

УДК 519.7

## Разработка и применение многокритериального метода муравьиных колоний в задаче оптимизации инвестиционного портфеля

**Пантелеев А.В.\***

Московский авиационный институт  
(национальный исследовательский университет) (МАИ)  
г. Москва, Российская Федерация  
ORCID: <https://orcid.org/0000-0003-2493-3617>  
e-mail: [avpanteleev@inbox.ru](mailto:avpanteleev@inbox.ru)

**Попова Н.С.\*\***

Московский авиационный институт  
(национальный исследовательский университет) (МАИ)  
г. Москва, Российская Федерация  
ORCID: <https://orcid.org/0009-0000-7196-7786>  
e-mail: [popovanatalya472@gmail.com](mailto:popovanatalya472@gmail.com)

Предложен численный метод многокритериальной оптимизации для приближенного решения задачи на основе генерации допустимых решений непрерывным методом муравьиных колоний, недоминируемой сортировки и метода эпсилон-ограничений. Под решением задачи понимается нахождение фронта Парето. Приведены решения типовых модельных примеров. Решена прикладная задача оптимизации инвестиционного портфеля, в которой исходными данными являются таблично заданные средние доходности и ковариации ценных бумаг.

**Ключевые слова:** многокритериальная оптимизация, метаэвристические методы, недоминируемая сортировка, метод  $\varepsilon$ -ограничений, метод муравьиных колоний, оптимальность по Парето, оптимизация инвестиционного портфеля.

**Для цитаты:**

Пантелеев А.В., Попова Н.С. Разработка и применение многокритериального метода муравьиных колоний в задаче оптимизации инвестиционного портфеля // Моделирование и анализ данных. 2024. Том 14. № 2. С. 80–97. DOI: <https://doi.org/10.17759/mda.2024140205>

\***Пантелеев Андрей Владимирович**, доктор физико-математических наук, профессор, заведующий кафедрой математической кибернетики института «Компьютерные науки и прикладная математика» Московского авиационного института (Национального исследовательского



университета) (МАИ), г. Москва, Российская Федерация, ORCID: <https://orcid.org/0000-0003-2493-3617>, e-mail: [avpanteleev@inbox.ru](mailto:avpanteleev@inbox.ru)

**\*\*Попова Наталья Сергеевна**, студент бакалавриата института «Компьютерные науки и прикладная математика» Московского авиационного института (Национального исследовательского университета) (МАИ), г. Москва, Российская Федерация, ORCID: <https://orcid.org/0009-0000-7196-7786>, e-mail: [popovanatalya472@gmail.com](mailto:popovanatalya472@gmail.com)

## 1. ВВЕДЕНИЕ

Сложность современных систем и множество противоречивых целей при их проектировании требуют разработки методов, способных учитывать несколько критериев в процессе принятия решений [1–7]. В работе представлен подход, основанный на аппроксимации границы Парето. Предлагается модификация непрерывной версии метода муравьиных колоний для решения задачи однокритериальной оптимизации [8–10]. Он относится к группе метаэвристических алгоритмов глобальной оптимизации, которые позволяют найти решения хорошего качества за приемлемое время [10]. В разработанном алгоритме предлагается использовать метод недоминируемой сортировки [6,7,11] с учетом возможного незначительного нарушения заданных ограничений типа равенств и неравенств [12–16]. Эффективность алгоритма продемонстрирована на типовых модельных примерах с разной структурой множества Парето. В результате сформированы рекомендации по выбору гиперпараметров алгоритма, что позволило поставить и решить задачу оптимизации инвестиционного портфеля [17–19].

## 2. ПОСТАНОВКА ЗАДАЧИ

Рассматривается задача многокритериальной конечномерной оптимизации с ограничениями. Определим множество допустимых решений

$$D = \{x \in \mathbb{R}^n \mid g_i(x) \leq 0, i = 1, \dots, s; h_j(x) = 0, j = 1, \dots, w\}. \quad (1)$$

При приближенном решении задачи в качестве множества допустимых решений, как правило, рассматривается пересечение множества  $D$  с множеством параллелепипедного типа, образованного ограничениями вида  $a_i \leq x_i \leq b_i, i = 1, \dots, n$  (получаемое множество также будем обозначать через  $D$  для упрощения обозначений). Предполагается, что каждое решение  $x$  оценивается по  $t$  критериям ( $t \geq 2$ ), имеющим одинаковую важность, причем уменьшение значения одного критерия при фиксированных значениях остальных критериев более предпочтительно. Целевая вектор-функция имеет вид

$$F(x) = (f_1(x), f_2(x), \dots, f_t(x))^T,$$

где  $f_1(x), f_2(x), \dots, f_t(x)$  – заданные непрерывные функции,  $x \in D$ .



Тогда желаемым решением задачи является результат минимизации каждой из частных целевых функций с учетом заданных ограничений типа равенств и неравенств:

$$\min_{x \in D} F(x) = F(x^*) = F^*.$$

Однако, в общем случае невозможно обеспечить достижения минимума всех частных целевых функций на одном и том же решении  $x^*$ , поэтому решение проблемы понимается иначе. Приведем несколько используемых далее определений [6].

Определение 1. Вектор  $F(x) \in \mathbb{R}^t : F(x) = (f_1(x), \dots, f_t(x))^T$  называется векторной оценкой решения  $x \in D$ .

Определение 2. Вектор оценок  $F^1 = F(x^1)$ ,  $x^1 \in D$  доминирует вектор  $F^2 = F(x^2)$ ,  $x^2 \in D : F^1 \prec F^2$ , если  $\forall i \in \{1, \dots, t\} : f_i^1 \leq f_i^2$  и  $\exists j \in \{1, \dots, t\} : f_j^1 < f_j^2$ .

Определение 3. Вектор решений  $x^1 \in D$  доминирует вектор  $x^2 \in D$ , если  $F^1 \prec F^2$ .

Определение 4. Множество  $P$  является множеством решений, оптимальных по Парето, если  $P = \{x \in D \mid \nexists x' \in D : F(x') \prec F(x)\}$ .

Определение 5. Множество векторных оценок  $F(P) = \{F(x) \mid x \in P\}$  называется фронтом Парето.

Требуется найти аппроксимации множества векторных оценок (фронта Парето) и множества решений, оптимальных по Парето.

