23} x_{23}	c_{24} x_{24}	a_2
A_3	c_{31} x_{31}	c_{32} x_{32}	c_{33} x_{33}	c_{34} x_{34}	a_3
Потребности	b_1	b_2	b_3	b_4	

Содержательная постановка задачи.

Однородный продукт, сосредоточенный в m пунктах отправления в количествах $a_1, a_2, \ldots a_m$ единиц соответственно, необходимо доставить в каждый из n пунктов назначения в количествах $b_1, b_2, \ldots b_n$ единиц соответственно.

Стоимость (расстояние) перевозки единицы продукта из i-го пункта отправления в j-й пункт назначения равна c_{ij} (стоимость доставки) и известна для каждого маршрута d_{ij}. Пусть x_{ij} – количество продукта, перевозимого из i-го пункта отправления в j-й пункт назначения.

Задача заключается в определении таких величин x_{ij} для всех маршрутов, при которых суммарная стоимость или расстояние F перевозок были бы минимальными.

Обозначения:

m – количество пунктов отправления (поставщиков);

i – номер поставщика;

n – количество пунктов назначения (потребителей);

j – номер потребителя;

a_i – исходный объем однородного груза i-го поставщика (запасы);

b_i – исходный объем однородного груза, требуемого j-ому потребителю (спрос);

c_{ij} – стоимость доставки единицы груза i-го поставщика j-ому потребителю;

x_{ij} – количество груза, доставляемое от i-го поставщика к j-му потребителю;

D − набор маршрутов между множеством пунктов отправления A и множеством пунктов назначения B;

d_{ij} − маршрут между пунктом отправления A_i и пунктом назначения B_j;

C – общие затраты на перевозки.

2. Разработка алгоритма решения транспортной задачи методом кристаллизации россыпи альтернатив. В основу метаэвристики кристаллизация россыпи альтернатив (КРА) положена стратегия, основанная на запоминании и повторении прошлых успехов. Стратегия делает упор на «коллективную память», под которой подразумевается любой вид информации, которая отражает прошлую историю развития и хранится независимо от индивидуумов.

В работе рассматривается эволюционный популяционный метод решения транспортной задачи на основе метаэвристики кристаллизации россыпи альтернатив.

В работе используется кодированное представление опорного плана. В качестве кода решения Q_k рассматривается упорядоченная последовательность маршрутов − $D_к$=<d_{kij}|i=1,…,m, $j=1,\ldots n$>. Возможное число маршрутов $R=mn$. Существенную роль в общем процессе нахождения решения играет декодер, осуществляющий трансформацию кода от списка к решению. Важной характеристикой декодера является его способность получить оптимальное решение по заранее известному оптимальному коду (приоритетному списку), т.е. способность правильно декодировать.

Формирование опорного плана путем декодирования списка $D_к$ осуществляется путем последовательного формирования частичных планов перевозок маршрутами и их интеграции в общий план.

На каждом шаге t решается задача выбора очередного в последовательности $D_к$ маршрута d_{kij} и определения количества $x_{ij}(t)$ груза, перевозимого из пункта отправления A_i в пункт назначения B_j по этому маршруту.

Введем табл. 2 для отражения текущих значений остаточных запасов $a_i(t)$ в пунктах отправления, текущих потребностей $b_j(t)$ в пунктах назначения и количества перевезенного груза $x_{ij}(t)$ после выполнения очередного шага t.

Таблица 2

Остаточные запасы

t	$a_i(t)$	$b_j(t)$	$x_{ij}(t)$
0	$a_i(0)=a_i$	$b_j(0)=0$	0
1	$a_i(1)$	$b_j(1)$	$x_{ij}(1)$
2	…	…	…
…	…	…	…
mn	…	…	…

Отметим, в процессе поиска плана параметры $a_i(t)$ уменьшаются, а параметры $b_j(t)$ растут.

Маршрут груза d_{ij} считается реализованным, если хотя бы один из параметров $a_i(t)=0$ *(запас)* или $b_j(t)$ *(потребность)*$=b_j$.

На шаге t выбирается t*-й по списку* неиспользованный маршрут $d_{ij}=(A_i, B_j)$ с параметрами $(a_i(t), b_j(t), x_{ij}(t))$.

Возможны 3 случая:

$a_i(t) = b_j(t)$;

$a_i(t) > b_j(t)$;

$a_i(t) < b_j(t)$.

Рассмотрим тариф c_{ij}.

а) Если $a_i(t)=b_j(t)$, то $x_{ij}(t+1)=x_{ij}(t)+a_i(t)$, $a_i(t+1)=a_i(t)-a_i(t)=0$, $b_j(t+1)=b_j(t)-b_j(t)=0$. В этом случае остаточные запасы $a_i(t)$ пункта A_i полностью исчерпаны. Потребности пункта B_i полностью удовлетворены, поэтому все маршруты, у которых пунктом отправления является пункт A_i, а пунктом назначения является пункт B_j, исключаются из дальнейшего рассмотрения.

Если $a_i(t)>b_j(t)$, то $x_{ij}(t+1)=x_{ij}(t)+b_j(t)$, $a_i(t+1)=a_i(t)-b_j(t)$, $b_j(t+1)=b_j(t)-b_j(t)=0$. Потребности пункта B_i полностью удовлетворены, поэтому все маршруты, у которых пунктом назначения является пункт B_j, из дальнейшего рассмотрения исключаются.

