

Научная статья  
УДК 517.977:519.7  
DOI: [10.34759/trd-2022-122-24](https://doi.org/10.34759/trd-2022-122-24)

## ПРИМЕНЕНИЕ БИОИНСПИРИРОВАННЫХ МЕТОДОВ ОПТИМИЗАЦИИ В ЗАДАЧЕ ОПТИМАЛЬНОГО ПРОГРАММНОГО УПРАВЛЕНИЯ СОЛНЕЧНЫМ ПАРУСОМ

Андрей Владимирович Пантелеев<sup>1✉</sup>, Иван Андреевич Беляков<sup>2</sup>

<sup>1,2</sup>Московский авиационный институт (национальный исследовательский университет), МАИ, Москва, Россия

<sup>1</sup>[avpanteleev@inbox.ru](mailto:avpanteleev@inbox.ru) ✉

<sup>2</sup>[vanbelyakov@yandex.ru](mailto:vanbelyakov@yandex.ru)

**Аннотация.** Изложена методика решения задачи быстрогодействия для космического аппарата, реализующего перелет между двумя планетами, при помощи параметризации закона управления и использования двух современных биоинспирированных алгоритмов условной глобальной оптимизации – методов, имитирующих поведение стай серых волков, стай горбатых китов.

Данные алгоритмы относятся к классу методов роевого интеллекта, особенностью которого является возможность обмена информацией между особями. Предложена методика нахождения оптимального программного управления нелинейными непрерывными детерминированными динамическими системами в условиях, когда правый конец промежутка времени функционирования системы не фиксирован.

Программное управление ищется в классе разложений по различным системам финитных базисных функций с неопределенными коэффициентами.

Предложенная методика применяется для решения задачи оптимального управления солнечным парусом, используемым для перелета с орбиты Земли на орбиту Меркурия за минимальное время при одновременном обеспечении выполнения требований к параметрам конечной орбиты. Сформировано алгоритмическое и программное обеспечение двух биоинспирированных алгоритмов. Приводятся рекомендации по выбору параметров алгоритмов для решения типовых модельных задач оптимизации и прикладной задачи управления солнечным парусом.

**Ключевые слова:** космический аппарат, солнечный парус, параметрическая оптимизация управления, биоинспирированные стратегии, финитные базисные функции

**Для цитирования:** Пантелеев А.В., Беляков И.А. Применение биоинспирированных методов оптимизации в задаче оптимального программного управления солнечным парусом // Труды МАИ. 2022. № 122. DOI: [10.34759/trd-2022-122-24](https://doi.org/10.34759/trd-2022-122-24)

## **APPLICATION OF THE BIO-INSPIRED OPTIMIZATION METHODS IN THE SOLAR SAIL OPTIMAL OPEN-LOOP CONTROL PROBLEM**

**Andrey V. Panteleev<sup>1</sup>✉, Ivan A. Belyakov<sup>2</sup>**

<sup>1,2</sup>Moscow Aviation Institute (National Research University), Moscow, Russia

<sup>1</sup>[avpanteleev@inbox.ru](mailto:avpanteleev@inbox.ru) ✉

<sup>2</sup>[vanbelyakov@yandex.ru](mailto:vanbelyakov@yandex.ru)

**Abstract.** This article discusses application of bio-inspired metaheuristic global optimization methods: Grey Wolf Optimizer (GWO) and Whale Optimization Algorithm (WOA) for the problem of optimal loop-open control. Optimal control needs to manipulate space probe with solar sail. These algorithms belong to the class of swarm optimization methods, which feature is possibility of data transfer between individuals. At the same time GWO and WOA were inspired by nature, therefore we may relate them to bio-inspired category. The behavior of a nonlinear deterministic continuous model of a plant is described by a system of differential equations with given initial conditions and a non-fixed terminal time. It is assumed that only time information is available during control, i.e. the open-loop control is used. Numerical solution is found in the form of saturation function, which guarantees fulfillment of parallelepiped control constraints. Function arguments are finding in the linear combination of given base functions. Since terminal moment is undefined, this article applies method of transformation to the fixed terminal time problem, which introduces new independent variable, related with time.

The task is to find optimal control law for realizing solar system mission flight from the Earth's orbit to the Mercury's orbit in the shortest possible time. Algorithm of two bio-inspired metaheuristic optimization methods application and software were created for numerical solution solar sail open-loop control problem. Software gives mathematical modelling with various parameters for a subsequent visualization. Comparison with known results was provided. Recommendations were provided for parameters choose to solve typical model optimization and applied solar sail control problem.

**Keywords:** optimal open-loop control problems, global optimization algorithm, software, bio-inspired algorithm, metaheuristic, time optimal control, solar sail

**For citation:** Panteleev A.V., Belyakov I.A. Application of the bio-inspired optimization methods in the solar sail optimal open-loop control problem. *Trudy MAI*, 2022, no. 122.

