

УДК 517.977:519.8

Применение модификации метода, имитирующего поведение стаи мотыльков, для решения задачи оптимального программного управления мобильным роботом

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

Надоров И.С.**

Московский авиационный институт

(национальный исследовательский университет) (МАИ)

г. Москва, Российская Федерация

ORCID: <https://orcid.org/0009-0008-2085-2987>

e-mail: nnadorovivan@gmail.com

В статье предложена модификация метаэвристического метода оптимизации, имитирующего поведение стаи мотыльков, относящегося к группе биоинспирированных методов. Сформирован пошаговый алгоритм и изучена эффективность модифицированного метода на общепринятом наборе тестовых функций многих переменных со сложной структурой поверхностей уровня, на задаче оптимального программного управления с известным точным решением, а также на задаче определения параметров натяжной/компрессионной пружины с ограничениями типа неравенств. Показано преимущество модифицированного метода над оригинальной версией. Продемонстрировано, что метод позволяет найти решения достаточно хорошего качества за приемлемое с практической точки зрения время. Приведено решение прикладной задачи поиска оптимального программного управления мобильным роботом на плоскости при наличии препятствий. Целью управления является достижение заданной конечной точки при минимизации затрачиваемого времени и выполнении условия огибания запретных областей. Для нахождения закона управления как функции времени применялась кусочно-постоянная аппроксимация, позволяющая свести задачу к нахождению конечного числа неизвестных параметров. Рассмотрены решения терминальной задачи быстродействия при различной структуре и параметрах составного функционала качества с помощью последовательного применения



разработанного модифицированного метода, имитирующего поведение стаи мотыльков, метода случайного поиска с последовательной редукцией области исследования и метода перекоммутации. Приведены результаты сравнения с известными решениями, подтвердившие эффективность разработанного алгоритмического и программного обеспечения.

Ключевые слова: глобальная оптимизация, метаэвристические методы, оптимальное управление.

Для цитаты:

Пантелеев А.В., Надоров И.С. Применение модификации метода, имитирующего поведение стаи мотыльков, для решения задачи оптимального программного управления мобильным роботом // Моделирование и анализ данных. 2025. Том 15. № 1. С. 81–109. DOI: 10.17759/mda.2025150105

***Пантелеев Андрей Владимирович**, доктор физико-математических наук, профессор, заведующий кафедрой математической кибернетики, институт «Компьютерные науки и прикладная математика», Московский авиационный институт (Национальный исследовательский университет) (МАИ), г. Москва, Российская Федерация, ORCID: <https://orcid.org/0000-0003-2493-3617>, e-mail: avpanteleev@inbox.ru

****Надоров Иван Сергеевич**, студент, институт «Компьютерные науки и прикладная математика», Московский авиационный институт (Национальный исследовательский университет) (МАИ), г. Москва, Российская Федерация, ORCID: <https://orcid.org/0009-0008-2085-2987>, e-mail: nnadorovivan@gmail.com

1. ВВЕДЕНИЕ

Задачи параметрической оптимизации при наличии интервальных ограничений возникают при решении разнообразных проблем проектирования сложных авиационно-космических систем. При этом целевые функции, отражающие требования проектировщика к системе, часто оказываются многоэкстремальными, недифференцируемыми или вычисляемыми в результате реализации многошаговых вспомогательных алгоритмов. Поэтому одним из требований к применяемым для решения задачи методам оптимизации является возможность покидать окрестности локальных экстремумов в процессе поиска глобального экстремума. Применение традиционных методов оптимизации первого и второго порядков для целевых функций общего вида, как правило, приводит к нахождению локальных экстремумов. В настоящее время в научной литературе большое внимание уделяется разработке приближённых методов глобальной оптимизации, которые позволяют найти решение достаточно хорошего качества за приемлемое время [1–7]. Как правило, выделяются два класса методов нулевого порядка, в которых для определения направления поиска используется только информация о величине целевой функции: стохастические метаэвристические [1–7] и детерминированные [8, 9]. Метаэвристические методы можно условно разделить на пять групп [1]: эволюционные методы; методы, имитирующие физические процессы; мультистартовые методы; методы роевого интеллекта; биоинспирированные и мультиагентные методы. Биоинспирированные методы

на сегодняшний день получили значительное распространение при решении прикладных задач параметрической оптимизации технических систем и законов управления динамическими системами [1, 7]. К ним относится метод, имитирующий поведение стаи мотыльков (Moth-Flame Optimization, MFO) [10], имеющий свои преимущества и недостатки [11]. Разработка новых и усовершенствование существующих методов является актуальным вследствие справедливости теоремы (No free lunch, NFL) [12] о том, что не существует универсального метода, позволяющего наилучшим образом решить общую задачу конечномерной оптимизации. В данной работе предложена модификация метода MFO, продемонстрирована её эффективность для общеизвестных тестовых функций [13], для задач оптимального программного управления, для задач оптимизации с ограничениями типа неравенств, а также преимущество перед оригинальной версией. Модификация метода MFO применена для решения задачи поиска оптимального программного управления мобильным роботом [14, 15], которая может оказаться полезной на этапе эскизного проектирования планетоходов.

2. ПОСТАНОВКА ЗАДАЧИ ОПТИМИЗАЦИИ

Дана целевая функция $f(x) = f(x_1, x_2, \dots, x_n)$, определённая на параллелепипеде в пространстве допустимых решений $D \subseteq R^n$. Требуется найти глобальный условный минимум функции $f(x)$ на множестве D , т. е. такую точку $x^* \in D$, что

$$f(x^*) = \min_{x \in D} f(x),$$

где $x = (x_1, x_2, \dots, x_n)^T$, $D = \{x \mid x_i \in [a_i, b_i], i = 1, 2, \dots, n\}$. Задача поиска максимума целевой функции $f(x)$ сводится к задаче поиска минимума путём замены знака перед функцией на противоположный:

$$f(x^*) = \max_{x \in D} f(x) = -\min_{x \in D} [-f(x)].$$

3. МОДИФИЦИРОВАННЫЙ МЕТОД, ИМИТИРУЮЩИЙ ПОВЕДЕНИЕ СТАИ МОТЫЛЬКОВ. СТРАТЕГИЯ ПОИСКА

Метод мотыльков и источников света (MFO) имитирует поведение мотыльков при их приближении к источникам света. Каждый мотылек из популяции мотыльков реализует движение по спирали к центру – источнику света [10]. В начале создается случайный набор из N_p точек (положений мотыльков) в пространстве допустимых решений, который можно представить следующей матрицей:

$$M(\text{Moth}) = \begin{pmatrix} m_1^1 & \dots & m_n^1 \\ \vdots & \ddots & \vdots \\ m_1^{N_p} & \dots & m_n^{N_p} \end{pmatrix},$$

где m_i^j – значение i -й координаты j -го мотылька, $j = 1, \dots, N_p$, $i = 1, \dots, n$.



Для всех текущих положений мотыльков $m^j = (m_1^j, \dots, m_n^j)^T$ вычисляются значения целевой функции $OM^j = f(m^j)$, $j = 1, \dots, Np$. В результате формируется

$$\text{вектор } OM = \begin{pmatrix} OM^1 \\ OM^2 \\ \vdots \\ OM^{Np} \end{pmatrix}.$$

Помимо мотыльков в решении задачи играют роль источники света – лучшие положения мотыльков, среди всех ранее занятых, с точки зрения достижения наименьшего значения целевой функции. Поэтому число источников света тоже равно Np . Аналогично мотылькам формируется матрица координат источников света и вектор значений целевой функции, упорядоченный по возрастанию (наилучший источник света имеет номер 1):

$$F(\text{Flame}) = \begin{pmatrix} f_1^1 & \dots & f_n^1 \\ \vdots & \ddots & \vdots \\ f_1^{Np} & \dots & f_n^{Np} \end{pmatrix}, \quad OF = \begin{pmatrix} OF^1 \\ OF^2 \\ \vdots \\ OF^{Np} \end{pmatrix},$$

где f_i^j – значение i -й координаты j -го источника света, $j = 1, \dots, Np$, $i = 1, \dots, n$, OF^j – значение целевой функции в точке $f^j = (f_1^j, \dots, f_n^j)^T$, $j = 1, \dots, Np$, соответствующей положению j -го источника света.

В процессе поиска мотыльки меняют свое положение: двигаются по спирали к источникам света по каждой координате. Спираль должна удовлетворять трем требованиям: начальная точка спирали начинается с текущего положения мотылька, в полюсе спирали расположен источник света, точки спирали не должны выходить за пределы множества допустимых решений D . В качестве такой спирали можно взять логарифмическую спираль (рис. 1):

$$S(m^j, f^j) = H^j e^{st} \cos(2\pi t) + f^j,$$

где m^j – положение j -го мотылька, f^j – положение j -го источника света, $H^j = |f^j - m^j|$ – вектор покоординатных расстояний между j -м источником света и j -м мотыльком, s – константа для определения формы спирали, $t \in [r-1, 0]$ – случайное число, определяющее положение мотыльков относительно спирали, задаваемое равномерным законом распределения. Число r – константа, значение которой уменьшается с ростом числа итераций по линейному закону от -1 до -2 .

Модификация метода MFO (MMFO) заключается в рассмотрении сразу нескольких точек на спирали, имитирующих передвижение мотылька к источнику света, в процессе которого реализуется стратегия поиска экстремума. Если одна из таких точек окажется за пределами множества допустимых решений, вместо неё можно

рассматривать случайную точку в пределах D . В качестве следующего положения мотылька следует выбирать то положение, в котором достигается наименьшее значение целевой функции.