Если $a_i(t)<b_j(t)$, то $x_{ij}=a_i(t)$. В этом случае остаточные запасы $a_i(t)$ пункта A_i полностью исчерпаны $a_i(t+1)=a_i(t)-a_i(t)=0$. Поэтому все тарифы и соответствующие им маршруты, у которых пунктом отправления является пункт A_i, из дальнейшего рассмотрения исключаются, а в пункте B_j потребность $b_j(t)$, уменьшится на $a_i(t)$, т.е. $b_i(t+1)=b_i(t)-a_i(t)$.

Рассмотрим метод декодирования на конкретном примере.

Пример 1. На три базы A_1, A_2, A_3 поступил очередной груз в количествах равных 50, 40, 30 единиц. Этот груз требуется перевезти в четыре пункта назначения B_1, B_2, B_3, B_4 в количествах 42, 33, 25, 20.

Решение. Запишем все данные в таблицу условий:

Таблица 3

Таблица условий

Пункты отправления	Пункты назначения				Запасы
	B_1	B_2	B_3	B_4	
A_1	c_{11} - 4	c_{12} - 3	c_{13} - 2	c_{14} - 1	a_1 - 50
A_2	c_{21} - 3	c_{22} - 2	c_{23} - 4	c_{24} - 1	a_2 - 40
A_3	c_{31} - 2	c_{32} - 4	c_{33} - 1	c_{34} - 3	a_3 - 30
Потребности	b_1 - 42	b_2 - 33	b_3 - 25	b_4 - 20	

Найти опорный план перевозок данной транспортной задачи, решение которой представлено в виде кода D=<d_{14}, d_{24}, d_{34}, d_{33}, d_{23}, d_{13}, d_{22}, d_{12}, d_{32}, d_{21}, d_{31}, d_{11} >.

Число пунктов отправления m=3, а число пунктов назначения n=4.

Наличие груза у поставщиков равно: $\sum A_i$=50+40+30=120.

Общая потребность в грузе в пунктах назначения равна: $\sum B_j$=42+33+25+20=120.

$\sum A_i=\sum B_j$. Модель транспортной задачи является закрытой (сбалансированной). Следовательно, она разрешима. Отметим, что число маршрутов, используемых для перевозок в построенном плане может оказаться меньше числа маршрутов в коде.

На шаге *t*=0 формируются начальные значения параметров:

$a_1(0)$=50, $a_2(0)$=40, $a_3(0)$=30.

$b_1(0)$=42, $b_2(0)$=33, $b_3(0)$=25, $b_4(0)$=20.

Для всех маршрутов $\boldsymbol{x_{ij}(0)=0}$.

Задается код решения*:*

$D(0)$=<d_{14}, d_{24}, d_{34}, d_{33}, d_{23}, d_{13}, d_{22}, d_{12}, d_{32}, d_{21}, d_{31}, d_{11} >.

На шаге *t*=1 из списка $D(0)$ выбирается первый элемент – маршрут d_{14}: $A_1 \rightarrow B_4$, *с* тарифом c_{14}=1:

$a_1(0)$=50, $b_4(0)$=20. Проверяем соотношение параметров $a_1(0)$ и $b_4(0)$:

$a_1(0)>b_4(0)$, ситуация соответствует пункту (а) и $a_1(0)$ полностью покрывает $b_4(0)$.

Отсюда $a_1(1)=a_1(0)-b_4(0)$=50-20=30, $b_4(1)=0$. $\boldsymbol{x_{14}(1)=b_4(0)=20}$.

Потребности пункта B_4 полностью удовлетворены, поэтому маршруты, у которых пунктом назначения является пункт B_4, из дальнейшего рассмотрения исключаются, а в пункте A_1 запас $a_1(0)$ уменьшится на $b_4(0)$. Трансформированный код после первого шага примет вид:

$D(1)$=<d_{33}, d_{23}, d_{13}, d_{22}, d_{12}, d_{32}, d_{21}, d_{31}, d_{11}>.

$a_1(1)$=30, $a_2(1)$=40, $a_3(1)$=30.

$b_1(1)$=42, $b_2(1)$=33, $b_3(1)$=25, $b_4(1)$=0.

$\boldsymbol{x_{14}(1)=20}$.

На шаге *t*=2 в $D(1)$ выбирается маршрут d_{33}: $A_3 \rightarrow B_3$:

$a_3(1)$=30, $b_3(1)$=25, $a_3>b_3$ ситуация соответствует пункту (а) и $a_3(1)$ полностью покрывает $b_3(1)$.

Отсюда в пункте A_3 запас $a_3(1)$ уменьшится на $b_3(1)$, т.е. $a_3(2)=a_3(1)-b_3(1)=30-25=5$, $b_3(2)=0$. $\boldsymbol{x_{33}=b_3(1)=25}$. Потребности пункта B_3 полностью удовлетворены, поэтому все тарифы и соответствующие им маршруты, у которых пунктом назначения является пункт B_3, из дальнейшего рассмотрения исключаются

Трансформированный код после второго шага примет вид:

$D(2)$=<d_{22}, d_{12}, d_{32}, d_{21}, d_{31}, s_{11}>.

$a_1(2)$= 30, $a_2(2)$=40, $a_3(2)$=5.

$b_1(2)$=42, $b_2(2)$=33, $b_3(2)$=0, $b_4(2)$=0.

$\boldsymbol{x_{33}(2)=25}$.