DOI: [10.34759/trd-2022-122-24](https://doi.org/10.34759/trd-2022-122-24)

## Введение

В теории оптимального управления значительное место занимают задачи поиска программного управления как с фиксированным, так и нефиксированным промежутком времени [1-6]. Для поиска решения обычно применяется необходимое условие оптимальности - принцип максимума, применение которого в общем случае сводится к численному решению двухточечных краевых задач различной сложности. Применение достаточных условий оптимальности связано с решением уравнения Беллмана с целью нахождения оптимального управления с полной обратной связью, что, как правило, связано с еще большими вычислительными трудностями. Поэтому сведение задачи оптимального управления к конечномерной параметрической задаче оптимизации остается одним из актуальных путей решения прикладных задач, особенно в области авиационной и ракетно-космической техники.

В статье рассматривается решение задачи наискорейшего перехода космического аппарата (солнечного паруса) с орбиты Земли на заданную орбиту Меркурия. Законы оптимального программного управления ищутся на основе принятых в спектральной теории разложений по различным системам базисных функций с неизвестными коэффициентами, подлежащими определению. В данном случае используются четыре возможных системы финитных базисных функций,

заданных на конечных отрезках времени: кусочно-постоянные, кусочно-линейные, квадратичные и кубические сплайны. Для нахождения неопределенных коэффициентов разложения предлагается использовать два родственных между собой биоинспирированных алгоритма оптимизации функций многих переменных при параллелепипедных ограничениях – алгоритмы, имитирующие поведение стай серых волков и горбатых китов. Эти алгоритмы доказали свою эффективность при решении общепринятого стандартного набора текстовых задач оптимизации (benchmark functions). В результате сформировано алгоритмическое и программное обеспечения решения задачи управления космическим аппаратом с помощью солнечного паруса. Проведено исследование влияния параметров используемых биоинспирированных алгоритмов условной оптимизации и выбора типа базисной системы на время, требуемое для реализации перелета между заданными орбитами, и точность. Выработаны соответствующие рекомендации для рассматриваемого класса задач оптимального управления.

### **Постановка задачи поиска оптимального программного управления**

Пусть поведение модели объекта управления описывается обыкновенным дифференциальным уравнением

$$\dot{x}(t) = f(x(t), u(t)) , \quad (1)$$

где  $x$  – вектор состояния системы,  $x = (x_1, \dots, x_n)^T \in R^n$ ;  $u$  – вектор управления,  $u = (u_1, \dots, u_q)^T \in U \subseteq R^q$ ,  $U$  – некоторое заданное множество допустимых значений управления, определяемое прямым произведением отрезков  $[a_1, b_1] \times \dots \times [a_q, b_q]$ ;  $t \in T = [0, t_f]$  – промежуток времени функционирования системы; момент окончания

$t_f$  заранее не задан;  $f(x, u)$  – непрерывная вектор-функция;  $R^n$  –  $n$ -мерное евклидово пространство.

Начальное состояние системы  $x_0$  определяет начальное условие  $x(0) = x_0$ .

Определим множество допустимых процессов  $D(x_0)$  как множество троек  $d = (t_f, x(\cdot), u(\cdot))$ , которые включают момент окончания процесса управления, траекторию  $x(\cdot)$  и управление  $u(\cdot)$  (где  $\forall t \in T: x(t) \in R^n, u(t) \in U$ , функции  $x(\cdot)$  непрерывны и кусочно-дифференцируемы, а  $u(\cdot)$  кусочно-непрерывны), удовлетворяющие уравнению (1) с заданным начальным условием.

Определим критерий качества управления на множестве допустимых процессов  $D(x_0)$ :

$$I(d) = F(t_f, x(t_f)), \quad (2)$$

где  $F(t_f, x)$  – непрерывная функция, задающая так называемый терминальный член функционала.

Требуется найти такой элемент множества допустимых процессов  $d^* = (t_f^*, x^*(\cdot), u^*(\cdot)) \in D(x_0)$ , на котором достигается минимум критерия качества:

$$I(d^*) = \min_{d \in D(x_0)} I(d).$$

### **Алгоритм поиска оптимального программного управления**

Искомый закон оптимального управления предлагается искать в форме конструкции, определяемой функцией насыщения (sat), которая автоматически гарантирует выполнение типичных в прикладных задачах ограничений на управление параллелепипедного типа. В качестве аргументов функции насыщения

предлагается использовать разложения по известным системам базисных функций с неопределенными коэффициентами, применяемые в спектральной форме математического описания систем управления [7-10]. Поскольку момент окончания явно не задан, то возникает проблема определения семейства базисных функций. Поэтому применяется предложенный в [11] подход, связанный с введением новой независимой переменной  $\tau \in [0,1]$ , отрезок изменения которой делится на  $P$  подынтервалов одинаковой длины, равной  $\frac{1}{P}$ . Отрезок времени функционирования системы  $[0, t_f]$  разбивается на  $P$  подынтервалов переменной длины  $h_{k+1} = t_{k+1} - t_k \geq 0$ ,  $k = 0, 1, \dots, P-1$ ;  $t_0 = 0$ . При этом момент окончания процесса находится по формуле

$$t_f = \sum_{k=0}^{P-1} h_{k+1}. \quad (3)$$

Моменту  $\tau = 0$  соответствует  $t = 0$ , а моменту  $\tau = 1$  момент окончания  $t_f$ .