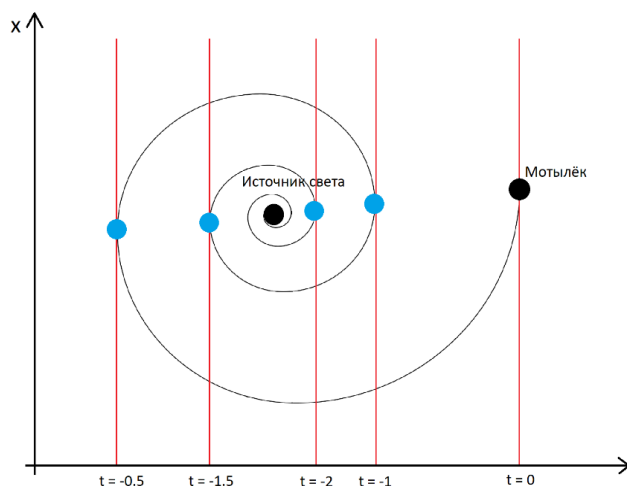


Рис. 1. Логарифмическая спираль

Продвижение мотыльков обеспечивает поиск лучшего решения, а источники света постоянно меняют свое положение. Это происходит следующим образом: рассмотрим положение первого мотылька, сравнивая значения целевой функции в точке его положения и в точке положения последнего источника света (в начале процесса поиска он имеет номер N_p). Если выполняется условие $OM^1 < OF^{last}$, т.е. положение мотылька лучше, то за положение последнего источника света принимается текущее положение первого мотылька: $f^{last} := m^1$ и заменяется значение целевой функции последнего источника света на соответствующее значение для мотылька $OF^{last} := OM^1$. Затем следует упорядочить координаты вектора положений источников света $sort(OF)$ по возрастанию значений целевой функции. Заметим, что при компьютерной реализации для реализации процедуры упорядочения предлагается использовать быструю сортировку Хоара [16]. Описанная последовательность действий повторяется для всех положений мотыльков.

Чтобы мотыльки не находили быстро локальные минимумы и не прекращали поиск, следует связать каждого мотылька ровно с одним источником света: первый летит по спирали к первому источнику, второй – ко второму, последний – к последнему. Учитывая, что источники постоянно меняются и сортируются, произойдет обеспечение тщательного исследования множества допустимых решений. Однако при длительном поиске этот процесс может ослабить целенаправленное движение к минимуму. Чтобы исключить такую возможность, число источников света сокращается постепенно по следующей формуле



$$No_f = \text{round}(Np - k \frac{Np-1}{T}),$$

где $\text{round}(x)$ – округление до ближайшего целого числа, k – номер текущей итерации, Np – первоначальное число источников света, T – максимальное число итераций.

Сокращение числа источников происходит путем удаления из рассмотрения худшего на данный момент источника, т.е. последнего в векторе OF . Мотылек, ранее связанный с этим источником, теперь связывается с источником, бывшим ранее предпоследним. Таким образом, достигается баланс между обширным исследованием множества допустимых решений и движением к минимуму.

Процесс поиска оканчивается при достижении максимального числа итераций или заданного предельного времени работы алгоритма.

4. МОДИФИЦИРОВАННЫЙ МЕТОД, ИМИТИРУЮЩИЙ ПОВЕДЕНИЕ СТАИ МОТЫЛЬКОВ. АЛГОРИТМ

Шаг 1. Задать параметры метода: $Np > 0$ – размер популяции мотыльков; $n > 0$ – размерность задачи; s – параметр логарифмической спирали; $T > 0$ – максимальное число итераций; $\tau > 0$ – количество возможных положений мотылька на спирали.

Шаг 2. Генерация начальной популяции $\{m^1, \dots, m^{Np}\}$.

2.1. Создание популяции и заполнение матрицы положений мотыльков:

$$m_i^j = (b_i - a_i)x_i^j + a_i, \quad j=1, \dots, Np, \quad i=1, \dots, n,$$

где x_i^j – i -я координата случайного вектора x^j , описываемая равномерным законом распределения на отрезке $[0, 1]$, a, b – векторы границ множества допустимых решений,

$$M = \begin{pmatrix} m_1^1 & \dots & m_n^1 \\ \vdots & \ddots & \vdots \\ m_1^{Np} & \dots & m_n^{Np} \end{pmatrix}.$$

2.2. Положить $k=1$, $No_f = Np$.

Шаг 3. Изменение числа источников света:

$$No_f = \text{round}(Np - k \frac{Np-1}{T}),$$

где $\text{round}(x)$ – округление до ближайшего целого числа.

Если число No_f уменьшилось и $k \leq T$, то удалить последний элемент вектора OF и последнюю строку матрицы F .

Шаг 4. Получить вектор OM значений целевой функции для всех мотыльков

из текущей популяции: $OM = \begin{pmatrix} OM^1 \\ OM^2 \\ \vdots \\ OM^{Np} \end{pmatrix}$, где $OM^j = f(m^j)$, $j = 1, \dots, Np$.

Шаг 5. Получение матрицы положений источников света и соответствующего вектора значений целевой функции.

Если $k = 1$, то упорядочить строки вектора OM по возрастанию величины целевой функции. При этом получить вектор $OF = \text{sort}(OM)$ и соответствующую ему матрицу положений источников света $F = \text{sort}(M)$:

$$OF = \begin{pmatrix} OF^1 \\ OF^2 \\ \vdots \\ OF^{Np} \end{pmatrix}, \quad F = \begin{pmatrix} f_1^1 & \dots & f_n^1 \\ \vdots & \ddots & \vdots \\ f_1^{Np} & \dots & f_n^{Np} \end{pmatrix}.$$

Если $k > 1$, то сформировать блочную матрицу следующего вида $P = \|p_i^j\| = \begin{pmatrix} F \\ M \end{pmatrix}$.

Строки матрицы P соответствуют положениям источников света и мотыльков.

Затем упорядочить строки матрицы P по возрастанию значений целевой функции, соответствующих строкам матрицы.

Составить матрицу F следующим образом: $f_i^j = p_i^j$, $i = 1, \dots, n$ $j = 1, \dots, No_f$ и получить вектор

$$OF = \begin{pmatrix} OF^1 \\ OF^2 \\ \vdots \\ OF^{No_f} \end{pmatrix},$$

где $OF^j = f(f^j)$, $j = 1, \dots, No_f$.

Если $k < T$, перейти к шагу 6, иначе поиск завершить и найти приближенное решение $x^* = f^1$, $f(x^*) = OF^1$.

Шаг 6. Для каждого j -го мотылька определить новое положение:

6.1. Найти r ; t_c , $H^j = (H_1^j, \dots, H_n^j)^T$:

если $k = 1$, то положить $r = -1$, а если $k > 1$, то $r = -1 - \frac{k}{T}$.

Сгенерировать наборы равномерно распределённых случайных чисел $t_c = (t_{\tilde{n},1}, t_{\tilde{n},2}, \dots, t_{\tilde{n},n})$, $t_{\tilde{n},i} \in [r-1, 0]$, $\tilde{n} = 1, \dots, \tau$, $i = 1, \dots, n$.

Если $j \leq No_f$, то $H_i^j = |f_i^j - m_i^j|$, $i = 1, \dots, n$. Если $j > No_f$, то $H_i^j = |f_i^{No_f} - m_i^j|$, $i = 1, \dots, n$.

6.2. Найти новое положение мотылька.

Определить $m^{j(\tilde{n})} := H^j e^{st_{\tilde{n}}} \cos(2\pi t_{\tilde{n}}) + f^j$, $j = 1, \dots, Np$; $\tilde{n} = 1, \dots, \tau$.



Если не выполняется условие $a_i \leq m_i^{j(n)} \leq b_i \quad i=1, \dots, n$, то сгенерировать значение $m_i^{j(n)}$ на отрезке $[a_i, b_i]$ при помощи равномерного закона распределения.

Выбрать $m^j := \arg \min \{f(m^{j(1)}), \dots, f(m^{j(\tau)})\}$.

Шаг 7. Положить $k := k + 1$ и перейти к шагу 3.

5. ТЕСТИРОВАНИЕ МОДИФИЦИРОВАННОГО МЕТОДА, ИМИТИРУЮЩЕГО ПОВЕДЕНИЕ СТАИ МОТЫЛЬКОВ

На основе изложенного алгоритма разработана программа поиска глобального экстремума функций модифицированным методом MFO. Для ее создания использовалась среда разработки Microsoft Visual Studio, язык программирования C#. Характеристики компьютера, на котором проводились вычисления: процессор Intel Core i7–10750H CPU @ 2.60GHz, 6 ядер, RAM: 16 Gb.

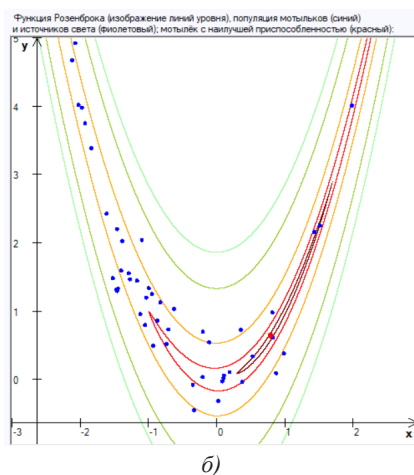
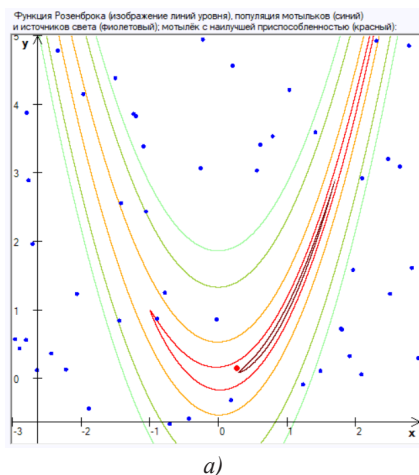
Пример 1. Найти глобальный максимум функции Розенброка

$$f(x, y) = -100(y - x^2)^2 - (1 - x)^2.$$

Зададим множество допустимых решений $x \in [-3; 3], y \in [-1; 5]$. Выберем следующие параметры алгоритма: размер популяции мотыльков $Np = 50$; параметр логарифмической спирали $s = 1$; максимальное число итераций $T = 30$; количество возможных положений мотылька на спирали $\tau = 4$.