На шаге *t*=3 в $D(2)$ выбирается маршрут d_{22}: $A_2 \rightarrow B_2$.

$a_2(2)$=0, $b_2(2)$=33. $a_2>b_2$ ситуация соответствует пункту (а) и $a_2(2)$ полностью покрывает b_2.

Отсюда $a_2(3)$=40-33=7, $b_2(3)=0$. $\boldsymbol{x_{22}(3)=b_2(2)=33}$. Потребности пункта B_2 полностью удовлетворены, поэтому все тарифы и соответствующие им маршруты, у которых пунктом назначения является пункт B_2, из дальнейшего рассмотрения исключаются, а в пункте A_2 запас $a_2(2)$ уменьшится на $b_2(2)$, т.е. $a_2(3)=a_2(2)-b_2(2)$=40-33=7. Трансформированный код после третьего шага примет вид:

$D(3)=<d_{21}, d_{31}, d_{11}>$.

$a_1(3)=30$, $a_2(3)=7$, $a_3(3)=5$.

$b_1(3)=42$, $b_2(3)=0$, $b_3(3)=0$, $b_4(3)=0$.

$\boldsymbol{x_{22}(3)=33}$.

На 4, 5 и 6 шагах последовательно в пункт B_1 назначаются 5 единиц груза из пункта A_3, 7 единиц груза из пункта A_2 и 30 единиц груза из пункта A_1.

Знания параметров после шестого шага примут вид:

$a_1(6)=0$, $a_2(6)=0$, $a_3(6)=0$.

$b_1(6)=0$, $b_2(6)=0$, $b_3(6)=0$, $b_4(6)=0$.

$\boldsymbol{x_{31}(4)=5, x_{21}(5)=7, x_{11}(6)=30}$.

Полученный план перевозок имеет вид:

$\boldsymbol{x_{14}=20,\ x_{33}=25,\ x_{22}=33,\ x_{31}=5,\ x_{21}=7,\ x_{11}=30}$.

Суммарная стоимость перевозок P, соответствующая плану определяется как:

$P=c_{14}\cdot x_{14}+c_{33}\cdot x_{33}+c_{22}\cdot x_{22}+c_{31}\cdot x_{31}+c_{21}\cdot x_{21}+c_{11}\cdot x_{11}$.

$\boldsymbol{P=1\cdot 20+1\cdot 25+2\cdot 33+2\cdot 5+3\cdot 7+4\cdot 30=262}$.

Код, соответствующий построенному решению, имеет вид:

$$\boldsymbol{D(0)=<d_{14}, d_{33}, d_{22}, d_{31}, d_{21}, d_{11}>}.$$

Отметим, что для выполнения плана перевозок используется 6 маршрутов из 12.

3. Механизмы решения транспортной задачи на основе метаэвристики кристаллизации россыпи альтернатив. В методе кристаллизации россыпи альтернатив [14–16] каждое решение Q_k формируется (представляется) множеством объектов (маршрутов) $D_к=<d_{kij}|i=1,\ldots,m,\ j=1,\ldots n>$. Возможное число маршрутов $R=mn$. Каждому маршруту d_{kij} соответствует множество альтернативных состояний $P=\{p_\beta|\beta=1, 2, \ldots, n\}$ где n – число состояний маршрута d_{kij}. Состоянию маршрута соответствует позиция в векторе $D_к$ в которой размещен элемент (маршрут) d_{kij}. Каждый маршрут d_{kij} может находиться в одном из альтернативных состояний (позиции в векторе $D_к$). Решение Q_k определяется совокупностью альтернативных состояний множества маршрутов.

Применение любой метаэвристики для решения комбинаторной задачи заключается в представлении исходной формулировки задачи в виде компонент метаэвристики.

В качестве кода решения ТЗ Q_k рассматривается упорядоченная последовательность маршрутов − $D_к=<d_{kij}|i=1,\ldots,m,\ j=1,\ldots n>$. В разработанном алгоритме решения ТЗ объектами являются маршруты, альтернативами – множество позиций $P=\{p_\beta|\beta=1, 2, \ldots, n\}$ в списке D_k, где n_p – число позиций в списке $D_к$ в которые помещаются элементы d_{kij}. Другими словами множество объектов $D_к=<d_{kij}|i=1,\ldots,m,\ j=1,\ldots n>$, соответствует множеству всех маршрутов. Множество альтернативных состояний $P=\{p_\beta|\beta=1, 2, \ldots, n\}$ объекта d_{kij} соответствует множеству альтернативных вариантов размещения объекта *в* списке $D_к$.

Каждому элементу d_{kij} списка $D_к$ соответствует один единственный элемент p_β. Номер позиции p_β в которую помещен элемент d_{kij} определяется соотношением $p_\beta=F_k(d_{kij})$.

Работа популяционного эволюционного алгоритма КРА опирается на коллективную эволюционную память, называемую **россыпью альтернатив**.

Под россыпью альтернатив (РА) решения в работе называется структура данных, используемая в качестве коллективной эволюционной памяти (КЭП), несущая информацию о решении, включающую сведения о реализованных альтернативах агентов в данном решении и о полезности решения.

Установим биективное отображения $F_k=NP\rightarrow NP$ между множеством чисел натурального ряда NP от 1 до $r=mn$ и множеством NP всех пар (i,j), таких, что $1\le i\le m;\ 1\le j\le n\ |NP|=mn$. Каждому элементу α множества NR соответствует один единственный элемент $(i,j)_l$ множества NP и наоборот. Номер элемента α определяется соотношением $\alpha=F_k((i,j)_\alpha)$ и наоборот. Тогда $d_{kij}=d_{k\alpha}$.