Используем соотношение, связывающее дифференциалы переменных  $t$  и  $\tau$ :

$$dt = h_{k+1} P d\tau, \quad t_k \leq t < t_{k+1}, \quad k = 0, 1, \dots, P-1. \quad (4)$$

Интегрируя по отдельному промежутку времени  $t \in [t_k, t_{k+1}]$ , получаем

$$\int_{t_k}^{t_{k+1}} dt = h_{k+1} P \int_{\tau_k}^{\tau_{k+1}} d\tau, \quad \underbrace{t_{k+1} - t_k}_{h_{k+1}} = h_{k+1} P (\tau_{k+1} - \tau_k). \quad \text{Отсюда следует, что шаг по переменной } \tau$$

постоянный и равен  $\frac{1}{P}$ , так как  $\tau_{k+1} - \tau_k = \frac{1}{P}$ ,  $k = 0, 1, \dots, P-1$ ;  $\tau_0 = 0, \tau_P = 1$ .

Найдем связь между переменными  $t$  и  $\tau$ :  $\int_{t_k}^t dt = h_{k+1} P \int_{\tau_k}^{\tau} d\tau$ ,

$t - t_k = h_{k+1} P (\tau - \tau_k) = h_{k+1} P (\tau - \frac{k}{P})$ ,  $k = 0, 1, \dots, P-1$ . Поскольку  $t_k = \sum_{j=0}^{k-1} h_{j+1}$ , имеем

$$t = \sum_{j=0}^{k-1} h_{j+1} + h_{k+1} P \left( \tau - \frac{k}{P} \right), \quad k = 0, 1, \dots, P-1. \quad (5)$$

С учетом (4) уравнение (1) можно переписать в форме

$$\frac{dx(\tau)}{d\tau} = h_{k+1} P f(x(\tau), u(\tau)), \quad \tau_k \leq \tau < \tau_{k+1}; \quad k = 0, 1, \dots, P-1; \quad x(0) = x_0. \quad (6)$$

Искомое управление предлагается искать в виде функции насыщения  $\text{sat}$ , гарантирующей выполнение ограничений на управление:

$$u_j(\tau) = \text{sat} \{ g_j(\tau) \}, \quad j \in \overline{1, q}; \quad (7)$$

$$\text{sat } g_j(\tau) = \begin{cases} a_j, & g_j(\tau) \leq a_j, \\ g_j(\tau), & a_j < g_j(\tau) < b_j, \\ b_j, & g_j(\tau) \geq b_j, \end{cases} \quad \forall \tau \in [0, 1]$$

$$g_j(\tau) = \sum_{i=0}^P c_i^j S(i, \tau).$$

В качестве базисной функций  $S(i, \tau)$ , можно использовать сплайны:

$$S(i, \tau) = Sp^*(\tau P - i), \quad h = \frac{1}{P}, \quad i = 0, 1, \dots, P - \text{финитные функции, порожденные}$$

сплайнами, на отрезке  $[0, 1]$ :

$$Sp^*(t) = \begin{cases} 2^{p-1} (1+t)^p, & t \in [-1, -\frac{1}{2}], \\ 1 - 2^{p-1} |t|^p, & t \in [-\frac{1}{2}, \frac{1}{2}], \\ 2^{p-1} (1-t)^p, & t \in [\frac{1}{2}, 1], \\ 0, & t \notin [-1, 1], \end{cases}$$

где  $Sp^*(t)$  при  $p = 0$  задает кусочно-постоянный сплайн, при  $p = 1$  – кусочно-линейный (крышки), при  $p = 2$  – квадратичный, при  $p = 3$  – кубический.



## Методика подсчета функционала

1. На каждой итерации выбранным метаэвристическим алгоритмом оптимизации генерируется расширенный вектор-столбец

$$(h_1, \dots, h_p \mid c_0^1, \dots, c_p^1 \mid \dots \mid c_0^q, \dots, c_p^q)^T .$$

2. Интегрируя уравнение (6) с заданным начальным условием и управлением, вычисляемым по формуле (7), на отрезке  $[0,1]$ , получаем решение  $x(\tau), \tau \in [0,1]$ .

3. Найти величину критерия качества  $I = F(t_f, x(t_f)) = F\left(\sum_{k=0}^{P-1} h_{k+1}, x(1)\right)$ .

### Алгоритм оптимизации, имитирующий поведение стаи серых волков

Метод серых волков (Grey Wolf Optimizer – GWO), решающий задачу оптимизации целевой функции многих переменных на множестве допустимых решений  $D$ , т.е.  $f(x) = f(x_1, \dots, x_n) \rightarrow \min_{x \in D}$ , моделирует процесс последовательной охоты за добычей [12]. При этом члены стаи используют информацию о положении трех лидеров стаи, имеющих наибольшую информацию о положении жертвы, а также процедуры приближения к ней, последующего окружения и нападения. Его можно отнести к биоинспирированным методам, порожденным наблюдением за природными явлениями.