### 3. СТРАТЕГИЯ ПОИСКА РЕШЕНИЯ

Для решения задачи предлагается модифицировать метод муравьиных колоний, применяемый в однокритериальных задачах оптимизации [8–10], дополнив его механизмами недоминируемой сортировки решений [6,7,11] и способом учета выполнения ограничений [12,13]. Базовый метод муравьиных колоний моделирует действия колонии муравьев при поиске оптимального маршрута в изменяющейся среде обитания. При прохождении по какому-либо участку пути муравей откладывает особое пахучее вещество, называемое феромоном. Чем сильнее концентрация феромонов на тропе, тем более привлекательна она для других муравьев. Таким образом, муравьи узнают, какие маршруты чаще всего используются колонией.

На каждой итерации на множестве  $D$  поиск решения производят  $m$  муравьев, образующих текущее поколение. Роль памяти муравьиной колонии играет архив решений – таблица, содержащая  $r$  наилучших найденных решений. Величина  $r$  называется размером архива решений.

На каждой  $k$ -й итерации муравей с номером  $j$  по данным из архива решений, должен получить новое решение  $\tilde{x}^j = (\tilde{x}_1^j, \dots, \tilde{x}_n^j)^T$ . При этом реализуется вероятностный выбор на основе известной плотности вероятности. Для  $i$ -й координаты произвольной точки  $x$  она имеет вид:

$$G(x_i) = \sum_{l=1}^r \omega_l g_l(x_i) = \sum_{l=1}^r \omega_l \frac{1}{\sigma_l^j \sqrt{2\pi}} \exp\left(\frac{-(x_i - \mu_l^j)^2}{2(\sigma_l^j)^2}\right)$$

где  $a_i \leq x_i \leq b_i$ ,  $i = 1, \dots, n$ ;  $\omega_l$ ,  $l = 1, \dots, r$ , – весовой коэффициент,  $g_l(x_i)$  – гауссовская плотность вероятности, определяемая математическим ожиданием  $\mu_i^l$  и среднеквадратическим отклонением  $\sigma_i^l$ .

Данные о текущем поколении муравьев заносятся в архив решений. Подсчитываются значения координат целевой вектор-функции, после чего по ним проводится недоминируемая сортировка – ранжирование решений на основе их векторных оценок с использованием метода  $\varepsilon$ -ограничений.

Рассмотрим недоминируемую сортировку подробнее: пусть  $I = \{x^p \mid x^p \in D, p = 1, \dots, r\}$  – множество решений на текущей итерации, где  $r \geq 1$ . Результатом сортировки является разбиение множества  $I$  на  $n$  непересекающихся подмножеств  $A_i$ ,  $i = 1, \dots, n$ ,  $1 \leq n \leq r$ ,  $n$  – номер последнего подмножества в разбиении:

$$\begin{aligned}
 I &= \bigcup_{i=1}^n A_i, \quad A_i \cap A_j = \emptyset, i \neq j, \\
 A_1 &= \{x \in I \mid \nexists x' \in I : F(x') \prec F(x)\}, \\
 &\vdots \\
 A_i &= \left\{x \in I \setminus \bigcup_{i=1}^{i-1} A_i \mid \nexists x' \in I \setminus \bigcup_{i=1}^{i-1} A_i : F(x') \prec F(x)\right\}, \\
 &\vdots \\
 A_n &= I \setminus \bigcup_{i=1}^{n-1} A_i.
 \end{aligned} \tag{2}$$

Другими словами, недоминируемая сортировка представляет собой повторяющуюся процедуру выделения предпочтительных решений. На первом шаге выбираются предпочтительные решения из множества  $I$ . Далее эти предпочтительные решения удаляются из  $I$ , и процедура повторяется к оставшейся части.

В недоминируемую сортировку включается метод  $\varepsilon$ -ограничений. Его основная идея заключается в том, что для всех решений  $x_k \in I$ ,  $k = 1, \dots, r$ , вычисляется степень ошибки (Constraint Violation, CV):

$$CV(x_k) = \sum_{i=1}^s \frac{\max\{g_i(x_k) - \varepsilon, 0\}}{\max_{x \in I} \{g_i(x)\}} + \sum_{j=1}^w \frac{\max\{|h_j(x_k)| - \varepsilon, 0\}}{\max_{x \in I} |h_j(x_k)|}, \tag{3}$$

где  $\varepsilon > 0$  – параметр, задающийся в алгоритме. Решения, у которых  $CV(x) = 0$ , считаются приоритетнее решений, у которых  $CV(x) \neq 0$ .

В архив решений записываются решения в том же порядке, что и во множествах  $A_1, A_2, \dots, A_n$ . После этого последние  $m$  точек из архива решений отбрасываются, таким образом, размер архива решений остается неизменным. Решения из архива используются для нахождения трех параметров: векторов  $\omega$ ,  $\mu^j$ ,  $\sigma^j$ , которые применяются для формирования плотностей  $g_j(x_i)$ ,  $i = 1, \dots, n$ , определяющих движение муравьев (их возможные последующие координаты) в процессе поиска решения.



Для выбранной муравьем с номером  $j$  плотности вероятности  $g_{j^*}(x_i)$  вектор математического ожидания  $\mu^{j^*}$  принимается равным координатам точки с номером  $j^*$ :

$$\mu_i^{j^*} = x_i^{j^*}, \quad i = 1, \dots, n,$$

а среднее квадратическое отклонение вычисляется следующим образом:

$$\sigma_i^{j^*} = \xi \cdot \sum_{l=1}^r \frac{|x_i^l - x_i^{j^*}|}{r-1}, \quad i = 1, \dots, n,$$

где среднее расстояние от точки  $x_i^{j^*}$  до значений соответствующих координат других решений умножается на параметр  $\xi > 0$ . Плохие решения забываются, а поиск перемещается к новым точкам множества  $D$ , отличным от хранящихся в архиве.

Каждой плотности вероятности  $g_l(x_i)$ ,  $l = 1, \dots, r$ , ставится в соответствие весовой коэффициент

$$\omega_l = \frac{1}{qr\sqrt{2\pi}} \exp\left(\frac{-(l-1)^2}{2q^2r^2}\right),$$

где  $q$  – параметр алгоритма.