Представим структуру данных для отображения одного решения Q_k, формируемого множеством объектов (маршрутов), в виде матрицы альтернатив $R_k=||r_{k\alpha\beta}||_{m\times n}$. α – индекс объекта, β – индекс позиции.

Строки матрицы соответствуют – объектам (маршрутам) $d_{k\alpha}$. Размерность строки (вектора $R_{k\alpha}=\{r_{k\alpha\beta}|\beta=1,2,\ldots,\ n;\ \alpha\}$ определяется числом n возможных альтернатив − состояний объекта (маршрута $d_{k\alpha}$) – (числом позиций, в которые назначаются объекты).

Столбцы матрицы $R_k=||r_{k\alpha\beta}||_{m\times n}$ соответствуют позициям. Размерность столбца (вектора $R_{k\beta}=\{d_{k\alpha}|\alpha=1,2,\ldots,m,\ \beta\}$, определяется числом m объектов (маршрутов).

В каждой строке (векторе $R_{k\alpha}$) только один элемент $r_{k\alpha\beta}$, соответствующий позиции p_β, в которой находится объект $d_{k\alpha}$, имеет значение, отличное от нуля. Остальные элементы вектора $R_{k\alpha}$ имеют нулевые значения.

В каждом столбце (векторе $R_{k\beta}$ только один элемент $r_{k\alpha\beta}$, соответствующий объекту $d_{k\alpha}$, который расположен в позиции p_β, имеет значение, отличное от нуля. Остальные элементы вектора $R_{k\beta}$ имеют нулевые значения.

Каждый, отличный от нуля элемент $r_{k\alpha\beta}$ матрицы $R_k=||r_{k\alpha\beta}||_{m\times n}$, имеет значение, равное полезности δ_k решения Q_k, при котором объект $d_{k\alpha}$ (маршрут) назначен в позицию p_β, $\delta_k=f(\xi_k)$, где ξ_k – оценка решения, а δ_k – оценка полезности этого решения.

Таким образом, в матрицу R_k заносится информация о состояниях, реализованных объектами (маршрутами) в решении Q_k, т.е. о позициях в которые они назначены (с не нулевыми значениями) и об оценке полезности – δ_k этих состояний.

Пример. Представим матрицу россыпи альтернатив R_k для рассмотренного выше примера. Пусть рассчитанная оценка полезности имеет значение δ_k=6 для кода:

$D=<d_{14}, d_{24}, d_{34}, d_{33}, d_{23}, d_{13}, d_{22}, d_{12}, d_{32}, d_{21}, d_{31}, d_{11}>$.

$F_k(D)=<d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8, d_9, d_{10}, d_{11}, d_{12}>$.

Тогда матрица россыпи альтернатив для одного решения имеет вид, представленный в табл. 4:

Таблица 4

Матрица россыпи альтернатив

P	1	2	3	4	5	6	7	8	9	10	11	12
d_1	6											
d_2		6										
d_3			6									
d_4				6								
d_5					6							
d_6						6						
d_7							6					
d_8								6				
d_9									6			
d_{10}										6		
d_{11}											6	
d_{12}												6

На базе сгенерированного популяцией агентов множества решений $Q=\{Q_k|k=1,2,\ldots,n_k\}$ формируется множество индивидуальных россыпей альтернатив $R=\{R_k|k=1,2,\ldots,n_k\}$.

Платформой для организации эволюционной процедуры поиска решений является интегральная россыпь альтернатив (ИРА) − R^*, которая формируется путем объединения всех россыпей альтернатив:

$$R^*=||r^*_{\alpha\beta}||_{m\times n},\ \text{где}\ r^*_{\alpha\beta}=\sum_k(r_{k\alpha\beta}),\ \ k=\{1,n_k\}.$$

Фактически $r^*_{\alpha\beta}$ является суммарным значением полезностей решений, в которых агентами популяции маршрут d_α был назначен в позицию p_β. ИРА используется в качестве коллективной эволюционной памяти и служит базой для формирования новых решений.

Пример. Пусть агентами популяции построены 7 решений Q_1-Q_7, на множестве маршрутов $D(0)=<d_{14}, d_{33}, d_{22}, d_{31}, d_{21}, d_{11}>$, для которых рассчитаны оценки полезности – δ_k, приведенные в табл. 5.

Таблица 5

Решения с оценками полезности

P	p_1	p_2	p_3	p_4	p_5	p_6	
Q_1	d_{14}	d_{33}	d_{22}	d_{21}	d_{31}	d_{11}	$\delta_k = 6$
Q_2	d_{33}	d_{11}	d_{14}	d_{22}	d_{21}	d_{31}	$\delta_k = 4$
Q_3	d_{22}	d_{14}	d_{11}	d_{33}	d_{31}	d_{21}	$\delta_k = 5$
Q_4	d_{21}	d_{22}	d_{31}	d_{14}	d_{33}	d_{11}	$\delta_k = 7$
Q_5	d_{31}	d_{21}	d_{14}	d_{22}	d_{11}	d_{33}	$\delta_k =9$
Q_6	d_{22}	d_{11}	d_{33}	d_{14}	d_{21}	d_{31}	$\delta_k =8$
Q_7	d_{11}	d_{31}	d_{14}	d_{33}	d_{21}	d_{22}	$\delta_k =10$

Переименуем индексы маршрутов $D(0)$ в соответствии с графиком соответствия F_k.