В начале процесса поиска на множестве допустимых решений  $D$  при использовании равномерного закона распределения генерируется некоторый набор начальных позиций (волков в стае):  $I = \{x^j = (x_1^j, x_2^j, \dots, x_n^j)^T, j = 1, \dots, NP\} \subset D$ , где  $x^j$  – вектор, образованный координатами волка с номером  $j$ ,  $NP$  – общее число волков в

стае. Поскольку в задаче оптимизации не известно положение точки экстремума (положение жертвы), то члены стаи волков используют информацию о текущем положении трех лидеров, считая, что они располагают более точной информацией об искомой точке минимума целевой функции. Поэтому среди волков в стае находятся три последовательно лучших  $(\alpha, \beta, \gamma)$  по величине целевой функции  $f(x)$ :  $x^\alpha, x^\beta, x^\gamma$ . Каждый волк в стае на каждой итерации изменяет свою позицию с учетом ее сравнения с тремя найденными наилучшими положениями:

$$x^j(k+1) = \frac{x^{j,1}(k+1) + x^{j,2}(k+1) + x^{j,3}(k+1)}{3},$$

$$x^{j,1}(k+1) = x^\alpha(k) - A_\alpha^j \otimes D_\alpha^j(k),$$

$$x^{j,2}(k+1) = x^\beta(k) - A_\beta^j \otimes D_\beta^j(k),$$

$$x^{j,3}(k+1) = x^\gamma(k) - A_\gamma^j \otimes D_\gamma^j(k),$$

$$D_\alpha^j(k) = \left| C_\alpha^j \otimes x^\alpha(k) - x^j(k) \right|,$$

$$D_\beta^j(k) = \left| C_\beta^j \otimes x^\beta(k) - x^j(k) \right|,$$

$$D_\gamma^j(k) = \left| C_\gamma^j \otimes x^\gamma(k) - x^j(k) \right|,$$

где  $k$  – номер итерации,  $x^j(k+1), x^j(k)$  – последующая и текущая позиции волков,

$j=1, \dots, NP$ ;  $\otimes$  – операция поэлементного произведения векторов по Адамару,

$A_\alpha^j, A_\beta^j, A_\gamma^j$  – векторы, которые находятся по правилу  $A_m^j = 2a \otimes r_1 - a, m = \alpha, \beta, \lambda; r_1$  –  $n$ -

мерный вектор с координатами, описываемыми равномерным на отрезке  $[0,1]$

законом распределения;  $a$  – вектор с одинаковыми координатами, величина

которых уменьшается линейно по закону  $a_i = 2(1 - \frac{k}{K}), i = 1, \dots, n, K$  – максимальное

число итераций;  $C_\alpha^j, C_\beta^j, C_\gamma^j$  – векторы, которые находятся по правилу

$C_m^j = 2r_2, m = \alpha, \beta, \lambda, r_2$  –  $n$ -мерный вектор, каждая координата которого описывается равномерным на отрезке  $[0,1]$  законом распределения. Также, как правило, применяется модификация алгоритма, в которой  $a_i = 2(1 - \frac{k^2}{K^2}), i = 1, \dots, n$  [13].

Комплекс программ, описанный в [14], протестирован на решении типовых стандартных задач оптимизации.

### **Алгоритм оптимизации, имитирующий поведение стаи горбатых китов**

Метод горбатых китов (Whales Optimization Algorithm – WOA) имитирует охоту стаи горбатых китов за крилем или мелкой рыбой, используя уникальный способ преследования и окружения цели – движение из глубины к поверхности океана по спирали [15].

В начале процесса поиска на множестве допустимых решений  $D$  при использовании равномерного закона распределения генерируется некоторый набор начальных позиций (горбатых китов в стае):  $I = \{x^j = (x_1^j, x_2^j, \dots, x_n^j)^T, j = 1, \dots, NP\} \subset D$ , где  $x^j$  – вектор, образованный координатами кита с номером  $j$ ,  $NP$  – число китов в стае. Поскольку в задаче оптимизации положение точки экстремума, то члены стаи китов ориентируются на позицию лидера, считая, что он располагает более точной информацией о точке экстремума (положении пищи).

В стае китов положение каждого кита характеризуется относительно лидера стаи. В процессе реализации охоты стая китов может, как исследовать множество допустимых решений в поисках косяка рыбы, так и окружать найденную добычу,

постепенно приближаясь к жертве. Одним из возможных способов приближения является спиралевидное движение вокруг потенциальной добычи.

На каждой итерации для каждой особи ( $j=1, \dots, NP$ ) генерируется случайное число  $p$  согласно равномерному закону распределения на отрезке  $[0,1]$ .

Если  $p < 0,5$ , то подсчитывается величина  $|A^j|$ , где  $A^j$  – вектор, определяемый по правилу  $A^j = 2a \otimes r_1 - a$ ;  $r_1$  –  $n$ -мерный вектор, каждая компонента которого описывается равномерным распределением на отрезке  $[0,1]$ ;  $a$  – вектор с одинаковыми компонентами, уменьшающимися линейно по закону  $a_i = 2(1 - \frac{k}{K})$ ,  $i = 1, \dots, n$ ,  $K$  – максимальное число итераций, знак  $|\cdot|$  обозначает евклидову норму вектора. Имеется модификация алгоритма, в которой координаты вектора  $a$  уменьшаются по квадратичному закону  $a_i = 2(1 - \frac{k^2}{K^2})$ ,  $i = 1, \dots, n$ .