На рис. 2 представлена популяция на начальной ($k=1$), промежуточных ($k=3, k=9$) и конечной ($k=30$) итерациях. Красная точка соответствует наилучшему решению на текущей итерации.

Результаты работы алгоритма: наилучшее решение $(x^*, y^*) = (1, 0095; 1, 0196)$; значение целевой функции $f(x^*, y^*) = -0,0001$; отклонение от точного решения $\Delta = 0,0001$. Графики изменения наилучшего и среднего по популяции значений целевой функции представлены на рис. 3.



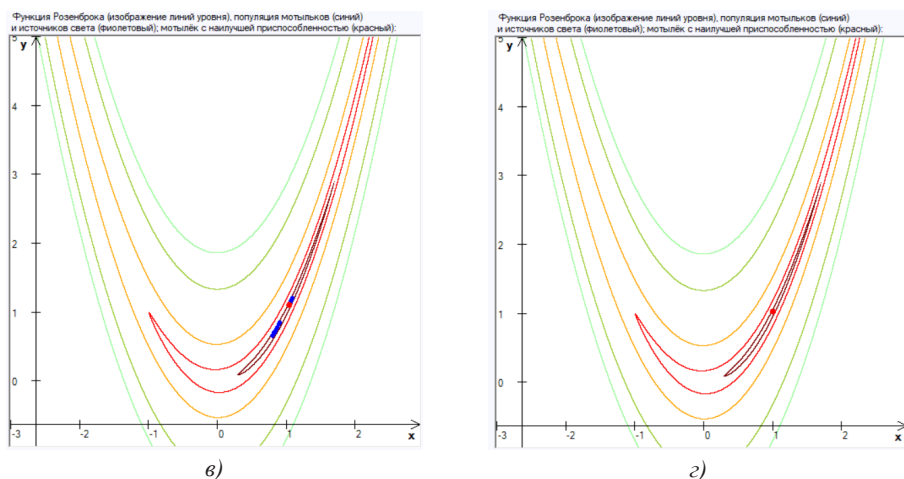


Рис. 2. Начальная, промежуточные и конечная итерации. Пример 1

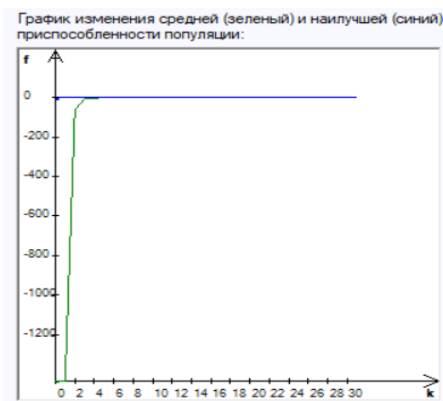


Рис. 3. Графики изменения наилучшего (синий) и среднего значения (зеленый) целевой функции. Пример 1

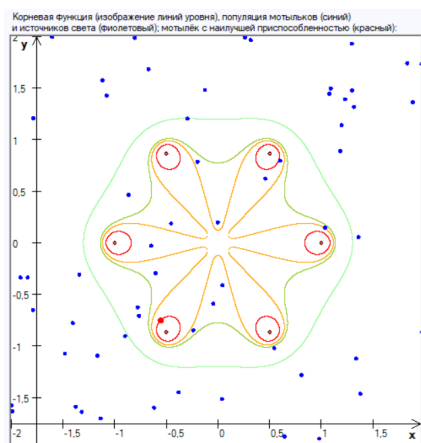
Пример 2. Найти глобальный максимум корневой функции $f(z) = \frac{1}{1 + |z^6 - 1|}$, $z \in \mathbb{C}$, $z = x + iy$.

Зададим множество допустимых решений $x \in [-2; 2]$, $y \in [-2; 2]$. Выберем следующие параметры алгоритма: размер популяции мотыльков $Np = 60$; параметр логарифмической спирали $s = 2$; максимальное число итераций $T = 30$; количество возможных положений мотылька на спирали $\tau = 8$.

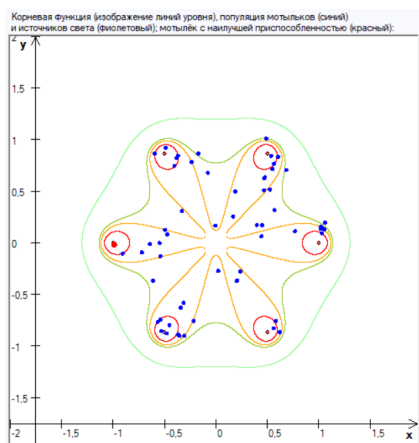
На рис. 4 представлена популяция на начальной ($k = 1$), промежуточных ($k = 3$, $k = 8$) и конечной ($k = 30$) итерациях. Красная точка соответствует наилучшему решению на текущей итерации. Результаты работы алгоритма: наилучшее решение $(x^*, y^*) = (-0,5; 0,866)$; значение целевой функции $f(x^*, y^*) = 0,999997$;



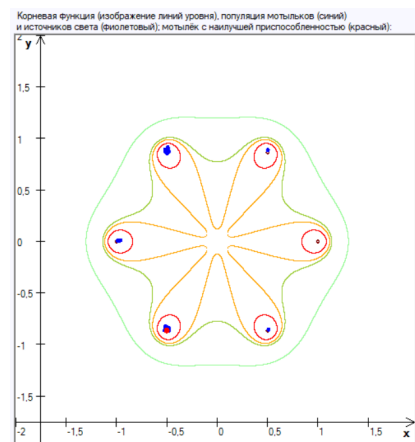
отклонение от точного решения $\Delta = 3 \cdot 10^{-6}$. Графики изменения наилучшего и среднего по популяции значений целевой функции представлены на рис. 5.



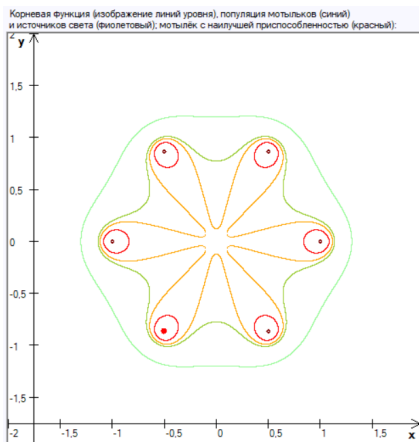
а)



б)



в)



г)

Рис. 4. Начальная, промежуточные и конечная итерации. Пример 2

Пример 3. Найти глобальный максимум функции Швепеля

$$f(x, y) = x \sin(\sqrt{|x|}) + y \sin(\sqrt{|y|}).$$

Зададим множество допустимых решений $x \in [-500; 500], y \in [-500; 500]$. Выберем следующие параметры алгоритма: размер популяции мотыльков $Np = 70$; параметр логарфмической спирали $s = 1$; максимальное число итераций $T = 40$; количество возможных положений мотылька на спирали $\tau = 8$.

На рис. 6 представлена популяция на начальной ($k = 1$), промежуточных ($k = 4, k = 7$) и конечной ($k = 40$) итерациях. Красная точка соответствует наилучшему

решению на текущей итерации. Результаты работы алгоритма: наилучшее решение $(x^*, y^*) = (420, 9699; 420, 9679)$; значение целевой функции $f(x^*, y^*) = 837,965774$; отклонение от точного решения $\Delta = 2,6 \cdot 10^{-5}$. Графики изменения наилучшего и среднего по популяции значений целевой функции представлены на рис. 7.

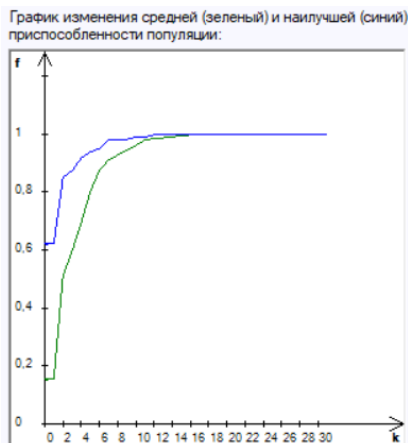


Рис. 5. Графики изменения наилучшего (синий) и среднего значения (зеленый) целевой функции. Пример 2

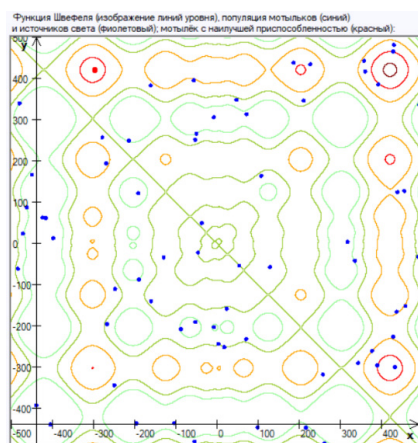
6. ДВУМЕРНАЯ ЗАДАЧА БОЛЬЦА С ИЗВЕСТНЫМ ТОЧНЫМ РЕШЕНИЕМ

Для оценки эффективности модификации алгоритм применялся для решения задачи оптимального программного управления. Поведение модели объекта управления описывается системой разностных уравнений:

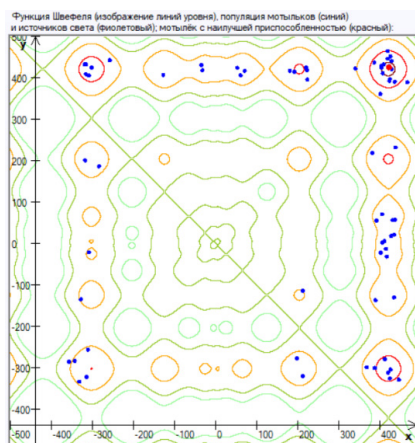
$$\begin{aligned} x_1(t+1) &= x_2(t), \\ x_2(t+1) &= 2x_2(t) - x_1(t) + \frac{1}{N^2}u(t), \end{aligned}$$

где $x \in R^2$, $u \in R$, $t = 0, 1, \dots, N-1$. Задано ограничение на управление: $0 \leq u \leq 100$.