На практике генерирование новой координаты  $x_i$  происходит в два этапа. На первом выбирается случайным образом плотность вероятности  $g_{j^*}(x_i)$  при помощи метода рулетки с вероятностью выбора

$$p_l = \frac{\omega_l}{\sum_{s=1}^r \omega_s}, \quad l = 1, \dots, r.$$

На втором этапе генерируется значение  $\tilde{x}_i^j$  в соответствии с выбранной гауссовской плотностью  $g_{j^*}(x_i)$ , определяемой параметрами  $\mu_i^{j^*}$ ,  $\sigma_i^{j^*}$ . Здесь  $j^*$  – номер плотности вероятности, выбранной муравьем с номером  $j$ .

В результате получается множество решений  $\{x^1, \dots, x^r, \tilde{x}^1, \dots, \tilde{x}^m\}$ , после чего по ним реализуется недоминируемая сортировка. Среди непересекающихся множеств  $A_1, A_2, \dots, A_n$  выбираются решения, пока их количество не будет равно  $r$ . Если в каком-то подмножестве решений больше, чем необходимо выбрать, то необходимая часть решений отбирается случайным образом на основе расстояний до других решений в этом подмножестве.

## 4. АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ

### Шаг 1. Генерация начальных позиций.

Шаг 1.1. Задать число муравьев  $m$ , размер архива  $r$ , максимальное число поколений  $K$ , параметры  $q$ ,  $\xi$  и  $\varepsilon$ . Положить номер поколения  $k = 0$ .

Шаг 1.2. Сгенерировать  $r$  точек  $I^k = \{x^1, \dots, x^r\}$  на множестве  $D$ , используя равномерное распределение.

### Шаг 2. Расчет весов и вероятностей выбора решений.

Шаг 2.1. Для каждого решения  $x^l$ ,  $l = 1, \dots, r$ , подсчитать вес  $\omega_l$ :

$$\omega_l = \frac{1}{qr\sqrt{2\pi}} \exp\left(\frac{-(l-1)^2}{2q^2r^2}\right).$$

Шаг 2.2. Вычислить вероятность  $p_l$  выбора  $l$ -й гауссовской плотности вероятности:

$$p_l = \frac{\omega_l}{\sum_{s=1}^r \omega_s}, \quad l = 1, \dots, r.$$

### Шаг 3. Выбор плотности вероятности.

Шаг 3.1. Положить  $j = 1$ .

Шаг 3.2. Методом рулетки выбрать номер муравья  $j^*$ .

### Шаг 4. Генерация вектора $\tilde{x}^j$ .

Шаг 4.1. Положить  $i = 1$ .

Шаг 4.2. Вычислить математическое ожидание  $\mu_i^{j^*}$  и среднеквадратическое отклонение  $\sigma_i^{j^*}$ :

$$\begin{aligned} \mu_i^{j^*} &= x_i^{j^*}, \quad i = 1, \dots, n \\ \sigma_i^{j^*} &= \xi \cdot \sum_{l=1}^r \frac{|x_l^j - x_i^{j^*}|}{r-1}, \quad i = 1, \dots, n \end{aligned}$$

Шаг 4.3. Генерировать случайную величину  $\tilde{x}_i^j$  на отрезке  $[a_i, b_i]$  согласно плотности вероятности

$$g_{j^*}(x_i) = \frac{1}{\sigma_i^{j^*} \sqrt{2\pi}} \exp\left(\frac{-(x_i - \mu_i^{j^*})^2}{2(\sigma_i^{j^*})^2}\right)$$

Шаг 4.4. Если  $i = n$ , сформировать и добавить новое решение  $\tilde{x}^j = (\tilde{x}_1^j, \dots, \tilde{x}_n^j)^T$  к текущему числу поколения  $I^k : I^k = I^k \cup \{\tilde{x}^j\}$ , и перейти к шагу 5. Если  $i < n$ , положить  $i = i + 1$  и перейти к шагу 4.2.



**Шаг 5.** Если  $j = m$ , процесс поиска решения поколением муравьев завершить и перейти к шагу 6. Если  $j < m$ , положить  $j = j + 1$  и перейти к шагу 3.2.

**Шаг 6.** Формирование нового поколения.

Шаг 6.1. Для каждого решения  $x^l$ ,  $l = 1, \dots, r + m$ , из множества  $I^k$  вычислить степень ошибки по формуле (3):

$$CV(x_k) = \sum_{i=1}^s \frac{\max\{g_i(x_k) - \varepsilon, 0\}}{\max_{x \in I} \{g_i(x)\}} + \sum_{j=1}^w \frac{\max\{|h_j(x_k)| - \varepsilon, 0\}}{\max_{x \in I} |h_j(x_k)|},$$

где  $g_i(x^l)$ ,  $i = 1, \dots, s$  – функции, задающие ограничения типа неравенств,  $h_j(x^l)$ ,  $j = 1, \dots, w$  – функции, задающие ограничения типа равенств.

Шаг 6.2. Разделить множество  $I^k$  на два подмножества:  $I_0^k$  – решения, у которых  $CV(x^l) = 0$ , и  $I_1^k$  – решения, у которых  $CV(x^l) \neq 0$ .

Шаг 6.3. Реализовать процедуру недоминируемой сортировки согласно (2)  $I_0^k : A_1, \dots, A_n$  и  $I_1^k : A_{n+1}, \dots, A_a$ , где  $n$  и  $a$  – номера последних подмножеств в разбиениях  $I_0^k$  и  $I_1^k$  соответственно. Увеличить счетчик числа итераций  $k = k + 1$ ,  $I^k = \emptyset$ .

Шаг 6.4. Соединить два подмножества  $I_0^k$  и  $I_1^k$  в одно  $I^k$ .

Шаг 6.5. Найти  $u_{\min}$  :

$$u_{\min} = \min_{1 \leq u \leq s} \left\{ u : \left| \bigcup_{i=1}^u A_i \right| \geq r \right\}.$$

Если  $\left| \bigcup_{i=1}^{u_{\min}} A_i \right| = r$ , то  $I^k = \bigcup_{i=1}^{u_{\min}} A_i$ ,  $P = \emptyset$ , иначе  $I^k = \bigcup_{i=1}^{u_{\min}-1} A_i$ ,  $P = A_{u_{\min}}$ . Если

$u_{\min} = 1$  и  $|A_{u_{\min}}| > r$ , то  $\bigcup_{i=1}^{u_{\min}-1} A_i = \emptyset$ , все решения попадают в множество  $P$ .