В случае, когда  $|A^j| < 1$ , используется формула, по которой новое положение кита определяется относительно расположения жертвы:

$$x^j(k+1) = x^*(k) - A^j \otimes D^j(k),$$

$$D^j(k) = |C^j \otimes x^*(k) - x^j(k)|,$$

где  $x^*(k)$  – лучшая особь на текущей итерации,  $\otimes$  – операция поэлементного произведения векторов по Адамару,  $k$  – номер итерации,  $x^j(k+1)$ ,  $x^j(k)$  – следующее и текущее положения китов,  $j=1, \dots, NP$ ;  $C^j$  – вектор, определяемый по правилу  $C^j = 2r_2$ ,  $r_2$  –  $n$ -мерный вектор, каждая компонента которого описывается

равномерным распределением на отрезке  $[0,1]$ . В этом случае реализуется процесс разработки найденного источника пищи.

Если  $|A^j| \geq 1$ , то реализуется исследование (поиск новых источников пищи) на множестве допустимых решений:

$$x^j(k+1) = x_{rand}(k) - A^j \otimes D^j(k),$$

$$D^j(k) = |C^j \otimes x_{rand}(k) - x^j(k)|,$$

где  $x_{rand}(k)$  – случайно выбранная особь на текущей итерации.

Если  $p \geq 0,5$ , то новое положение кита определяется по формуле

$$x^j(k+1) = D^{*j}(k) \cdot e^{\beta l} \cdot \cos(2\pi l) + x^*(k),$$

$$D^{*j}(k) = |x^*(k) - x^j(k)|,$$

где  $l$  – случайная величина, распределенная равномерно на отрезке  $[-1, 1]$ ,  $\beta$  – параметр логарифмической спирали.

### Постановка задачи об управлении солнечным парусом

Модель движения космического аппарата описывается системой дифференциальных уравнений [7, 16]

$$\begin{aligned} \dot{r}(t) &= u(t), \\ \dot{\theta}(t) &= \frac{v(t)}{r(t)}, \\ \dot{u}(t) &= \frac{v^2(t)}{r(t)} - \frac{\mu}{r^2(t)}(1 - \beta \cos^3 \alpha), \\ \dot{v}(t) &= -\frac{u(t)v(t)}{r(t)} + \mu\beta \frac{\sin \alpha \cos^2 \alpha}{r^2(t)}, \end{aligned} \tag{8}$$

где  $r, \theta$  – радиальная и угловая позиции соответственно (Рисунок 1);  $u, v$  – радиальная и тангенциальная скорости,  $\alpha$  – угол тангажа (переменная управления),  $\beta = 0,042$  – параметр яркости солнечного паруса,  $\mu = 1,327474512 \cdot 10^{20} \text{ м}^3 \cdot \text{с}^{-2}$  – солнечное гравитационное ускорение. Ограничение на управление:  $\alpha \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ .

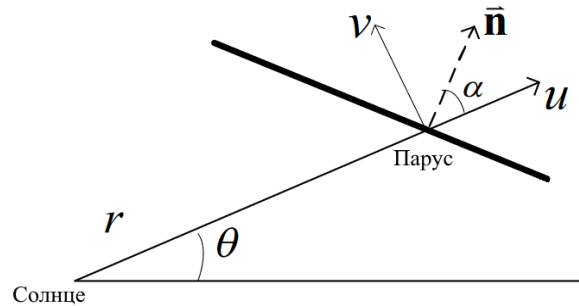


Рисунок 1. Модель солнечного паруса в двумерном пространстве

Начальные условия:

$$\begin{aligned}
 r(0) &= 1,496 \cdot 10^{11} \text{ м}, \\
 \theta(0) &= 0, \\
 u(0) &= 0 \text{ м / с}, \\
 v(0) &= 2,98 \cdot 10^4 \text{ м / с}.
 \end{aligned} \tag{9}$$

Конечные условия:

$$\begin{aligned}
 r_f &= 5,8344 \cdot 10^{10} \text{ м}, \\
 u_f &= 0 \text{ м / с}, \\
 v_f &= 4,79 \cdot 10^4 \text{ м / с}.
 \end{aligned} \tag{10}$$

Функционал качества управления:

$$I = \lambda_1 \frac{t_f}{86400} + \lambda_2 [r(t_f) - r_f]^2 + \lambda_3 [u(t_f) - u_f]^2 + \lambda_4 [v(t_f) - v_f]^2, \tag{11}$$

где  $\lambda_1, \lambda_2, \lambda_3, \lambda_4$  – весовые коэффициенты. Структура функционала соответствует реализации межпланетной миссии, заключающейся в переходе с орбиты Земли (ее задают начальные условия (9)) на орбиту Меркурия (ее определяют конечные условия (10)) за минимальное время. Первое слагаемое описывает число дней, затрачиваемое на перелет, а второе, третье и четвертое – точность выхода на заданную орбиту. Значения весовых коэффициентов характеризуют степень важности обеспечения требований к желаемой траектории. Для контроля степени выполнения конечных условий на каждой итерации контролируется величина невязки в виде модуля разности по всем координатам космического аппарата в конечный момент времени:

$$(|r(t_f) - r_f|, |u(t_f) - u_f|, |v(t_f) - v_f|). \quad (12)$$

### **Анализ полученных результатов моделирования**

На основе рассмотренных алгоритмов разработано программное обеспечение, позволяющее находить оптимальное программное управление для описанного класса задач при помощи биоинспирированных методов оптимизации. Результатом работы программы являются: минимальное значение критерия (11), а также наилучшее время, затрачиваемое на перелет; значения координат вектора состояния в найденный конечный момент времени, значения координат вектора невязки. Помимо этого, в процессе расчетов находятся наилучшие значения коэффициентов в разложении по базисным функциям и длины отрезков разбиения промежутка времени, затраченного на перелет.