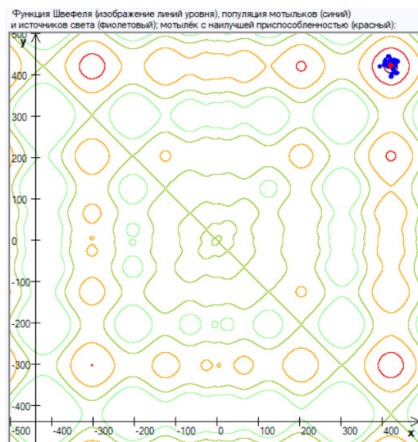
Начальное состояние: $x(0) = (0; 0)^T$.



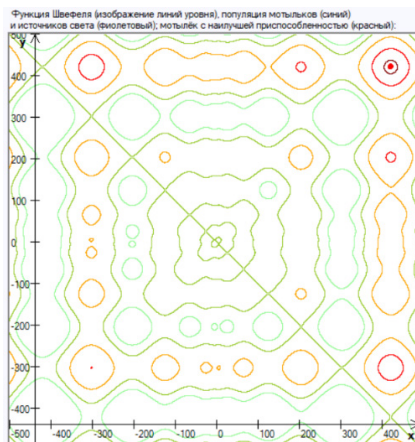
а)



б)



в)



г)

Рис. 6. Начальная, промежуточные и конечная итерации. Пример 3

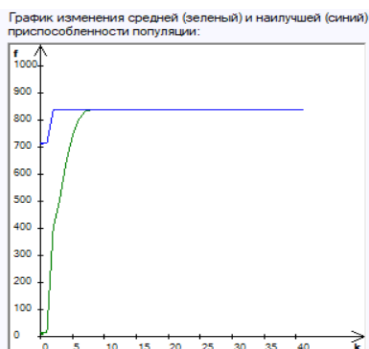


Рис. 7. Графики изменения наилучшего (синий) и среднего значения (зеленый) целевой функции. Пример 3

На множестве допустимых процессов задан функционал

$$I = -x_1(N) + \frac{1}{2N} \sum_{t=0}^{N-1} u^2(t).$$

Требуется найти минимальное значение функционала и оптимальный процесс, на котором это значение достигается.

Сформулированная задача имеет аналитическое решение:

$$u^*(t) = \frac{N-t-1}{N}, \quad \min I = -\frac{1}{3} + \frac{3N-1}{6N^2} + \frac{1}{2N^3} \sum_{t=0}^{N-1} t^2.$$

В монографии [7] эта задача решена при $N = 10$, $\min I = -0,1425$. Авторы получили следующие значения функционала качества методами, имитирующими поведение стаи окуней и стаи синиц, $I_{PSS}^* = -0,142499999999983$ и $I_{TFO}^* = -0,142499999999985$ за 3,69 сек и 0,49 сек соответственно.

С помощью метода, имитирующего поведение стаи мотыльков, с параметрами $Np = 300$, $s = 0,2$, $T = 300$ получено значение функционала: $I = -0,142381123984817$ за 0,1536818 сек. Применение модификации с параметрами $Np = 300$, $s = 0,2$, $T = 300$, $\tau = 16$ привело к следующему результату: $I = -0,1425$ за 2,6901271 сек.

При значительном повышении размерности задачи оптимизации $N = 100$ получены следующие результаты при точном минимальном значении $\min I = -0,164175$: для MFO $I = 39,3240879315545$ за 13,7644471 сек, для MMFO $I = -0,164174999999998$ за 6,7207604483333 мин. Параметры для оригинальной версии: $Np = 1000$, $s = 0,2$, $T = 1000$; параметры для модификации: $Np = 1000$, $s = 0,2$, $T = 1000$, $\tau = 32$.

Полученные результаты свидетельствуют о применимости разработанной модификации для решения задач подобного рода.

7. АНАЛИЗ ЭФФЕКТИВНОСТИ МОДИФИЦИРОВАННОГО МЕТОДА, ИМИТИРУЮЩЕГО ПОВЕДЕНИЕ СТАИ МОТЫЛЬКОВ

В данном разделе приведены результаты статистического анализа и сравнения работы предложенной модификации метода, имитирующего поведение стаи мотыльков, при различных значениях его параметров. Метод применялся к функциям, выбранным из общеизвестного набора тестовых функций [7, 13]. Для каждой функции проводились серии из 100 решений одной и той же задачи с одними и теми же значениями параметров.

Для полученной выборки $\{f^1, f^2, \dots, f^{100}\}$ вычислялись среднее значение отклонения полученного решения от точного $\overline{\Delta f} = \frac{1}{100} \sum_{i=1}^{100} \Delta f_i$, где $\Delta f_i = |f(x^*) - f^i|$;



наименьшее значение отклонения $\Delta f_{best} = \min_i \Delta f_i$; среднеквадратическое отклонение

$\bar{\sigma}_f = \sqrt{\bar{S}_{100}}$, где $\bar{S}_{100} = \frac{1}{100} \sum_{i=1}^{100} (\Delta f^i - \bar{\Delta f})^2$; количество успехов $n_{усп}$ (успехом считалось

попадание лучшей точки в ε -окрестность точного решения, $\varepsilon = \frac{\max_{i=1, \dots, n} |b_i - a_i|}{1000}$).

Результаты, полученные для разных функции, представлены в табл. 1–6. Вычисляемые характеристики округлены до 6 знака после запятой.

Таблица 1

Влияние параметров метода. Функция «Кожа»

$$f(x, y) = \left(\cos(2x^2) - 1, 1 \right)^2 + \left(\sin \frac{x}{2} - 1, 2 \right)^2 - \left(\cos(2y^2) - 1, 1 \right)^2 + \left(\sin \frac{y}{2} - 1, 2 \right)^2$$

Параметры метода				$\bar{\Delta f}$	Δf_{best}	$\bar{\sigma}_f$	$n_{усп}$
Np	T	s	τ				
30	200	5	1	0,441289	10^{-6}	0,388672	5
30	200	2	1	0,203057	0	0,237394	26
30	200	1	1	0,115555	0	0,171043	33
30	200	0,5	1	0,071228	10^{-6}	0,121099	45
30	200	0	1	0,038793	10^{-6}	0,060279	55
30	200	0,5	2	0,025898	10^{-6}	0,044631	62
30	200	0,5	12	0,005992	10^{-6}	0,015498	87
30	500	0,5	12	0,007374	10^{-6}	0,016894	84
30	1000	0,5	12	0,003231	10^{-6}	0,011757	93
50	1000	0,5	12	0,000923	10^{-6}	0,006452	98
150	1000	0,5	12	10^{-6}	10^{-6}	0	100

Таблица 2

Влияние параметров метода. Функция Экли

$$f(x, y) = -e + 20 \exp \left(-\sqrt{\frac{x^2 + y^2}{50}} \right) + \exp \left(\frac{1}{2} (\cos(2\pi x) + \cos(2\pi y)) \right)$$

Параметры метода				$\bar{\Delta f}$	Δf_{best}	$\bar{\sigma}_f$	$n_{усп}$
Np	T	s	τ				
20	200	0	1	0	0	0	100
20	200	0,001	1	0	0	0	100
20	200	0,1	1	0	0	0	100
20	200	0,5	1	0,000917	0	0,007558	99
20	200	1	1	0,197270	0	0,657170	75
20	200	2	1	1,100164	0,000158	1,253369	15
20	200	10	1	4,446471	0,163439	2,106565	0
20	200	100	1	6,256935	1,844658	2,162598	0



Таблица 3

Влияние параметров метода. Функция Розенброка

Параметры метода				$\overline{\Delta f}$	Δf_{best}	$\overline{\sigma}_f$	$n_{усп}$
Np	T	s	τ				
20	300	0	3	0,002209	0	0,008400	92
20	300	0,5	3	0,009439	0	0,027726	81
20	300	1	3	0,026629	0	0,051605	65
20	300	1,5	3	0,033928	0	0,075940	56
20	300	2	3	0,055406	0	0,097329	43
20	300	5	3	0,111751	$8 \cdot 10^{-6}$	0,174780	25
40	300	0	3	0	0	0	100
40	300	1	3	0,000500	0	0,001347	99
40	300	2	3	0,009795	0	0,021987	74

Таблица 4

Число успешных попыток алгоритма для разных целевых функций

Параметры \ Функция	$Np = 40$ $T = 300$ $s = 0,5$ $\tau = 4$	$Np = 20$ $T = 1000$ $s = 2$ $\tau = 4$	$Np = 30$ $T = 100$ $s = 0,3$ $\tau = 2$	$Np = 60$ $T = 100$ $s = 0,6$ $\tau = 5$	$Np = 100$ $T = 100$ $s = 0,2$ $\tau = 8$
Швефеля	99	86	92	100	100
Мульти-функция	100	94	95	100	100
Корневая	100	97	100	100	100
Шаффера	100	100	100	100	100
Растргина	100	71	99	100	100
Экли	100	83	100	100	100
«Кожа»	75	31	59	86	100
«Западняя»	100	82	100	100	100
Розенброка	98	53	91	100	100
Параболическая	100	99	100	100	100
Шаффера (с $\varepsilon/10$)	27	3	11	39	89

Таблица 5

Сравнение модификации и оригинальной версии. Функция «Кожа»

Параметры метода				$\overline{\Delta f}$	Δf_{best}	$\overline{\sigma}_f$	$n_{усп}$
Np	T	s	τ				
30	100	1	1	0,092848	0	0,131947	33
30	100	1	2	0,059833	10^{-6}	0,129252	50
30	100	1	4	0,034315	10^{-6}	0,060489	55
30	100	1	16	0,003236	10^{-6}	0,011756	93



Таблица 6

Сравнение модификации и оригинальной версии. Функция «Западня»

Параметры метода				$\overline{\Delta f}$	Δf_{best}	$\overline{\sigma_f}$	$n_{усп}$
Np	T	s	τ				
10	100	1	1	0,240984	0,001134	0,183174	2
10	100	1	2	0,101525	0	0,128072	22
10	100	1	4	0,029496	0	0,087174	83
10	100	1	16	0,000100	0	$5,362015 \cdot 10^{-5}$	100

Проанализировав данные, полученные в результате выполнения метода, имитирующего поведение стаи мотыльков, установлено, что алгоритм успешно справляется с задачей нахождения глобального экстремума большинства тестовых функций. Даже при задании небольших величин размера популяции, максимального числа итераций и количества возможных положений мотылька на спирали программа находит решение с достаточно высокой точностью.