$F_k(d_{14}, d_{33}, d_{22}, d_{31}, d_{21}, d_{11})=(d_1, d_2, d_3, d_4, d_5, d_6)$, табл. 6.

Таблица 6

Переименованные индексы маршрутов

P	p_1	p_2	p_3	p_4	p_5	p_6	
Q_1	d_1	d_2	d_3	d_4	d_5	d_6	$\delta_k = 6$
Q_2	d_2	d_6	d_1	d_3	d_4	d_5	$\delta_k = 4$
Q_3	d_3	d_1	d_6	d_2	d_5	d_4	$\delta_k = 5$
Q_4	d_4	d_3	d_5	d_1	d_2	d_6	$\delta_k = 7$
Q_5	d_5	d_4	d_1	d_3	d_6	d_2	$\delta_k =9$
Q_6	d_3	d_6	d_2	d_1	d_4	d_5	$\delta_k =8$
Q_7	d_6	d_5	d_1	d_2	d_4	d_3	$\delta_k =10$

Для каждого решения Q_k, формируемого множеством объектов (маршрутов) формируются матрицы альтернатив $R_k=||r_{k\alpha\beta}||_{m\times n}$.

На рис. 1 в целях компактности приведена совмещенная матрица альтернатив, позволяющая просмотреть матрицу альтернатив для каждого решения.

	p_1	p_2	p_3	p_4	p_5	p_6
d_1	6	5	4+ 9 +10	7+8		
d_2	4	6	8	5 +10	7	9
d_3	5+ 8	7	6	4+ 9		10
d_4	7	9		6	4+8 +10	5
d_5	9	10	7		6+ 5	4+8
d_6	10	4+8	5		9	6+7

Рис. 1. «Совмещенная» матрица альтернатив с отдельно показанными результатами всех решений

Интегральная россыпь альтернатив R^* имеет вид, представленный на рис. 2:

	p_1	p_2	p_3	p_4	p_5	p_6
d_1	6	5	23	15		
d_2	4	6	8	15	7	9
d_3	13	7	6	13		10
d_4	7	9		6	22	5
d_5	9	10	7		11	12
d_6	10	12	5		9	13

Рис. 2. Интегральная россыпь альтернатив R^*

Разработанный алгоритм является популяционным, реализующим стратегию случайного направленного поиска. Популяция – совокупность агентов, эволюционирующих в пространстве поиска. Каждый агент является кодом некоторого решения ТЗ. Процесс поиска решений итерационный. Алгоритм обладает памятью. На начальном этапе:

- ♦ Задается объем обучающей популяции решений − N_k.
- ♦ Генерируется начальное множество кодов решений $D=\{D_k \mid k=1,2,\dots, N_k\}$ путем случайного выбора альтернатив p_β.
- ♦ Формирование начальной интегральной россыпи альтернатив $R^*(0)$: присвоение начального значения всем элементам матрицы$R^*=||r^*_{\alpha\beta}||_{m\times n}$, где $r^*_{\alpha\beta}=\sum_k(r_{k\alpha\beta})$, $k=\{1,n_k\}$.

Каждая итерация l начинается с приведения в текущего состояние служебной памяти в исходное и выполнения трех этапов.

На первом этапе каждой итерации l конструктивным алгоритмом на базе интегральной россыпи альтернатив формируется n_k кодов решений D_k (популяция решений). Код решения D_k формируется (представляется) в виде упорядоченного множества объектов (маршрутов) $D_к=<d_{kij}|i=1,\dots,m, j=1,\dots n>$.

Каждый код решения D_k является биективным отображением $F_k=D\rightarrow P$ и формируется путем последовательного назначения элементов множества D в позиции множества P.

На первом этапе каждой итерации l конструктивным алгоритмом на базе интегральной россыпи альтернатив формируется n_k кодов решений D_k (популяция решений). На каждом шаге *процесс* назначения объекта в позицию включает две стадии. На первой стадии выбирается объект$(d_{k\alpha})$, а на второй стадии − позиция p_β. При этом должно выполняться ограничение: каждому объекту $d_{k\alpha}$ множества D_k соответствует один единственный элемент множества P и наоборот.

В текущей памяти хранится множество объектов $D^*(t)$, не назначенных в позиции, и множество $P^*(t)$ свободных позиций.

На <u>первой стадии</u> для каждого $d_{k\alpha}\in D^*(t)$ в соответствующей строке $R^*_{k\alpha}$ матрицы интегральной россыпи альтернатив определяется позиция $p_\beta^*(t)\in P^*(t)$ с максимальным значением суммарной полезности $\delta_k=r^*_{\alpha\beta}$ (где $r^*_{\alpha\beta}=\sum_k(r_{k\alpha\beta})$), которое обозначим как $\pi_{k\alpha}$. $\pi_{k\alpha}$ характеризует перспективность выбора $d_{k\alpha}$.

Среди множества не размещенных элементов $D^*(t)$ с вероятностью $P(d_{k\alpha})=\pi_{k\alpha}/\sum_\alpha(\pi_{k\alpha})$, пропорциональной $\pi_{k\alpha}$, выбирается элемент $d^*_{k\alpha}$.