Задача решается с использованием различных систем базисных функций. В частности, в качестве базисных функций используются кусочно-постоянные, кусочно-линейные, квадратичные и кубические сплайны. В связи с этим проводится сравнительный анализ результатов, полученных при различных сплайнах, а также лучшее решение сравнивается с результатами, полученными в [17].

Все необходимые расчеты проводились на ЭВМ со следующими характеристиками: процессор Intel Core i5-8250U 1.60 GHz 8 ядер, 8 ГБ оперативной памяти DDR 4.

**А.** Результаты применения метода, имитирующего поведение стаи серых волков (GWO). Эффективность алгоритма проанализирована при различных параметрах сплайна, используемого в функции управления.

Рассмотрим случай, где в качестве базисных функций взят сплайн второго порядка. При управлении солнечным парусом с применением сплайна второго порядка целевая орбита была достигнута за 941 день.

Оптимальное число коэффициентов разложения  $P = 6$ . Также определены наилучшие параметры метода:  $NP = 300$ ;  $K = 300$ ; величина максимального отрезка времени: 150000000 секунд. Установлено, что наименьшее время перелета между орбитами вместе с требуемой точностью выполнения конечных условий (10) достигается при следующих значениях весовых коэффициентов:  $\lambda_1 = 9 \cdot 10^{14}$ ,  $\lambda_2 = 3$ ,  $\lambda_3 = 9 \cdot 10^{10}$ ,  $\lambda_4 = 2 \cdot 10^{11}$ . Результаты численного эксперимента приведены в Таблице 1, на Рисунке 2 изображены графики траекторий и управления.



## Результаты работы алгоритма GWO для квадратичного сплайна

Невязка (м, м/с, м/с) $(r(t_f), u(t_f), v(t_f))$	(722190,97; 3,14; 73,83)
Коэффициенты разложения управления	(-0,539913; -0,6735712; -0,5763237; -0,6763708; -0,4169942; -0,7204174; -0,2719546)
$h_k, k = 0, 1, \dots, P-1$ (д)	(170,866; 150,611; 153,054; 144,562; 154,364; 167,876)
Конечное время $t_f$ (д)	941

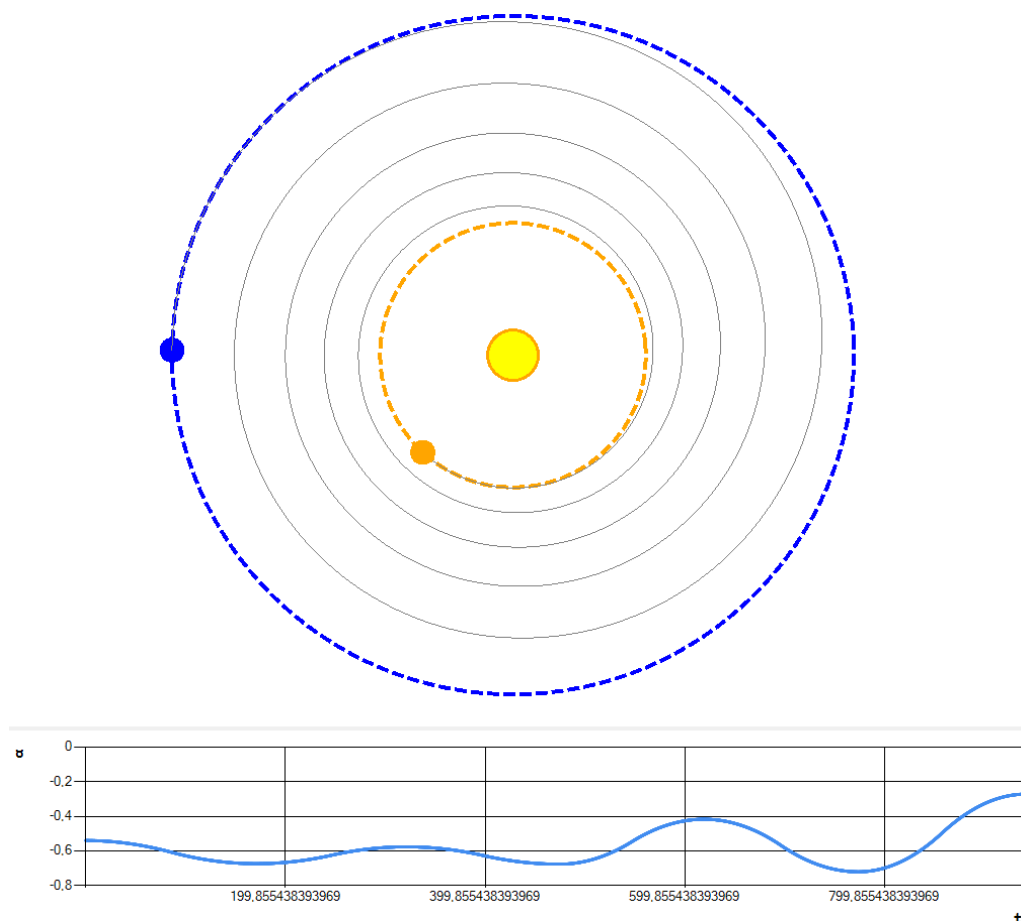


Рисунок 2. Графики траекторий и управления в случае квадратичного сплайна

**Б.** Результаты применения метода имитации стаи горбатых китов (WOA). В качестве базисных функций также взят сплайн второго порядка. В результате моделирования целевая орбита была достигнута за 940 дней.