Установлено, что увеличение параметра τ модификации существенно увеличивает количество успехов по сравнению с оригинальной версией (табл. 5, 6), оригинальной версии соответствует $\tau = 1$.

Наиболее эффективно алгоритм показал себя на примере функций Шаффера, Корневой и параболической функции. При статистическом анализе было получено максимальное число успешных серий из 100 запусков. Следует заметить, что успех

с функцией Шаффера прямо следует из выбора параметра $\varepsilon = \frac{\max_{i=1,...,n} |b_i - a_i|}{1000}$; если брать $\varepsilon = \frac{\max_{i=1,...,n} |b_i - a_i|}{10000}$, т.е. существенно увеличить требуемую точность решения, то эта функция будет наиболее трудной для оптимизации данным методом (табл. 4). Среди тестовых функций из табл. 4 алгоритм наиболее неудачно работает с функцией «Кожа».

8. ПОСТАНОВКА ЗАДАЧА ОПТИМИЗАЦИИ С ОГРАНИЧЕНИЯМИ ТИПА НЕРАВЕНСТВ

Дана целевая функция

$$f(x) = f(x_1, \dots, x_n),$$

определенная на множестве допустимых решений $D \subseteq R^n$ и удовлетворяющая ограничениям

$$g^1(x) \leq 0, \dots, g^m(x) \leq 0,$$

где $D = \{x | x_i \in [a_i, b_i], i = 1, 2, \dots, n\}$, число m ограничений типа неравенств задано.

Предполагается, что целевая функция и функции, задающие ограничения, непрерывны.

Требуется минимизировать значение целевой функции $f(x)$ с учетом выполнения всех заданных ограничений.

Стратегия поиска решения заключается в переходе от исходной задачи к задаче поиска глобального экстремума вспомогательной функции $F(x_1, \dots, x_n)$, полученной с помощью применения метода внешних штрафов. При этом функция $F(x_1, \dots, x_n)$ принимает вид:

$$F(x, c) = f(x) + \sum_{i=1}^m c_i \cdot [\max\{0, g^i(x)\}]^2 \rightarrow \min_{x \in D},$$

где $c_1, \dots, c_m$ – коэффициенты штрафной функции.

9. ЗАДАЧА ОПРЕДЕЛЕНИЯ ПАРАМЕТРОВ НАТЯЖНОЙ/КОМПРЕССИОННОЙ ПРУЖИНЫ

Требуется определить параметры натяжной/ компрессионной пружины, учитывая физические ограничения.

Целью является определение минимальной по весу конструкции пружины, заключающейся в определении вектора параметров $x = (x_1, x_2, x_3)^T$, соответствующих диаметру проволоки, среднему диаметру витка и числу активных витков. Конструкция пружины должна удовлетворять ограничениям по минимальному отклонению, напряжению сдвига, частоте колебаний и ограничениям на внешний диаметр.

Задача может быть формализована следующим образом:

$$\begin{aligned} f(x) &= (x_3 + 2) \cdot x_1^2 \cdot x_2, \\ g^1(x) &= 1 - \frac{x_2^3 \cdot x_3}{71875 \cdot x_1^4} \leq 0, \quad g^2(x) = \frac{4 \cdot x_2^2 - x_1 \cdot x_2}{12566 \cdot (x_1^3 \cdot x_2 - x_1^4)} + \frac{2,46}{12566 \cdot x_1^2} - 1 \leq 0, \\ g^3(x) &= 1 - \frac{140,54 \cdot x_1}{x_2^2 \cdot x_3} \leq 0, \quad g^4(x) = \frac{x_1 + x_2}{1,5} - 1 \leq 0, \\ D &= [0,05; 2,0] \times [0,25; 1,3] \times [2,0; 15,0]. \end{aligned}$$

В монографии [7] данная задача решалась с помощью применения нескольких метаэвристических алгоритмов, а также проведено сравнение с решением в первоисточнике [17]. Коэффициенты штрафной функции задавались так: $c_1 = 6$, $c_2 = c_3 = 1$, $c_4 = 0,5$.

Результаты, полученные с помощью применения методов MFO и MMFO при тех же коэффициентах штрафной функции, сравнимы с результатами из [7] и представлены в табл. 7. Параметры применяемых методов: для MFO $Np = 500$, $s = 0,2$, $T = 300$, для MMFO $Np = 500$, $s = 0,2$, $T = 300$, $\tau = 64$.



Таблица 7

Сравнение результатов решения задачи определения параметров натяжной/компрессионной пружины

	Метод, имитирующий поведение стаи мотыльков	Модификация метода, имитирующего поведение стаи мотыльков	Гибридный мультиагентный алгоритм интерполяционного поиска	Метод, имитирующий империалистическую конкуренцию	Решение, полученное в [17]
$f(x^*)$	0,0123870	0,0123868	0,0108	0,0216	0,012665

Указанным в табл. 7 значениям целевой функции соответствуют следующие результаты: $x^* = (0,051, 0,347, 11,719)^T$, $g_1(x^*) = 0,001$, $g_2(x^*) = 0,012$, $g_3(x^*) = -4,098$, $g_4(x^*) = -0,735$ и $x^* = (0,051, 0,354, 11,270)^T$, $g_1(x^*) = 0,001$, $g_2(x^*) = 0,012$, $g_3(x^*) = -4,113$, $g_4(x^*) = -0,730$ для MFO и для MMFO соответственно.

10. ОПТИМАЛЬНОЕ ПРОГРАММНОЕ УПРАВЛЕНИЕ МОБИЛЬНЫМ РОБОТОМ

Математическая модель двухгусеничного мобильного робота описывается системой дифференциальных уравнений [14, 15]

$$\begin{cases} \dot{x}_1(t) = 0,5[u_1(t) + u_2(t)]\cos x_3(t), \\ \dot{x}_2(t) = 0,5[u_1(t) + u_2(t)]\sin x_3(t), \\ \dot{x}_3(t) = 0,5[u_1(t) - u_2(t)], \end{cases}$$

где $x \in R^3$, $u \in R^2$, $t \in [0, t_f]$. Ограничения на управление: $|u_i| \leq 10, i = 1, 2$. Робот имеет два управления в виде скоростей вращения левой и правой гусениц, поворот осуществляется за счет разности скоростей вращения гусениц. Движение вперед и назад определяется полусуммой управлений, а поворот – полуразностью. Координаты x_1, x_2 задают положение робота на плоскости движения, а x_3 – угол направления.

Начальные условия:

$$x_1(0) = 10, x_2(0) = 10, x_3(0) = 0.$$

Конечные условия:

$$x_{1f} = 0, x_{2f} = 0, x_{3f} = 0.$$

Имеются два фазовых ограничения на плоскости движения робота

$$(x_1 - 2,5)^2 + (x_2 - 2,5)^2 \geq 2,5^2, \quad (x_1 - 7,5)^2 + (x_2 - 7,5)^2 \geq 2,5^2,$$

т.е. две запретные зоны, ограниченные окружностями.

Требуется перевести робота из заданной начальной точки в заданную конечную точку, не нарушая ограничений, за минимальное время.

Для формирования критерия качества в [14] предлагается записать ограничения в форме

$$h_1(x) = 2,5 - \sqrt{(x_1 - 2,5)^2 + (x_2 - 2,5)^2} \leq 0, \quad h_2(x) = 2,5 - \sqrt{(x_1 - 7,5)^2 + (x_2 - 7,5)^2} \leq 0.$$

Тогда функционал представляется в виде

$$I = t_f + \sqrt{\sum_{i=1}^3 (x_i(t_f) - x_{if})^2} + \int_0^{t_f} \sum_{i=1}^2 h_i(x(t)) \cdot 1(h_i(x(t))) dt \rightarrow \min,$$

$$\text{где } t_f = \begin{cases} t, & \text{если } \sqrt{\sum_{i=1}^3 (x_i(t) - x_{if})^2} \leq 0,01, \\ 2,5, & \text{иначе,} \end{cases} \quad 1(h_i(x(t))) = \begin{cases} 1, & \text{если } h_i(x(t)) > 0, \\ 0, & \text{если } h_i(x(t)) \leq 0. \end{cases}$$

В [14, 15] для решения поставленной задачи применялась кусочно-линейная аппроксимация закона управления. Здесь для поиска приближенного решения применяется кусочно-постоянное управление. Оно принадлежит классу законов управления с переключениями, что следует из соотношений принципа максимума для данной постановки задачи.

Для реализации более гибкого процесса поиска решения предлагается дополнить функционал весовыми коэффициентами, учитывающими важность каждого из трех слагаемых:

$$I = \lambda_1 t_f + \lambda_2 \sqrt{\sum_{i=1}^3 (x_i(t_f) - x_{if})^2} + \lambda_3 \int_0^{t_f} \sum_{i=1}^2 h_i(x(t)) \cdot 1(h_i(x(t))) dt \rightarrow \min,$$

где $\lambda_i \geq 0$, $i = 1, 2, 3$ – весовые коэффициенты.

Для уточнения результатов решения поставленной задачи, полученных с помощью применения описанного выше модифицированного метода, имитирующего поведение стаи мотыльков, применялись алгоритмы Luus–Jaakola [18] и перекоммутации [19]. В первом случае полученное на первом этапе решение выбиралось в качестве начального приближения в методе Luus–Jaakola. Во втором случае в результате многократного запуска связки методов из первого случая из найденных решений формируется множество *Pool*. Далее для уточнения полученных решений используется приведенный далее алгоритм перекоммутации (используемые обозначения соответствуют рассматриваемой общей задаче оптимизации).