На <u>второй стадии</u> среди множества свободных позиций $P^*(t)$, соответствующих выбранному элементу $d^*_{k\alpha}$, т.е. при выбранном α с вероятностью $P(p_\beta^*(t))=r^*_{\alpha\beta}/\sum_\beta (r^*_{\alpha\beta})$, пропорциональной значению оценки полезности $\delta=r^*_{\alpha\beta}$ выбирается позиция $p_\beta^*(t)$. Объект $d^*_{k\alpha}$ помещается в позицию $p_\beta^*(t)$, затем объект $d^*_{k\alpha}$ удаляется из $D^*(t)$, а позиция $p_\beta^*(t)$ удаляется из множества $P^*(t)$ и переход к следующему шагу.

Для построенного кода решения D_k рассчитывается оценка решения ξ_k и оценка полезности δ_k. Формируется индивидуальная россыпь альтернатив R_k для Q_k и переход к построению следующего кода решения.

На втором этапе итерации производится суммирования интегральной россыпи альтернатив $R^*(l\text{-}1)$, сформированной на предыдущих итерациях от 1 до $(l\text{-}1)$, со всеми индивидуальными россыпями альтернатив R_k, сформированных на итерации l.

На третьем этапе итерации l производится снижение всех интегральных оценок полезности $r^*_{\alpha\beta}$ интегральной россыпи альтернатив $R^*(l)$на величину δ^*.

Фиксация и вывод лучшего решения.

4. Экспериментальные исследования. Алгоритм решения транспортной задачи был реализован на языке C++ в среде Windows. Экспериментальные исследования проводились на ЭВМ типа IBM PC/AT. Проведение экспериментов преследовало две цели: исследование эффективности и качества механизмов алгоритма кристаллизации россыпи альтернатив для решения транспортной задачи. Для этих целей была использована процедура синтеза контрольных примеров с известным оптимумом по аналогии с известным методом PEKU (Placement Examples with Known Upper bounds on wirelength) [17-21].

Сравнение значений критерия, на тестовых примерах, с известным оптимумом показало, что у 90% примеров полученное решение было оптимальным, у 2% примеров решения были на 5% хуже, а у 8% примеров решения были хуже не более, чем на 2%. Для проведения объективных экспериментов были использованы известные тестовые задачи, представленные в литературе и Интернет. Задачи, на которых был протестирован разработанный алгоритм, доступны в библиотеке OR-объектов (http://www.ms.ic.ac.uk/info.html).

Для составления достоверных выводов был проведен не один, а серия опытов-экспериментов. По сравнению с существующими алгоритмами достигнуто улучшение результатов на 2–3%.

Была произведена серия по измерению продолжительности работы алгоритма от количества маршрутов. Общая оценка временной сложности лежит в пределах $O(n^2)$-$O(n^3)$. Временная сложность алгоритма (ВСА), полученная экспериментальным путем, практически совпадает с теоретическими исследованиями и для рассмотренных тестовых задач составляет (ВСА $\approx O(n^2)$).

На основе обработки экспериментальных исследований была построена средняя зависимость качества решений от числа итераций (рис. 3). Оценкой качества служит величина $F/F_{\text{опт}}$, где F –оценка полученного решения. Исследования показали, что число итераций, при которых алгоритм находил лучшее решение, лежит в пределах 130–150. Из графика видно, что в среднем на 140 итерации решение близко к оптимальному.

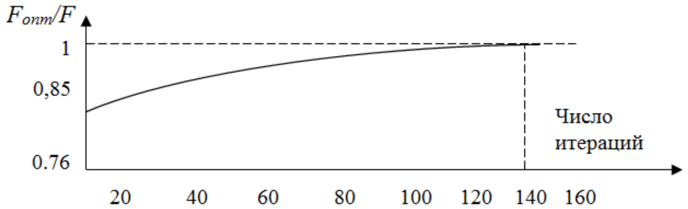

Рис. 3. Зависимость качества решений от числа итераций

Заключение. На основе сравнительного анализа существующих подходов и методов для решения транспортной задачи использованы мультиагентные методы интеллектуальной оптимизации.

Разработана новая модификация метаэвристики интеллектуальной оптимизации, учитывающая тенденцию к использованию альтернатив из наилучших найденных решений, базирующаяся на моделировании коллективного интеллекта. Совокупность данных об альтернативах и их оценках составляет россыпь альтернатив. В методе кристаллизации россыпи альтернатив каждое решение формируется (представляется) множеством объектов. Каждому объекту соответствует множество альтернативных состояний. Решение определяется совокупностью альтернативных состояний множества агентов.

Отличительной особенностью представленной метаэвристики является использование упорядоченной последовательности маршрутов в качестве кода решения ТЗ. Разработанный алгоритм построения опорного плана обладает линейной трудоемкостью и позволяет получать квазиоптимаьные решения. Рассмотрены ключевые моменты анализа альтернатив в процессе эволюционной коллективной адаптации, названной по аналогии с процессами вычленения объектов (формирования кристаллов) кристаллизацией. Алгоритм на основе кристаллизации россыпи альтернатив был успешно применен для решения транспортной задачи и является эффективным способом поиска рациональных решений для задач оптимизации, допускающих интерпретацию в виде россыпи альтернатив.