Шаг 6.6. Для каждой точки  $x^w \in P$  подсчитать  $R(x^w)$  – сумму расстояний до остальных точек,  $p(x^w)$  – вероятность выбора решения  $x^w$ .

$$R(x^w) = \sum_{x^s \in P} \rho(F(x^w), F(x^s)), \quad p(x^w) = \frac{R(x^w)}{\sum_{x^s \in P} R(x^s)}.$$

Шаг 6.7. Используя вероятность  $p(x^w)$ , случайным образом выбрать из множества  $P$  множество точек (решений) в количестве  $r - |I^k|$  и добавить их в множество  $I^k$ .

Шаг 6.8. Если  $k \leq K$ , то положить  $k = k + 1$  и перейти к шагу 2. Иначе в качестве приближенного решения принять  $I^k$  и  $F(I^k)$ .

## 5. МОДЕЛЬНЫЕ ПРИМЕРЫ

Для тестирования метода были взяты задачи с известным точным решением:  $ZDT1$  (рис. 1),  $ZDT2$  (рис. 2),  $ZDT3$  (рис. 3). Чтобы оценить, насколько полученное решение близко к истинной границе Парето, были использованы метрики  $IGD$  (Inverted Generational Distance) и  $HV$  (HyperVolume, гиперобъем).

Идея  $IGD$  заключается в том, чтобы измерить среднее расстояние от всех точек Парето-оптимального фронта до ближайшей точки из набора решений, который возвращает программа. Чем меньше это расстояние, тем лучше качество приближения. Метрика рассчитывается следующим образом:

$$IGD = \frac{1}{|I|} \sum_{x \in I} \min_{\tilde{x} \in P} \rho(F(x), F(\tilde{x})),$$

где  $\rho(\cdot, \cdot)$  – расстояние от одной точки до другой:  $\rho(y, \tilde{y}) = \sqrt{\sum_{i=1}^{|I|} (y_i - \tilde{y}_i)^2}$ .

Метрика  $HV$  основывается на концепции измерения объема пространства, ограниченного гиперплоскостью, которая определена оптимальным значением каждого критерия. Поскольку  $HV$  не учитывает распределение точек внутри этого пространства, она может быть несовершенной для сравнения множеств Парето, которые имеют различные распределения точек. Тем не менее, она остается важным инструментом для сравнения алгоритмов оптимизации и оценки их эффективности. Гиперобъем вычисляется следующим образом:

$$HV(L, r) = \mu \left( \bigcup_{x \in L} [r_1; f_1(x)] \times \dots \times [r_n; f_n(x)] \right),$$

где  $\mu(\cdot)$  – мера Лебега,  $r = (r_1, \dots, r_n)$  – опорная точка,  $L \subseteq D$ .

В процессе тестирования метода для каждого примера при определенных параметрах (табл. 1, 3, 5) были вычислены 10 решений при  $n = 10$ . В табл. 2, 4, 6 занесены следующие значения:

$$IGD_{\min} = \min_{1 \leq i \leq 10} IGD_i, \quad HV_{\min} = \min_{1 \leq i \leq 10} |HV_i - HV^P|,$$

$$IGD_{\max} = \max_{1 \leq i \leq 10} IGD_i, \quad HV_{\max} = \max_{1 \leq i \leq 10} |HV_i - HV^P|,$$

$$\overline{IGD} = \frac{1}{10} \sum_{i=1}^{10} IGD_i, \quad \overline{HV} = \frac{1}{10} \sum_{i=1}^{10} |HV_i - HV^P|,$$

где  $HV^P$  – гиперобъем точной границы Парето.

**Пример 1** ( $ZDT1$ ). Частные целевые функции:

$$f_1(x) = x_1,$$

$$f_2(x) = g(x) \cdot \left( 1 - \sqrt{\frac{x_1}{g(x)}} \right),$$



где  $g(x) = 1 + 9 \sum_{i=2}^n \frac{x_i}{n-1}$ ,  $n \geq 2$ , при этом  $0 \leq x_i \leq 1$ ,  $i = 1, \dots, n$ .

Таблица 1

### Параметры метода для ZDT1

№ теста	$M$	$R$	$K$	$\xi$	$q$
1	50	200	100	0,5	0,5
2	100	300	200	0,5	0,5
3	100	300	200	0,85	0,1
4	200	400	200	0,85	0,1
5	200	500	300	0,85	0,1

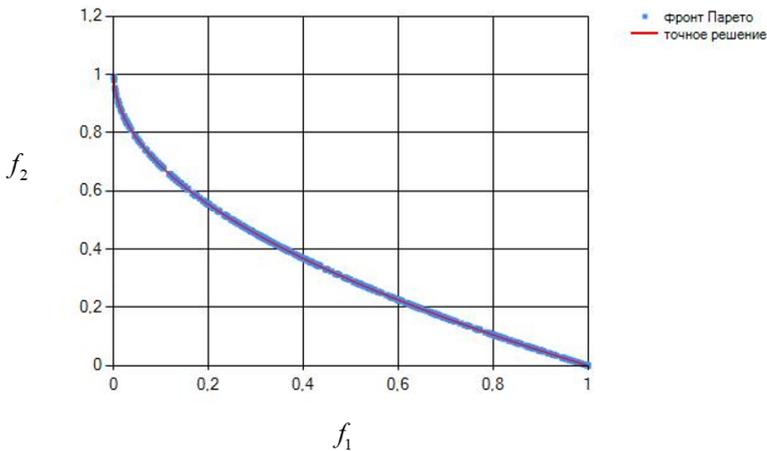


Рис. 1. Фронт Парето найденного решения для ZDT1 (тест № 5)

Таблица 2

### Значения метрик для ZDT1

№ теста	$IGD_{\min}$	$IGD_{\max}$	$\overline{IGD}$	$HV_{\min}$	$HV_{\max}$	$\overline{HV}$
1	0,29530	0,79670	0,43320	0,37651	0,65736	0,46522
2	0,01438	0,17763	0,06773	0,01778	0,24343	0,09253
3	0,01134	0,01549	0,01286	0,01273	0,01928	0,01517
4	0,00436	0,00820	0,00597	0,00179	0,00767	0,00428
5	0,00274	0,00610	0,00417	0,00007	0,00465	0,00189

**Пример 2 (ZDT2).** Частные целевые функции:

$$f_1(x) = x_1,$$



$$f_2(x) = g(x) \cdot \left( 1 - \left( \frac{x_1}{g(x)} \right)^2 \right),$$

где  $g(x) = 1 + 9 \sum_{i=2}^n \frac{x_i}{n-1}$ ,  $n \geq 2$ , при этом  $0 \leq x_i \leq 1$ ,  $i = 1, \dots, n$ .