11. АЛГОРИТМ ПЕРЕКОММУТАЦИИ

Шаг 0. Задать параметры $PR_{\max}, \Delta_{pr}$ – максимальное число итераций и количество точек разбиения.



Шаг 1. Положить $pr = 1$.

Шаг 2. Выбрать три различных случайных решения $x_{pool}^p, x_{pool}^q, x_{pool}^r$ из множества $Pool$.

Шаг 3. Найти решение

$$x_{pool}^{pq} = \arg \min_{j=1, \dots, \Delta_{pr}-1} f(x_{pool}^p + j(x_{pool}^q - x_{pool}^p) / \Delta_{pr}).$$

Шаг 4. Добавить решение

$$x^{new} = \arg \min_{j=1, \dots, \Delta_{pr}-1} f(x_{pool}^{pq} + j(x_{pool}^r - x_{pool}^{pq}) / \Delta_{pr})$$

в множество $Pool$. Положить $pr := pr + 1$.

Шаг 5. Если $pr > PR_{\max}$, то перейти к шагу 6, иначе – к шагу 2.

Шаг 6. Среди решений, принадлежащих множеству $Pool$, выбрать наилучшее.

12. ПРИМЕНЕНИЕ МОДИФИЦИРОВАННОГО МЕТОДА, ИМИТИРУЮЩЕГО ПОВЕДЕНИЕ СТАИ МОТЫЛЬКОВ, ДЛЯ РЕШЕНИЯ ЗАДАЧИ ОПТИМАЛЬНОГО ПРОГРАММНОГО УПРАВЛЕНИЯ МОБИЛЬНЫМ РОБОТОМ

Для решения поставленной задачи разработана программа на основе алгоритма метода, имитирующего поведение стаи мотыльков. Для ее создания использовалась среда разработки Microsoft Visual Studio, язык программирования C#. Характеристики компьютера, на котором проводились вычисления: процессор Intel Core i7–10750H CPU@ 2.60GHz, 6 ядер, RAM:16 Gb.

Поиск управления осуществлялся в классе кусочно-постоянных функций. На основе выполненных экспериментов выбор 25 участков постоянности управлений u_1 и u_2 привёл к наилучшему результату. Для интегрирования системы обыкновенных дифференциальных уравнений, описывающей движение робота, применялся численный метод Адамса второго порядка, а для достижения желаемой точности каждый участок управления разбивался на 10 равных частей.

Рассмотрим два варианта задания функционала качества управления.

Вариант 1. $\lambda_1 = \lambda_2 = \lambda_3 = 1$ (слагаемые в функционале имеют одинаковую важность).

Приведены результаты трех способов выбора параметров в реализации последовательного применения модифицированного метода MFO (MMFO), метода случайного поиска с последовательной редукцией области исследования (метода Luus–Jaakola) и метода перекоммутаций (path-relinking).

1. Применялся модифицированный метод MFO с параметрами: $Np = 100$, $Np = 100$, $s = 0.5$, $T = 100$, $\tau = 16$. После многократного применения алгоритма получены результаты, среди которых выбран наилучший $I = 2,50599809962037$ и представлен на рис. 8.

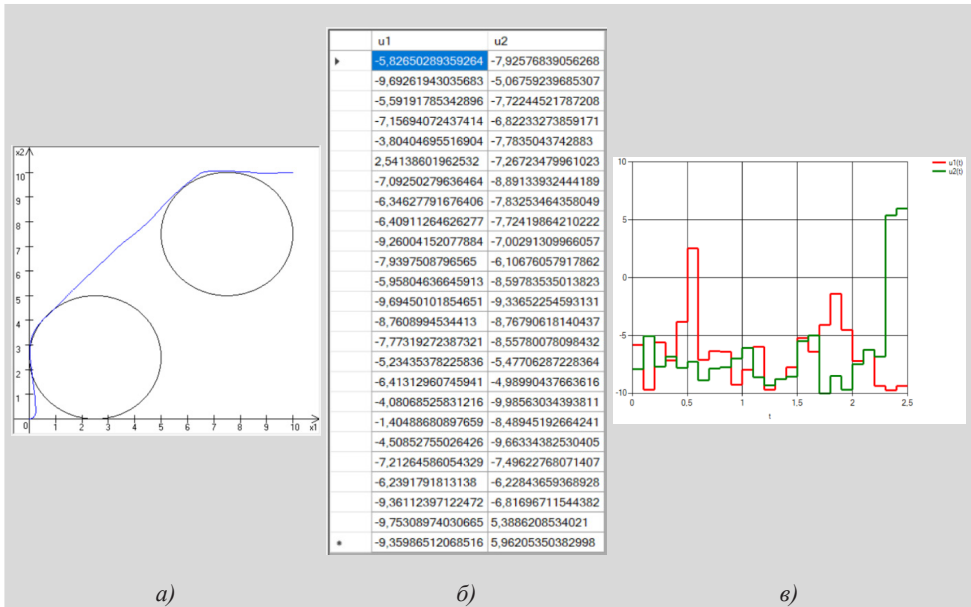


Рис. 8. Вариант 1, № 1; а) траектория движения,
б) программное управление (таблица), в) программное управление (график)

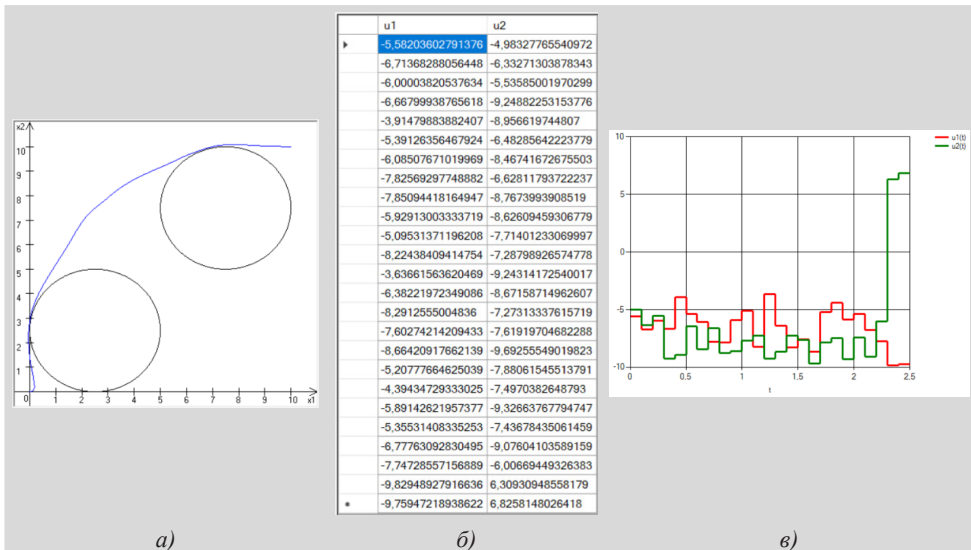


Рис. 9. Вариант 1, № 2; а) траектория движения,
б) программное управление (таблица), в) программное управление (график)



2. Применялся модифицированный метод MFO с параметрами: $Np=100$, $s=0.5$, $T=100$, $\tau=16$, далее метод Luus–Jaakola с параметрами: $R=100$, $\gamma=0.9$, $\eta=0.8$, $P=100$, $ITER=100$, $\varepsilon_1=10^{-10}$, $\varepsilon_2=10^{-10}$, а в качестве начальной точки выбирался результат применения модифицированного MFO. После многократного применения данной связки алгоритмов получены результаты, среди которых выбран наилучший $I = 2,50000000002668$ и представлен на рис. 9.
3. Применялся модифицированный метод MFO с параметрами: $Np=100$, $s=0.5$, $T=100$, $\tau=16$, далее метод Luus–Jaakola с параметрами: $R=300$, $\gamma=0.9$, $\eta=0.8$, $P=300$, $ITER=300$, $\varepsilon_1=10^{-5}$, $\varepsilon_2=10^{-5}$, а в качестве начальной точки выбирался результат применения модифицированного MFO. Из 6 полученных результатов формировалось множество $Pool$, после чего применялся алгоритм перекоммутации на этом множестве с параметрами: $PR_{\max}=5000$, $\Delta_{pr}=50$. После многократного применения данной связки алгоритмов получены результаты, среди которых выбран наилучший $I = 2,45943580840689$ и представлен на рис. 10. Для двух из трех случаев следует отметить небольшое нарушение ограничений в процессе движения, что можно отнести к недостаткам найденных решений.