Экспериментальные исследования показали, что алгоритмы на основе предлагаемого подхода могут давать лучшие результаты, чем при использовании классических методов для решения транспортных задач.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Корбут А.А., Финкельштейн Ю.Ю.* Дискретное программирование. – М.: Наука: Гл. ред. физ.-мат. лит., 1969.
2. *Вентцель Е.С.* Исследование операций: задачи, принципы, методология: учебник для вузов. – М.: Дрофа, 2004. – 208 с.
3. *Таха Х.А.* Введение в исследование операций. – 7-е изд. – М.: Изд. дом «Вильямс», 2005.
4. *Engelbrecht A.P.* Fundamentals of Computational Swarm Intelligence. – John Wiley & Sons: Chichester, UK, 2005.
5. *Юдин Д.Б., Гольштейн Е.Г.* Задачи и методы линейного программирования: Задачи транспортного типа. – М.: Изд. 3, 2013. – 184 с.
6. *Рудик И.Д., Величко В.В.* Понятие, виды и методы решения транспортной задачи // Международный студенческий научный вестник. – 2017. – № 4-4.
7. *Dorigo M., Stützle T.* Ant Colony Optimization. – MIT Press: Cambridge, MA, 2004. – 154 p.
8. *Poli R.* Analysis of the publications on the applications of particle swarm optimization // Journal of Artificial Evolution and Applications: Article ID 685175. – 2008. – P. 10-21.
9. *Курейчик В.М., Лебедев Б.К., Лебедев О.Б.* Поисковая адаптация: Теория и практика. – М.: Физматлит, 2006. – 272 с.
10. *Лебедев О.Б.* Модели адаптивного поведения муравьиной колонии в задачах проектирования. – Таганрог. Изд-во ЮФУ, 2013. – 199 с.
11. *Карпенко А.П.* Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой: учеб. пособие. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2016. – 448 с.
12. *Гладков Л.А., Гладкова Н.В.* Решение динамических транспортных задач на основе гибридных интеллектуальных методов и моделей // Известия ЮФУ. Технические науки. –2013. – № 7 (144). – С. 102-107.
13. *Курейчик В.М., Кажаров А.А.* Муравьиные алгоритмы для решения транспортных задач // Известия РАН. Теория и системы управления. – 2010. – № 1. – С. 32-45.
14. *Курейчик В.М., Емельянова Т.С.* Решение транспортных задач с использованием комбинированного генетического алгоритма // Одиннадцатая национальная конференция по искусственному интеллекту с международным участием КИИ-2008. – М.: Физматлит, 2008. – С. 158-164.
15. *Лебедев Б.К., Лебедев В.Б.* Оптимизация методом кристаллизации россыпи альтернатив // Известия ЮФУ. Технические науки. – 2013. – № 7. – С. 11-17.

16. *Лебедев Б.К., Лебедев В.Б.* Метод кристаллизации россыпи альтернатив // Сб. научных трудов VII Международной научно-практической конференции “Интегрированные модели и мягкие вычисления в искусственном интеллекте”. – М.: Изд-во Физматлит, 2013. – С. 903-912.
17. *Лебедев Б.К., Лебедев В.Б.* Глобальная трассировка методом кристаллизации россыпи альтернатив // Известия ЮФУ. Технические науки. – 2014. – № 7 (156). – С. 42-51.
18. *Лебедев Б.К., Лебедев В.Б.* Построение кратчайших связывающих соединений методом кристаллизации россыпи альтернатив // VI Всероссийская научно-техническая конференция «Проблемы разработки перспективных микро- и наноэлектронных систем – 2014»: Сб. трудов / под общ. ред. академика РАН А.Л. Стемпковского. – М.: ИППМ РАН, 2014. – С. 177-183.
19. *Mourelle M.* Swarm intelligent systems. – Berlin: Heidelberg: Springer, 2006. – 217 p.
20. *Cong J., Romesis M., Xie M.* UCLA Optimality Study Project. – http://cadlab.cs.ucla.edu/~pubbench. 2004.
21. MCNC Electronic and Information Technologies (Online). Available: www.mcnc.org.