Таблица 3

**Параметры метода для ZDT2**

№ теста	$M$	$R$	$K$	$\xi$	$q$
1	50	200	100	0,5	0,5
2	100	300	200	0,5	0,5
3	100	300	200	0,85	0,1
4	200	400	200	0,85	0,1
5	200	500	300	0,85	0,1

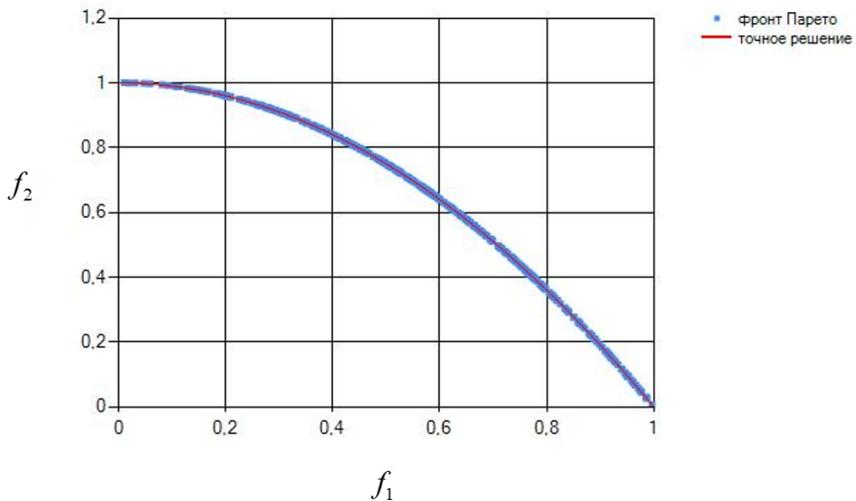


Рис. 2. Фронт Парето найденного решения для ZDT2 (тест № 5)

Таблица 4

**Значения метрик для ZDT2**

№ теста	$IGD_{\min}$	$IGD_{\max}$	$\overline{IGD}$	$HV_{\min}$	$HV_{\max}$	$\overline{HV}$
1	0,16477	0,46240	0,28424	0,17973	0,31477	0,24398
2	0,02469	0,09771	0,05939	0,03049	0,19172	0,07094
3	0,04961	0,06666	0,05722	0,01019	0,06985	0,03391
4	0,00652	0,01595	0,00997	0,00513	0,01943	0,01030
5	0,00286	0,00711	0,00347	0,00053	0,00781	0,00291



**Пример 3 (ZDT3).** Частные целевые функции:

$$f_1(x) = x_1,$$

$$f_2(x) = g(x) \cdot \left( 1 - \sqrt{\frac{x_1}{g(x)}} - \frac{x_1}{g(x)} \sin(10\pi x_1) \right),$$

где  $g(x) = 1 + 9 \sum_{i=2}^n \frac{x_i}{n-1}$ ,  $n \geq 2$ , при этом  $0 \leq x_i \leq 1$ ,  $i = 1, \dots, n$ .

Таблица 5

**Параметры метода для ZDT3**

№ теста	$M$	$R$	$K$	$\xi$	$q$
1	50	200	100	0,5	0,5
2	100	300	200	0,5	0,5
3	100	300	200	0,85	0,1
4	200	400	200	0,85	0,1
5	200	500	300	0,85	0,1

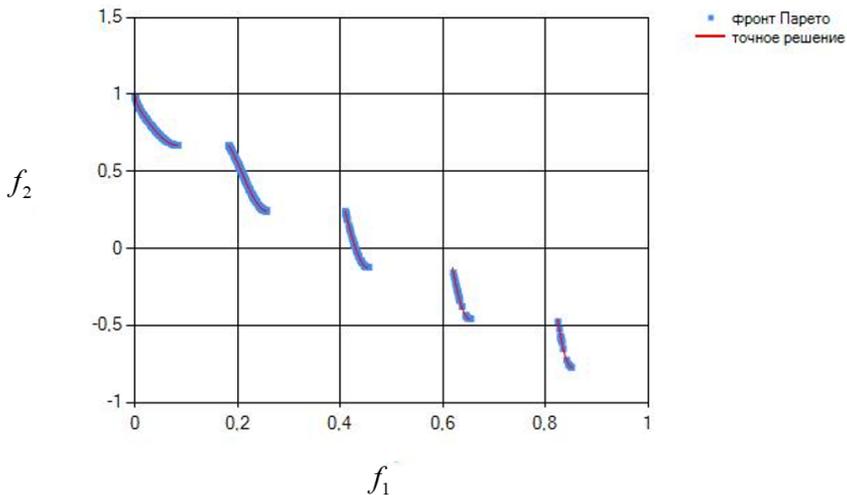


Рис. 3. Фронт Парето найденного решения для ZDT3 (тест № 5)

Таблица 6

**Значения метрик для ZDT3**

№ теста	$IGD_{\min}$	$IGD_{\max}$	$\overline{IGD}$	$HV_{\min}$	$HV_{\max}$	$\overline{HV}$
1	0,12370	0,33840	0,18056	0,24215	0,49522	0,32146
2	0,12253	0,27157	0,23666	0,23819	0,64079	0,53523

№ теста	$IGD_{\min}$	$IGD_{\max}$	$\overline{IGD}$	$HV_{\min}$	$HV_{\max}$	$\overline{HV}$
3	0,02940	0,26720	0,07840	0,07319	0,63168	0,18812
4	0,00764	0,01315	0,01035	0,02060	0,03533	0,02799
5	0,00632	0,00723	0,00664	0,01677	0,01997	0,01814

## 6. ЗАДАЧА ОПТИМИЗАЦИИ ИНВЕСТИЦИОННОГО ПОРТФЕЛЯ

Будем называть инвестиционным портфелем совокупность купленных инвестором активов. Он задается вектором относительных весов  $x = (x_1, \dots, x_n)^T$ . Каждый инвестор стремится собрать такой портфель ценных бумаг, который обеспечит наибольший доход  $E(x) = \sum_{i=1}^n m_i x_i$  при минимальном риске  $V(x) = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_i x_j$ , где  $m = (m_1, \dots, m_n)^T$  – вектор математических ожиданий (вектор средних доходностей),

$$C = \begin{pmatrix} c_{11} & \dots & c_{1n} \\ \vdots & & \vdots \\ c_{n1} & \dots & c_{nn} \end{pmatrix} \text{ – ковариационная матрица доходностей.}$$

Тогда многокритериальную задачу оптимизации инвестиционного портфеля можно записать следующим образом:

$$\begin{aligned} E(x) &= \sum_{i=1}^n m_i x_i \rightarrow \max, \\ V(x) &= \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_i x_j \rightarrow \min, \\ x_i &\geq 0, \quad i = 1, \dots, n, \\ e^T x &= 1, \end{aligned}$$

где  $e = (1, \dots, 1)^T$ .