Оптимальное число коэффициентов разложения  $P = 8$ . Также определены наилучшие параметры метода:  $NP = 400$ ;  $K = 400$ ;  $\beta = 0,01$ ; величина максимального отрезка времени: 125000000 секунд. Было установлено, что наилучшие результаты достигаются при следующих значениях весовых коэффициентов:  $\lambda_1 = 2,8 \cdot 10^{15}$ ;  $\lambda_2 = 4$ ;  $\lambda_3 = 9 \cdot 10^{11}$ ;  $\lambda_4 = 9 \cdot 10^{11}$ . Результаты численного эксперимента приведены в Таблице 2, на Рисунке 3 изображены графики траекторий и управления.

Таблица 2.

Результаты работы алгоритма для квадратичного сплайна

Невязка (м, м/с, м/с) $(r(t_f), u(t_f), v(t_f))$	(2775075,27; 6,31; 42,95)
Коэффициенты разложения управления	(-0,5796682; -0,6109807; -0,6273501; -0,5939147; -0,6480426; -0,3186944; -0,6390015; -0,6064228; -0,2844548)
$h_k, k = 0,1, \dots, P-1$ (д)	(143,233; 124,962; 122,081; 115,481; 130,249; 54,474; 127,505; 122,034)
Конечное время $t_f$ (д)	940

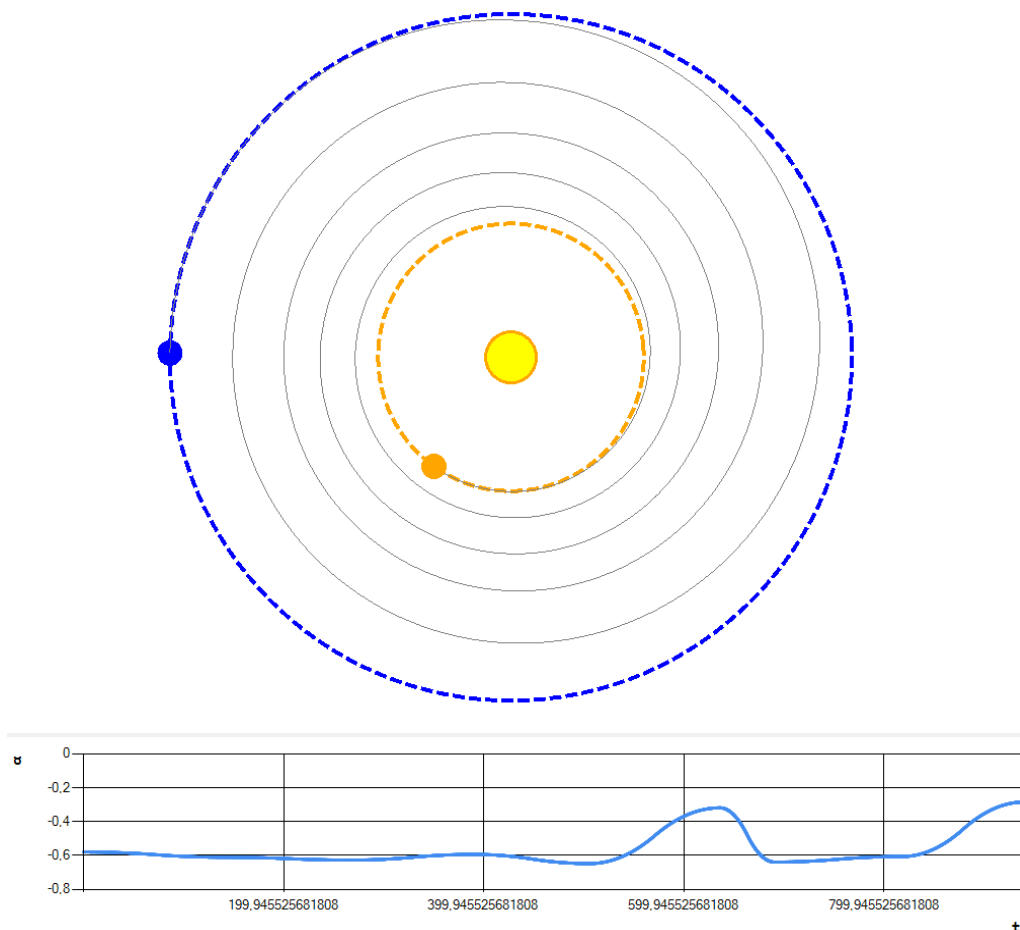


Рисунок 3. Графики траекторий и управления в случае квадратичного сплайна

Для каждого из методов был рассмотрен случай, когда в качестве базисных функций использовались кубические сплайны. В Таблице 3 представлены результаты работы программы при использовании алгоритмов с различными функциями управления. Из таблицы видно, что применение метода, имитирующего поведение стаи серых волков (GWO), позволяет достичь необходимой орбиты за меньшее время, нежели метод, имитирующий поведение стаи горбатых китов (WOA), при использовании кусочно-постоянных и кусочно-линейных сплайнов. При использовании квадратичных сплайнов метод GWO оказался менее эффективен. Для случая использования кубических сплайнов методы показали примерно одинаковые результаты, но для их достижения методу GWO

потребовалось использовать популяции удвоенного размера и вдвое больше итераций.