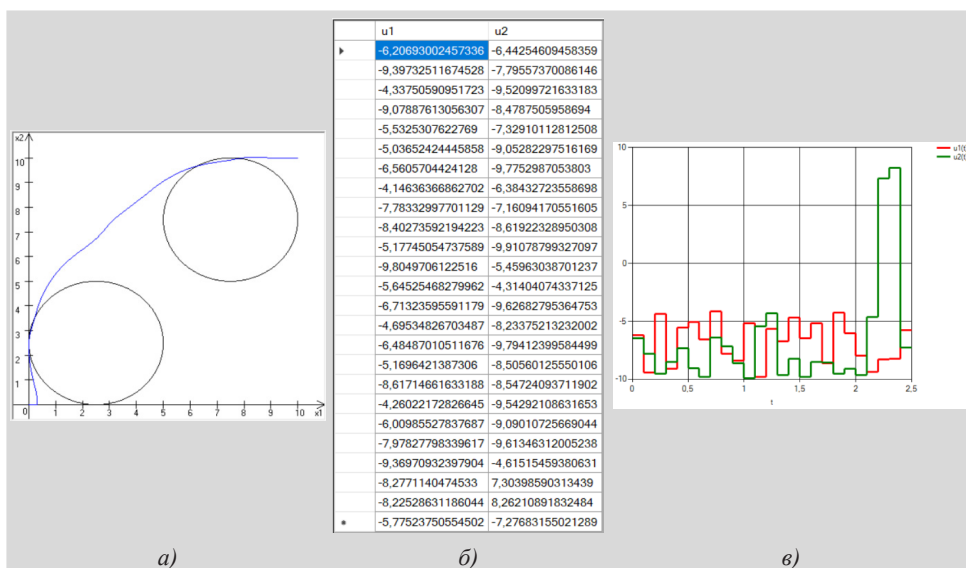


Рис. 10. Вариант 1, № 3; а) траектория движения, б) программное управление (таблица), в) программное управление (график)

Полученные численные результаты: $I = 2,50599809962037$, $I = 2,50000000002668$, $I = 2,45943580840689$ – сравнимы с результатами, полученными в [14, 15]. В своей статье авторы с помощью метаэвристических алгоритмов получили следующие значения функционала, наименьшие среди ранее полученных: пчелиный алгоритм (BA): 2,2404; метод роя частиц (PSO): 2,3777; алгоритм серых волков (GW): 2,4242;

алгоритм летучих мышей (BIA): 2,5138; генетический алгоритм (GA): 2,5054; алгоритм дифференциальной эволюции (DE): 2,9287. Только применение пчелиного алгоритма привело к оптимальной траектории, достигающей конечную точку без нарушения фазовых ограничений. Заметим, что применение связки алгоритмов MMFO и Luus–Jaakola также привело к оптимальной траектории без нарушения фазовых ограничений.

Вариант 2. $\lambda_1 = 1$, $\lambda_2 = 1$, $\lambda_3 = 6$ (увеличен штраф за нарушение ограничений). Приведем результаты для трех случаев.

1. Применялся модифицированный метод MFO с параметрами: $Np = 100$, $s = 0.5$, $T = 100$, $\tau = 16$. После многократного применения алгоритма получены результаты, среди которых выбран наилучший $I = 2,50002519885572$ и представлен на рис. 11.
2. Применялся модифицированный метод MFO с параметрами: $Np = 100$, $s = 0.5$, $T = 100$, $\tau = 16$, далее метод Luus–Jaakola с параметрами: $R = 100$, $\gamma = 0.9$, $\eta = 0.8$, $P = 100$, $ITER = 100$, $\varepsilon_1 = 10^{-5}$, $\varepsilon_2 = 10^{-5}$, а в качестве начальной точки выбирался результат применения модифицированного MFO. После многократного применения данной связки алгоритмов получены результаты, среди которых выбран наилучший $I = 2,50000000003196$ и представлен на рис. 12.
3. Применялся модифицированный метод MFO с параметрами: $Np = 100$, $s = 0.5$, $T = 100$, $\tau = 16$, далее метод случайного поиска с последовательной редукцией области исследования с параметрами: $R = 300$, $\gamma = 0.9$, $\eta = 0.8$, $P = 300$, $ITER = 300$, $\varepsilon_1 = 10^{-5}$, $\varepsilon_2 = 10^{-5}$, а в качестве начальной точки выбирался результат применения модифицированного MFO. Из 6 результатов формировалось множество *Pool*. После чего применялся алгоритм перекоммутации на этом множестве с параметрами: $PR_{\max} = 5000$, $\Delta_{pr} = 50$. После многократного применения данной связки алгоритмов получены результаты, среди которых выбран наилучший $I = 2,5$ и представлен на рис. 13.

Анализ полученных результатов свидетельствует о том, что в результате подбора параметров методов удастся добиться выполнения ограничений, попадания в заданную конечную точку с одновременной минимизацией времени, затрачиваемого на маневр.

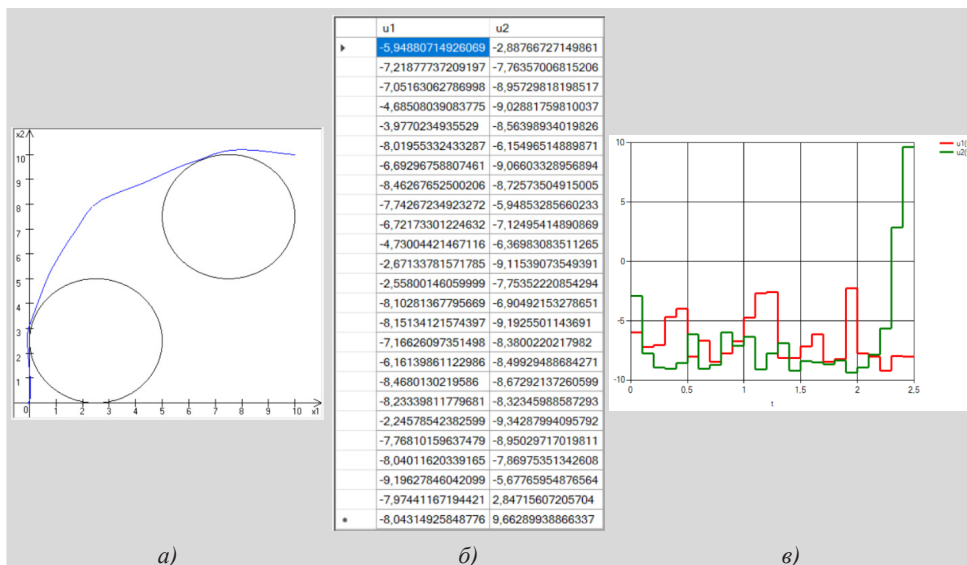


Рис. 11. Вариант 2, № 1; а) траектория движения, б) программное управление (таблица), в) программное управление (график)

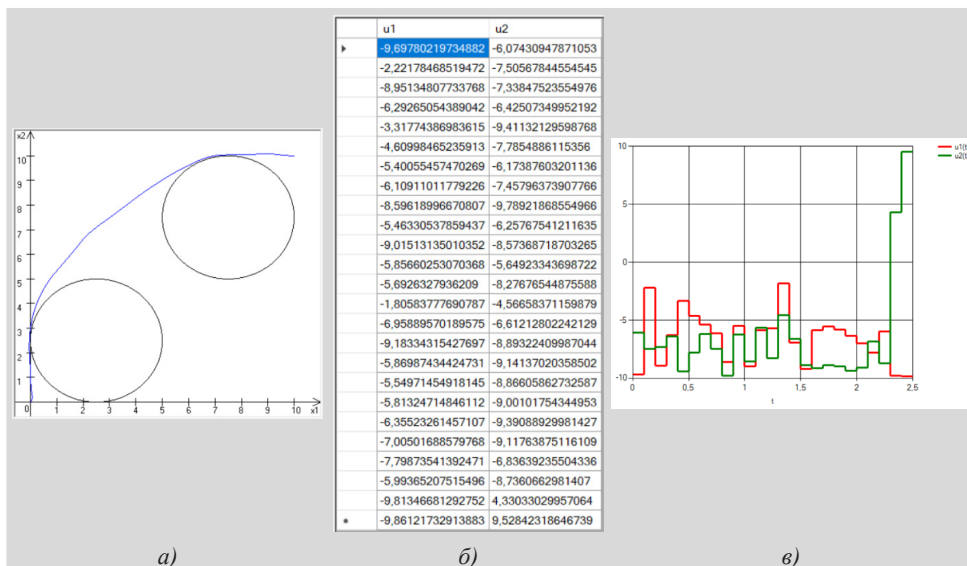


Рис. 12. Вариант 2, № 2; а) траектория движения, б) программное управление (таблица), в) программное управление (график)

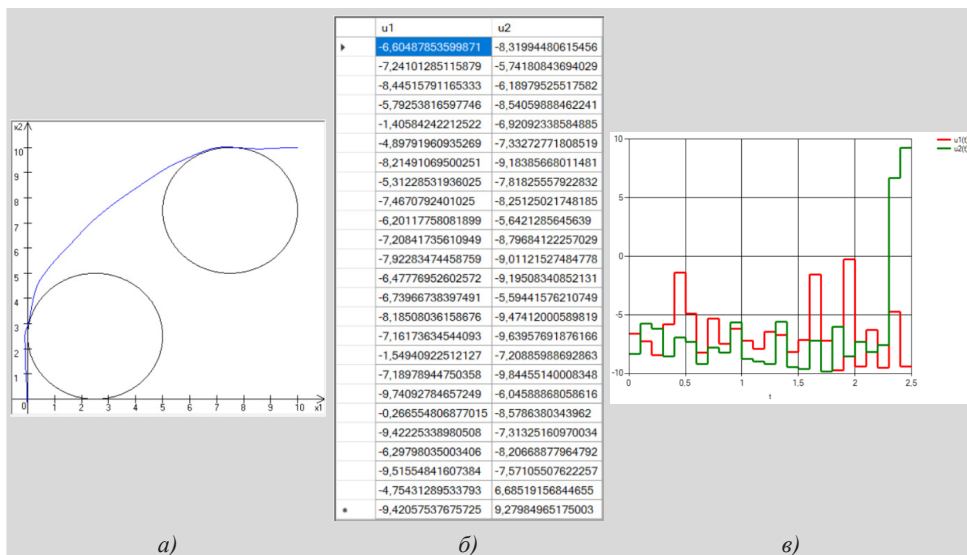


Рис. 13. Вариант 2, № 3; а) траектория движения,
б) программное управление (таблица), в) программное управление (график)

13. ЗАКЛЮЧЕНИЕ

Предложена модификация метода конечномерной оптимизации, имитирующего поведение стаи мотыльков. На основе описанного алгоритма разработана программа для исследования его эффективности. При применении разработанного метода для поиска экстремума тестовых функций из общеизвестного набора, для поиска оптимального программного управления с известным точным решением, а также для определения параметров натяжной/компрессионной пружины с ограничениями типа неравенств получены результаты, позволяющие утверждать об усовершенствовании способности метода находить глобальные экстремумы по сравнению с оригинальной версией.