В качестве примера рассматривается портфель ценных бумаг, состоящий из акций десяти ведущих российских компаний: ВТБ (VTBR), Газпром (GAZP), Лукойл (LKOH), Аэрофлот (AFLT), Сургутнефтегаз (SNGS), Мечел (MTLR), Алроса (ALRS), Яндекс (YNDX), Ютэйр (UTAR), Магнит (MGNT). Информация о средних доходностях (табл. 7) и ковариациях (табл. 8) акций представлена на момент закрытия за период с апреля 2023 года по апрель 2024 года.

Таблица 7

### Средние доходности активов

Актив	Доходность
ВТБ (VTBR)	-0,00731
Газпром (GAZP)	0,02070



Актив	Доходность
Роснефть (ROSN)	0,03487
Аэрофлот (AFLT)	0,01402
Сургутнефтегаз (SNGS)	0,06578
Мечел (MTLR)	0,041187
Алроса (ALRS)	0,017306
Яндекс (YNDX)	0,065089
Ютэйр (UTAR)	0,078605
Магнит (MGNT)	0,051897

Таблица 8

## Ковариации активов

	VTBR	GAZP	ROSN	AFLT	SNGS	MTLR	ALRS	YNDX	UTAR	MGNT
1	0,002305	0,000286	-0,00128	0,00159	0,00243	0,002187	0,003015	-0,0003	-0,00268	0,001998
2		0,00157	9,95E-05	0,000585	-0,00053	0,001445	0,001618	0,00122	-0,00077	0,000408
3			0,002899	-0,00104	-0,00126	0,000394	-0,00021	0,000238	0,003911	0,000349
4				0,005949	0,00227	-0,00059	0,004666	0,004885	0,003723	0,000895
5					0,012254	-0,00166	0,002244	-0,00216	0,001428	0,004201
6						0,015316	0,002534	-0,00044	-0,00112	0,002839
7							0,007566	0,003246	0,001021	0,003663
8								0,007368	0,004928	0,000278
9									0,054013	0,001045
10										0,006855

Результаты решения задачи представлены в табл. 10 и на рис. 4. Параметры, при которых было вычислено решение, приведены в табл. 9.

Таблица 9

## Параметры метода

$M$	$R$	$K$	$\xi$	$q$	$\varepsilon$
600	800	1500	0,85	0,1	0,000153

Таблица 10

## Веса ценных бумаг, значения дохода и риска

$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$	$x_9$	$x_{10}$	$E$	$V$
0,09	0,05	0,05	0,08	0,31	0,06	0,06	0,2	0,08	0,02	0,04645	0,05502
0,01	0,13	0,03	0,09	0,49	0,05	0	0,13	0,02	0,05	0,04942	0,066
0,04	0,15	0,23	0,02	0,15	0,03	0,06	0,07	0,03	0,22	0,03018	0,03085
0,02	0	0,14	0,07	0,06	0,06	0,06	0,09	0,46	0,04	0,05571	0,12077
0,06	0,13	0,02	0,01	0,19	0,01	0,1	0,25	0,2	0,03	0,04987	0,06808
0,03	0,08	0,09	0,04	0,42	0,04	0,02	0,12	0,09	0,07	0,04899	0,0594



$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$	$x_9$	$x_{10}$	$E$	$V$
0,03	0,14	0,17	0,07	0,21	0,1	0,08	0,05	0,08	0,07	0,03872	0,04231
0,04	0,04	0,32	0,06	0,14	0,04	0,04	0,16	0,12	0,04	0,0435	0,04787
0,05	0,07	0,05	0,03	0,36	0,03	0,05	0,18	0,1	0,08	0,04827	0,05731
0	0,29	0,11	0,07	0,14	0,05	0,05	0,16	0,09	0,04	0,04018	0,04449
0,09	0,19	0,25	0,1	0,07	0,1	0,04	0,1	0,05	0,01	0,03313	0,03695
0,07	0,17	0,24	0,03	0,13	0,04	0,09	0,11	0,01	0,11	0,03124	0,03408
0,03	0,06	0,09	0,02	0,24	0,05	0,02	0,23	0,13	0,13	0,0477	0,05579
0,02	0,11	0,11	0,05	0,22	0,06	0,03	0,12	0,23	0,05	0,05009	0,06854
0,07	0,05	0,21	0,01	0,24	0,09	0,06	0,05	0,03	0,19	0,03392	0,03826
0,12	0,11	0,17	0,07	0,15	0,13	0,04	0,1	0,02	0,09	0,03264	0,03608
0,09	0,12	0,15	0	0,12	0,06	0,07	0,15	0,09	0,15	0,03587	0,04009
0	0,19	0,06	0,05	0,05	0,05	0,01	0,22	0,34	0,03	0,05332	0,09371
0,01	0,08	0,04	0,07	0,16	0,17	0,05	0,26	0,07	0,09	0,04505	0,05207
0	0,14	0,12	0,06	0,13	0,17	0,08	0,12	0,08	0,1	0,03875	0,04315
0,04	0,09	0,21	0,04	0,25	0,03	0,04	0,12	0,1	0,08	0,04348	0,04597
0,02	0,17	0,04	0	0,15	0,04	0,08	0,29	0,2	0,01	0,05216	0,06949
0,02	0,02	0,39	0,1	0,19	0,05	0,03	0,08	0,11	0,01	0,04444	0,04831
0,06	0,08	0,08	0,05	0,34	0,02	0,01	0,15	0,14	0,07	0,04921	0,06031
0,07	0,1	0,21	0,01	0,11	0,07	0,07	0,18	0,11	0,07	0,04053	0,0448
0,07	0,14	0,06	0,03	0,27	0,01	0,07	0,19	0,11	0,05	0,0454	0,05297
0,02	0,06	0,13	0,04	0,13	0,14	0,08	0,16	0,06	0,18	0,03679	0,04084
0,04	0,17	0,08	0,05	0,14	0,17	0,06	0,11	0,04	0,14	0,03419	0,03854
0,04	0,04	0,32	0,12	0,2	0,08	0,01	0,05	0,08	0,06	0,03995	0,04401
0,09	0,18	0,09	0,06	0,21	0,08	0,06	0,1	0,06	0,07	0,03621	0,04042
0,02	0,17	0,15	0,05	0,18	0,03	0,05	0,14	0,08	0,13	0,03862	0,041
0,07	0,1	0,11	0,07	0,22	0,12	0,07	0,11	0,08	0,05	0,04061	0,04487
0,07	0,12	0,05	0,05	0,2	0,11	0,01	0,2	0,13	0,06	0,04587	0,05379
0,03	0,04	0,02	0,05	0,26	0,1	0,11	0,25	0,07	0,07	0,04706	0,05566