REFERENCES

1. *Korbut A.A., Finkel'shteyn Yu.Yu.* Diskretnoe programmirovanie [Discrete programming]. Moscow: Nauka: Gl. red. fiz.-mat. lit., 1969.
2. *Venttsel' E.S.* Issledovanie operatsiy: zadachi, printsipy, metodologiya: uchebnik dlya vuzov [Operations research: tasks, principles, methodology: textbook for universities]. Moscow: Drofa, 2004, 208 p.
3. *Takha Kh.A.* Vvedenie v issledovanie operatsiy [Introduction to operations research]. 7th ed. Moscow: Izd. dom «Vil'yams», 2005.
4. *Engelbrecht A.P.* Fundamentals of Computational Swarm Intelligence. John Wiley & Sons: Chichester, UK, 2005.
5. *Yudin D.B., Gol'shteyn E.G.* Zadachi i metody lineynogo programmirovaniya: Zadachi transportnogo tipa [Problems and methods of linear programming: Problems of transport type]. Moscow: Izd. 3, 2013, 184 p.
6. *Rudik I.D., Velichko V.V.* Ponyatie, vidy i metody resheniya transportnoy zadachi [The concept, types and methods of solving the transport problem], *Mezhdunarodnyy studencheskiy nauchnyy vestnik* [International Student Scientific Bulletin], 2017, No. 4-4.
7. *Dorigo M., Stützle T.* Ant Colony Optimization. MIT Press: Cambridge, MA, 2004, 154 p.
8. *Poli R.* Analysis of the publications on the applications of particle swarm optimization, *Journal of Artificial Evolution and Applications: Article ID 685175,* 2008, pp. 10-21.
9. *Kureychik V.M., Lebedev B.K., Lebedev O.B.* Poiskovaya adaptatsiya: Teoriya i praktika [Search adaptation: Theory and practice]. Moscow: Fizmatlit, 2006, 272 p.
10. *Lebedev O.B.* Modeli adaptivnogo povedeniya murav'inoy kolonii v zadachakh proektirovaniya [Models of adaptive behavior of an ant colony in design problems]. Taganrog. Izd-vo YuFU, 2013, 199 p.
11. *Karpenko A.P.* Sovremennye algoritmy poiskovoy optimizatsii. Algoritmy, vdokhnovlennye prirodoy: ucheb. posobie [Modern search engine optimization algorithms. Algorithms inspired by nature]. Moscow: Izd-vo MGTU im. N.E. Baumana, 2016, 448 p.
12. *Gladkov L.A., Gladkova N.V.* Reshenie dinamicheskikh transportnykh zadach na osnove gibridnykh intellektual'nykh metodov i modeley [Solving dynamic transport problems based on hybrid intelligent methods and models], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2013, No. 7 (144), pp. 102-107.
13. *Kureychik V.M., Kazharov A.A.* Murav'inye algoritmy dlya resheniya transportnykh zadach [Ant algorithms for solving transport problems], *Izvestiya RAN. Teoriya i sistemy upravleniya* [Izvestiya RAS. Theory and control systems], 2010, No. 1, pp. 32-45.
14. *Kureychik V.M., Emel'yanova T.S.* Reshenie transportnykh zadach s ispol'zovaniem kombinirovannogo geneticheskogo algoritma [Solving transport problems using a combined genetic algorithm], *Odinnadtsataya natsional'naya konferentsiya po iskusstvennomu intellektu s mezhdunarodnym uchastiem KII-2008* [Eleventh National Conference on Artificial Intelligence with International Participation CII-2008]. Moscow: Fizmatlit, 2008, pp. 158-164.

15. *Lebedev B.K., Lebedev V.B.* Optimizatsiya metodom kristallizatsii rossypi al'ternativ [Optimization by the method of crystallization of a placer of alternatives], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2013, No. 7, pp. 11-17.
16. *Lebedev B.K., Lebedev V.B.* Metod kristallizatsii rossypi al'ternativ [The method of crystallization of a placer of alternatives], *Sb. nauchnykh trudov VII Mezhdunarodnoy nauchno-prakticheskoy konferentsii «Integrirovannye modeli i myagkie vychisleniya v iskusstvennom intellekte»* [Collection of scientific papers of the VII International Scientific and Practical Conference "Integrated Models and Soft Computing in Artificial Intelligence"]. Moscow: Izd-vo Fizmatlit, 2013, pp. 903-912.
17. *Lebedev B.K., Lebedev V.B.* Global'naya trassirovka metodom kristallizatsii rossypi al'ternativ [Global tracing by the method of crystallization of a placer of alternatives], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2014, No. 7 (156), pp. 42-51.
18. *Lebedev B.K., Lebedev V.B.* Postroenie kratchayshikh svyazyvayushchikh soedineniy metodom kristallizatsii rossypi al'ternativ [Construction of the shortest binding connections by the method of crystallization of a placer of alternatives], *VI Vserossiyskaya nauchno-tekhnicheskaya konferentsiya «Problemy razrabotki perspektivnykh mikro- i nanoelektronnykh sistem – 2014»: Sb. trudov* [VI All-Russian scientific and technical conference "Problems of developing promising micro- and nanoelectronic systems – 2014": Collection of works], under the total. ed. academician of the RAS A.L. Stempkovskogo. Moscow: IPPM RAN, 2014, pp. 177-183.
19. *Mourelle M.* Swarm intelligent systems. Berlin: Heidelberg: Springer, 2006, 217 p.
20. *Cong J., Romesis M., Xie M.* UCLA Optimality Study Project. Available at: http://cadlab.cs.ucla.edu/~pubbench. 2004.
21. MCNC Electronic and Information Technologies (Online). Available at: www.mcnc.org.

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

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

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

Лебедева Екатерина Олеговна – e-mail: lbedevakate@mail.ru; тел.: 89289591426; кафедра систем автоматизированного проектирования; аспирант.

Lebedev Boris Konstantinovich – Southern Federal University; e-mail: lebedev.b.k@gmail.com; Taganrog, Russia; phone: +79282897933; the department of computer aided design; professor.

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

Lebedeva Ekaterina Olegovna – e-mail: lbedevakate@mail.ru; phone: +79289591426; the department of computer aided design; graduate student.

УДК 004.428.4+336.76.066 DOI 10.18522/2311-3103-2022-4-157-169

Т.Н. Кондратьева, Е.Р. Мунтян, И.Ф. Развеева

РЕАЛИЗАЦИЯ ТОРГОВОГО СОВЕТНИКА ДЛЯ МУЛЬТИРЫНОЧНОЙ ПЛАТФОРМЫ METATRADER 5

В статье рассмотрен процесс создания гибкой торговой стратегии для алготрейдинга в специализированной среде разработки MQL5 IDE в мультирыночной платформе MetaTrader 5. Показаны преимущества и целесообразность использования платформ MetaTrader 5, MetaTrader 4 и соответствующих им торговых приложений Trade Assistant, Forex Trade Manager, Trade Time Manager, CAP Gold Albatross EA и Fast Copy. Проведен