Для поставленной задачи оптимального программного управления мобильным роботом при наличии препятствий применялся предложенный модифицированный метод в совокупности с методом случайного поиска с последовательной редукцией области исследования и алгоритмом перекоммутации. Для избегания попадания робота в запретные зоны в функционал качества управления добавлены весовые коэффициенты, учитывающие важность вклада отдельных слагаемых, отражающих точность попадания робота в заданную точку, штраф за нарушение ограничений и величину промежутка времени, отведенного для достижения цели. Полученные результаты свидетельствуют о решении поставленной задачи, поскольку мобильный робот добирается из заданной начальной точки в фиксированную конечную точку, огибая запретные зоны с одновременной минимизацией затрачиваемого времени.



Литература

1. Пантелеев А.В. Метаэвристические алгоритмы оптимизации законов управления динамическими системами. М.: Факториал. 2020. 564 с.
2. Handbook of Metaheuristics / Eds M. Gendreau, J.-Y. Potvin. N.Y.: Springer. 2019. 610 p.
3. Карпенко А.П. Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой. М.: Изд-во МГТУ им. Н.Э. Баумана. 2021. 448 с.
4. Niculina Dragoi E., Dafinescu V. Review of metaheuristics inspired from the animal kingdom // Mathematics. 2021. Vol. 9. 2335.
5. Tzanetos A., Fister I., Dounias G. A comprehensive database of nature-inspired algorithms // Data in Brief. 2020. Vol. 31. 105792.
6. Bio-inspired computation: Where we stand and what's next / Del Ser J., Osaba E., Molina D., Yang X.-S., Salcedo-Sanz S., Camacho D., Das S., Suganthan P.N., Coello Coello C.A., Herrera F. // Swarm and Evolutionary Computation. 2019. Vol. 48. P. 220–250.
7. Пантелеев А.В., Каранэ М.М.С. Мультиагентные и биоинспирированные методы оптимизации технических систем. М.: Изд-во Доброе слово и Ко. 2024. 336 с.
8. Sergeyev Y.D., Kvasov D.E. Deterministic Global Optimization: An Introduction to the Diagonal Approach. N.Y.: Springer. 2017. 136 p.
9. Sergeyev Y.D., Kvasov D.E., Mukhametzhonov M.S. On the efficiency of nature-inspired metaheuristics in expensive global optimization with limited budget // Scientific Reports. 2018. Vol. 8. P. 453.
10. Mirjalili S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm // Knowledge-Based Systems. 2015. Vol. 89. P. 228–249.
11. A covariance-based Moth–flame optimization algorithm with Cauchy mutation for solving numerical optimization problems / Xiaodong Zhao, Yiming Fang, Le Liu, Miao Xu, Qiang Li. // Applied Soft Computing. 2022. Vol. 119.
12. Wolpert D.H., Macready W.G. No free lunch theorems for optimization // IEEE Transactions on Evolutionary Computation. 1997. Vol. 1. P. 67–82.
13. X.-S. Yang. Test problems in optimization // Engineering Optimization: An Introduction with Metaheuristic Applications (Eds Xin-She Yang). John Wiley & Sons. 2010.
14. Diveev A.I., Konstantinov S.V. Study the practical convergence of evolutionary algorithms for the optimal program control of a wheeled robot // Journal of Computer and Systems Sciences International. 2018. Vol. 57. No. 4. P. 561–580.
15. Konstantinov S.V., Diveev A.I., Balandina G.I., Baryshnikova A.A. Comparative Research of Random Search Algorithms and Evolutionary Algorithms for the Optimal Control Problem of the Mobile Robot. 13th Intern. Symp. “Intelligent Systems” (INTELS’18). Procedia Computer Science, 150. 2019. P. 462–470.
16. Hoare C.A.R. Algorithm 64: Quicksort // Communications of the ACM. 1961. Vol. 4. P. 321.
17. Golinski J. An adaptive optimization system applied to machine synthesis // Mech. Mash. Theory. 1973. Vol. 8. No. 3. P. 419–436.
18. Luus R. Iterative Dynamic Programming. London: Chapman & Hall/CRC. 2000. 331 p.
19. Glover. F., Marti R., Laguna M. Fundamentals of scatter search and path-relinking // Control & Cybernetics. 2000. Vol. 39. P. 653–684.



Application of the Modified Method Simulating the Behavior of a Flock of Moths to Solve the Optimal Open Loop Control Problem of a Mobile Robot Movement

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

Ivan S. Nadorov**

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

ORCID: <https://orcid.org/0009-0008-2085-2987>

e-mail: nnadorovivan@gmail.com

The article proposes a modification of the metaheuristic optimization method simulating the behavior of a swarm of moths, which belongs to the group of bio-inspired methods. A step-by-step algorithm is formed and the efficiency of the modified method is studied on a generally accepted set of test functions of many variables with a complex structure of level surfaces, on the problem of optimal open loop control with a known exact solution, as well as on the problem of determining the parameters of the tension/compression spring with constraints such as inequalities. The advantage of the modified method over the original version is shown. It is demonstrated that the method allows finding solutions of sufficiently good quality in a time acceptable from a practical point of view. A solution to an applied problem of finding the optimal open loop control of a mobile robot on a plane in the presence of obstacles is given. The goal of the control is to achieve a given end point while minimizing the elapsed time and fulfilling the condition of enveloping forbidden areas. To find the control law as a function of time, a piecewise constant approximation was used, which allows reducing the problem to finding a finite number of unknown parameters. The solutions of the terminal problem of speed of response with different structure and parameters of the composite quality functional are considered using the sequential application of the developed modified method simulating the behavior of a swarm of moths, the random search method with sequential reduction of the study area and the path-relinking method. The results of comparison with known solutions are presented.

***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

****Ivan S. Nadorov**, Student, «Institute of Computer Science and Applied Mathematics», Moscow Aviation Institute (National Research University) (MAI), Moscow, Russia, ORCID: <https://orcid.org/0009-0008-2085-2987>, e-mail: nnadorovivan@gmail.com



Keywords: global optimization, metaheuristic optimization algorithms, optimal open loop control

For citation:

Panteleev A.V., Nadorov I.S. Application of the Modified Method Simulating the Behavior of a Flock of Moths to Solve the Optimal Open Loop Control Problem of a Mobile Robot Movement. *Modelirovanie i analiz dannykh = Modelling and Data Analysis*, 2025. Vol. 15, no. 1, pp. 81–109. DOI: <https://doi.org/10.17759/mda.2025150105> (In Russ., abstr. in Engl.).

References

1. Panteleev A.V. Metaheuristic Algorithms for Optimizing the Laws of Control of Dynamic Systems. Moscow, Faktorial Publ., 2020. 564 p.
2. Handbook of Metaheuristics / Eds M. Gendreau, J-Y. Potvin. N.Y.: Springer, 2019. 610 p.
3. Karpenko A.P. Modern algorithms for search engine optimization. Algorithms inspired by nature. Moscow, Izd-vo MG TU im. N.E. Bauman, 2021. 448 p.
4. Niculina Dragoi E., Dafinescu V. Review of metaheuristics inspired from the animal kingdom. *Mathematics*, 2021. Vol. 9, 2335.
5. Tzanetos A., Fister I., Dounias G. A comprehensive database of nature-inspired algorithms. *Data in Brief*, 2020. Vol. 31, 105792.
6. Bio-inspired computation: Where we stand and what's next / Del Ser J., Osaba E., Molina D., Yang X.-S., Salcedo-Sanz S., Camacho D., Das S., Suganthan P.N., Coello Coello C.A., Herrera F. *Swarm and Evolutionary Computation*, 2019. Vol. 48, pp. 220–250.
7. Panteleev A.V., Karane M.M.S. Multi-agent and bio-inspired methods for technical systems optimization. M.: Izd-vo Dobroe slovo i Ko, 2024. 336 p.
8. Sergeyev Y.D., Kvasov D.E., Mukhametzhanov M.S. On the efficiency of nature-inspired metaheuristics in expensive global optimization with limited budget. *Scientific Reports*, 2018. Vol. 8, P. 453.
9. Sergeyev Y.D., Kvasov D.E. Deterministic Global Optimization: An Introduction to the Diagonal Approach. Springer, 2017. 136 p.
10. Mirjalili S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-Based Systems*, 2015. Vol. 89, pp. 228–249.
11. Xiaodong Zhao, Yiming Fang, Le Liu, Miao Xu, Qiang Li. A covariance-based Moth–flame optimization algorithm with Cauchy mutation for solving numerical optimization problems. *Applied Soft Computing*, 2022. Vol. 119.
12. Wolpert D.H., Macready W.G. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1997. Vol. 1, pp. 67–82.
13. X.-S. Yang, Test problems in optimization. Engineering Optimization: An Introduction with Metaheuristic Applications (Eds Xin-She Yang), John Wiley & Sons, 2010.
14. Diveev A.I., Konstantinov S.V. Study the practical convergence of evolutionary algorithms for the optimal program control of a wheeled robot. *Journal of Computer and Systems Sciences International*. 2018. Vol. 57, no. 4, pp. 561–580.
15. Konstantinov S.V., Diveev A.I., Balandina G.I., Baryshnikova A.A. Comparative Research of Random Search Algorithms and Evolutionary Algorithms for the Optimal Control Problem of the Mobile Robot. 13th Intern. Symp. “Intelligent Systems” (INTELS’18). *Procedia Computer Science*, 150, 2019. pp. 462–470.
16. Hoare C.A.R. Algorithm 64: Quicksort. *Communications of the ACM*, 1961. Vol. 4, P. 321.
17. Golinski J. An adaptive optimization system applied to machine synthesis. *Mech. Mash.Theory*, 1973. Vol. 8, no. 3, pp. 419–436.



18. Luus R. Iterative Dynamic Programming. London, *Chapman & Hall/CRC*, 2000. 331 p.
19. Glover. F., Marti R., Laguna M. Fundamentals of scatter search and path-relinking. *Control & Cybernetics*, 2000. Vol. 39, pp. 653–684.

Получена 19.12.2024

Принята в печать 17.01.2025

Received 19.12.2024

Accepted 17.01.2025