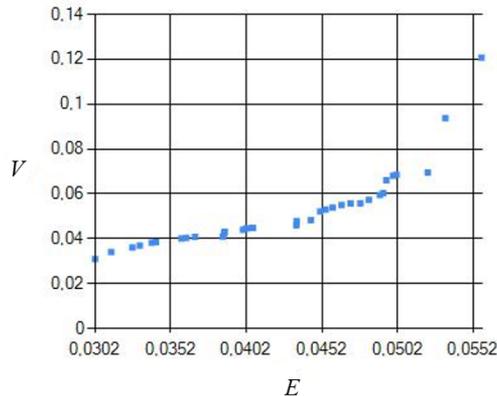


Рис. 4. График зависимости дохода и риска



Заметим, что представленный график является решением задачи, в которой одну из частных целевых функций (функцию дохода) нужно максимизировать, а другую (функцию риска) минимизировать, что отличается от общей постановки задачи, где у всех целевых функций необходимо найти минимум. Для решения задачи оптимизации инвестиционного портфеля была изменена функция дохода, путем умножения ее на  $-1$ , чтобы свести задачу к общей постановке.

## 7. ЗАКЛЮЧЕНИЕ

В статье предложен алгоритм решения многокритериальных задач оптимизации, основанный на методе муравьиных колоний в совокупности с процедурой недоминируемой сортировки. Приведены решения трех модельных примеров. Рассмотрена многокритериальная задача оптимизации инвестиционного портфеля. Показано, что предложенный алгоритм способен находить решения хорошего качества при приемлемых вычислительных затратах.

### *Литература*

1. *Gunantara N.* A review of multi-objective optimization: Methods and its applications, *Cogent Engineering*. 2018. Vol. 5. No. 1, 1502242, DOI: 10.1080/23311916.2018.1502242
2. *Tian Y. et al.* An evolutionary algorithm for large-scale sparse multiobjective optimization problems // *IEEE Transactions on Evolutionary Computation*. 2019. Vol. 24. No. 2. P. 380–393.
3. *Zuo M. et al.* A differential evolution algorithm with the guided movement for population and its application to interplanetary transfer trajectory design // *Engineering Applications of Artificial Intelligence*. 2022. Vol. 110. 104727.
4. *Liu Z.Z., Wang Y., Wang B.C.* Indicator-based constrained multiobjective evolutionary algorithms // *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. 2019. Vol. 51. No. 9. P. 5414–5426.
5. *Panichella A.* An improved Pareto front modeling algorithm for large-scale many-objective optimization // *Proceedings of the Genetic and Evolutionary Computation Conference*. 2022. P. 565–573.
6. *Пантелеев А.В., Крючков А.Ю.* Модификация метаэвристического метода фейерверков для задач многокритериальной оптимизации на основе недоминируемой сортировки // *Научный вестник Московского государственного университета гражданской авиации*. 2019. Т. 22. № 3. С. 67–78.
7. *Пантелеев А.В., Крючков А.Ю.* Разработка и применение многокритериального метода фейерверков в задаче стабилизации движения искусственного спутника по круговой орбите // *Моделирование и анализ данных*. 2019. № 3. С. 43–57.
8. *Dorigo M., Socha K.* Ant colony optimization for continuous domains // *Eur. J. of Operational Research*. 2008. Vol. 185. P. 1155–1173.
9. *Dorigo M., Stutzle T.* Ant colony optimization. MIT Press, 2004.
10. *Пантелеев А.В., Скавинская Д.В.* Метаэвристические алгоритмы глобальной оптимизации. М.: Вузовская книга, 2019.
11. *Ming F. et al.* A constraint-handling technique for decomposition-based constrained many-objective evolutionary algorithms // *IEEE Transactions on Systems, Man, and Cybernetics Systems*. 2023. Vol. 53. P. 7783–7793.



12. Takahama T., Sakai S. Constrained optimization by the  $\varepsilon$  constrained differential evolution with gradient-based mutation and feasible elites / *Proc. IEEE Int. Conf. Evol. Comput.* 2006. P. 1–8.
13. Zuo M., Xue Y. Population feasibility state guided autonomous constrained multi-objective evolutionary optimization // *Mathematics*. 2024. Vol. 12. No. 6. 913.
14. Morovati V., Pourkarimi L. Extension of Zoutendijk method for solving constrained multi-objective optimization problems // *European Journal of Operational Research*. 2019. Vol. 273. No. 1. P. 44–57.
15. Zapotecas-Martínez S., Ponsich A. Constraint handling within MOEA/D through an additional scalarizing function // *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*. 2020. P. 595–602.
16. Tian Y. et al. Balancing objective optimization and constraint satisfaction in constrained evolutionary multiobjective optimization // *IEEE Transactions on Cybernetics*. 2021. Vol. 52. No. 9. P. 9559–9572.
17. Бадалова А.Г., Пантелеев А.В. Промышленный риск-менеджмент. М.: Доброе слово, 2018.
18. Zanjirdar M. Overview of portfolio optimization models // *Advances in mathematical finance and applications*. 2020. Vol. 5. No. 4. P. 419–435.
19. Yang M. et al. A Synergistic multi-objective evolutionary algorithm with diffusion population generation for portfolio problems // *Mathematics*. 2024. Vol. 12. No. 9. 1368.



# Development and Application of a Multi-Objective Ant Colony Optimization Method for Portfolio Problem

**Andrei V. Pantelev\***

Moscow Aviation Institute (National Research University) (MAI), Moscow, Russia

ORCID: <https://orcid.org/0000-0003-2493-3617>

e-mail: [avpantelev@inbox.ru](mailto:avpantelev@inbox.ru)

**Natalya S. Popova\*\***

Moscow Aviation Institute (National Research University) (MAI), Moscow, Russia

ORCID: <https://orcid.org/0009-0000-7196-7786>

e-mail: [popovanatalya472@gmail.com](mailto:popovanatalya472@gmail.com)

A numerical method of multi-objective optimization is proposed for an approximate solution of the problem based on the generation of feasible solutions using the continuous ant colony method, non-dominated sorting and the epsilon-constraint technique. Solving a problem means finding the Pareto front. Solutions of typical model examples are given. The applied problem of optimizing an investment portfolio has been solved, in which the initial data are the tabulated average returns and covariance of stocks.

**Keywords:** investment portfolio optimization, multi-objective optimization, metaheuristic methods, non-dominated sorting,  $\varepsilon$ -constraint technique, ant colony optimization, Pareto optimality.

## For citation:

Pantelev A.V., Popova N.S. Development and Application of a Multi-Objective Ant Colony Optimization Method for Portfolio Problem. *Modelirovanie i analiz dannykh = Modelling and Data Analysis*, 2024. Vol. 14, no. 2, pp. 80–97. DOI: <https://doi.org/10.17759/mda.2024140205> (In Russ., abstr. in Engl.).

## References

1. Gunantara N. A review of multi-objective optimization: Methods and its applications, Cogent Engineering. 2018. Vol. 5. No. 1, 1502242, DOI: 10.1080/23311916.2018.1502242
2. Tian Y. et al. An evolutionary algorithm for large-scale sparse multiobjective optimization problems // IEEE Transactions on Evolutionary Computation. 2019. Vol. 24. No. 2. P. 380–393.

\***Andrei V. Pantelev**, Doctor of Physics and Mathematics, Professor, Head of the Department of Mathematics and Cybernetics, Institute “Computer Science and Applied Mathematics”, Moscow Aviation Institute (National Research University) (MAI), Moscow, Russia, ORCID: <https://orcid.org/0000-0003-2493-3617>, e-mail: [avpantelev@inbox.ru](mailto:avpantelev@inbox.ru)

\*\***Natalya S. Popova**, Bachelor’s Degree Student, Institute “Computer Science and Applied Mathematics”, Moscow Aviation Institute (National Research University) (MAI), Moscow, Russia, ORCID: <https://orcid.org/0009-0000-7196-7786>, e-mail: [popovanatalya472@gmail.com](mailto:popovanatalya472@gmail.com)



3. Zuo M. et al. A differential evolution algorithm with the guided movement for population and its application to interplanetary transfer trajectory design // *Engineering Applications of Artificial Intelligence*. 2022. Vol. 110. 104727.
4. Liu Z.Z., Wang Y., Wang B.C. Indicator-based constrained multiobjective evolutionary algorithms // *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. 2019. Vol. 51. No. 9. P. 5414–5426.
5. Panichella A. An improved Pareto front modeling algorithm for large-scale many-objective optimization // *Proceedings of the Genetic and Evolutionary Computation Conference*. 2022. P. 565–573.
6. Panteleev A.V., Kryuchkov A.Yu. Modifikaciya metaevristicheskogo metoda fejervertkov dlya zadach mnogokriterial'noj optimizacii na osnove nedominiruemoj sortirovki // *Nauchnyj vestnik Moskovskogo gosudarstvennogo universiteta grazhdanskoj aviacii*. 2019. Vol. 22. No. 3. P. 67–78. (In Russ.).
7. Panteleev A.V., Kryuchkov A.Yu. Razrabotka i primenenie mnogokriterial'nogo metoda fejervertkov v zadache stabilizacii dvizheniya iskusstvennogo sputnika po krugovoj orbite // *Modelirovanie i analiz dannyh*. 2019. No. 3. P. 43–57. (In Russ.).
8. Dorigo M., Socha K. Ant colony optimization for continuous domains // *Eur. J. of Operational Research*. 2008. Vol. 185. P. 1155–1173.
9. Dorigo M., Stutzle T. *Ant colony optimization*. MIT Press, 2004.
10. Panteleev A.V., Skavinskaya D.V. *Metaevristicheskie algoritmy global'noj optimizacii*. M.: Vuzovskaya kniga, 2019. (In Russ.).
11. Ming F. et al. A constraint-handling technique for decomposition-based constrained many-objective evolutionary algorithms // *IEEE Transactions on Systems, Man, and Cybernetics Systems*. 2023. Vol. 53. P. 7783–7793.
12. Takahama T., Sakai S. Constrained optimization by the  $\epsilon$  constrained differential evolution with gradient-based mutation and feasible elites / *Proc. IEEE Int. Conf. Evol. Comput.* 2006. P. 1–8.
13. Zuo M., Xue Y. Population feasibility state guided autonomous constrained multi-objective evolutionary optimization // *Mathematics*. 2024. Vol. 12. No. 6. 913.
14. Morovati V., Pourkarimi L. Extension of Zoutendijk method for solving constrained multi-objective optimization problems // *European Journal of Operational Research*. 2019. Vol. 273. No. 1. P. 44–57.
15. Zapotecas-Martínez S., Ponsich A. Constraint handling within MOEA/D through an additional scalarizing function // *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*. 2020. P. 595–602.
16. Tian Y. et al. Balancing objective optimization and constraint satisfaction in constrained evolutionary multiobjective optimization // *IEEE Transactions on Cybernetics*. 2021. Vol. 52. No. 9. P. 9559–9572.
17. Badalova A.G., Panteleev A.V. *Promyshlennyj risk-menedzhment*. M.: Dobroe slovo, 2018. (In Russ.).
18. Zanjirdar M. Overview of portfolio optimization models // *Advances in mathematical finance and applications*. 2020. Vol. 5. No. 4. P. 419–435.
19. Yang M. et al. A Synergistic multi-objective evolutionary algorithm with diffusion population generation for portfolio problems // *Mathematics*. 2024. Vol. 12. No. 9. 1368.