Таблица 3.

Сравнение полученных результатов

Подход	Затраченное время (дней)		Размер популяции		Число итераций	
	GWO	WOA	GWO	WOA	GWO	WOA
Кусочно- постоянные сплайны	942	943	500	350	400	350
Кусочно- линейные сплайны	942	944	500	500	500	500
Квадратичные сплайны	941	940	300	400	300	400
Кубические сплайны	939	938	600	300	600	300

Во всех случаях при использовании функций управления с различными сплайнами, алгоритм успешно справился с задачей управления перспективным космическим аппаратом с солнечным парусом.

В Таблице 4 представлено сравнение времени работы метода, имитирующего поведение стаи серых волков, и метода, имитирующего поведение стаи горбатых китов, при различных параметрах размера популяции и числа итераций. Время работы рассчитывается как среднее арифметическое продолжительности работы трех запусков алгоритма с одинаковыми параметрами.

Сравнение времени работы алгоритмов при различных параметрах

Размер популяции	Число итераций	Время работы	
		GWO	WOA
100	100	1 мин. 21 сек.	1 мин. 21 сек.
200	200	5 мин. 34 сек.	5 мин. 21 сек.
300	300	12 мин. 5 сек.	11 мин. 47 сек.
400	400	22 мин. 32 сек.	22 мин. 08 сек.

Рассмотрим решения, полученные алгоритмами GWO и WOA, в сравнении с другими методами. Так, в [18] с помощью кусочно-постоянного управления был произведен переход с орбиты Земли на орбиту Меркурия за 961 день, а для кусочно-линейного за 953 дня. В [15, 17-19] были найдены управления, с помощью которых поставленная задача была решена за 933,9, 1210, 1043,3 дней соответственно.

### Вывод

Таким образом, установлена состоятельность применения биоинспирированных методов оптимизации: метода, имитирующего поведение стаи серых волков, и метода, имитирующего поведение стаи горбатых китов, в прикладных задачах управления движением. Результаты рассмотренных алгоритмов превосходят большинство полученных ранее решений, что позволяет сделать вывод об успешном решении поставленной задачи об управлении солнечным парусом при перелете от Земли до Меркурия. Использование рассмотренных методов не ограничивается задачей межпланетных перелетов, они были успешно применены в различных типовых технических задачах оптимизации [20].

## Список источников

1. Panovski V.N., Pantelev A.V. Meta-heuristic interval methods of search of optimal in average control of nonlinear determinate systems with incomplete information about its parameters // Journal of Computer and Systems Sciences International, 2017, 56(1), pp. 52–63. DOI: [10.1134/s1064230717010117](https://doi.org/10.1134/s1064230717010117)
2. Pantelev A.V., Pis'mennaya V.A. Application of a memetic algorithm for the optimal control of bunches of trajectories of nonlinear deterministic systems with incomplete feedback // Journal of Computer and Systems Sciences International, 2018, 57(1), pp. 25–36. DOI: [10.1134/s1064230718010082](https://doi.org/10.1134/s1064230718010082)
3. Пантелеев А.В., Каранэ М.М.С. Применение гибридного мультиагентного метода интерполяционного поиска в задаче о стабилизации спутника // Труды МАИ. 2021. № 117. URL: <http://trudymai.ru/published.php?ID=156249>. DOI [10.34759/trd-2021-117-10](https://doi.org/10.34759/trd-2021-117-10)
4. Федоренко Р.П. Приближенное решение задач оптимального управления. – М.: Наука, 1978. – 488 с.
5. Казмерчук П.В., Вернигора Л.В. Метод линеаризации в задачах перелета космических аппаратов с электроракетной двигательной установкой на геостационарную орбиту // Труды МАИ. 2020. № 115. URL: <http://trudymai.ru/published.php?ID=119924>. DOI: [10.34759/trd-2020-115-09](https://doi.org/10.34759/trd-2020-115-09)
6. Бортакровский А.С., Урюпин И.В. Оптимизация маршрутов непрерывно-дискретного движения управляемых объектов при наличии препятствий // Труды МАИ. 2020. № 113. URL: <http://trudymai.ru/published.php?ID=118185>. DOI: [10.34759/trd-2020-113-17](https://doi.org/10.34759/trd-2020-113-17)

7. Вернигора Л.В., Казмерчук П.В. Оптимизация траекторий КА с малой тягой методом линеаризации // Труды МАИ. 2019. № 106. URL: <http://trudymai.ru/published.php?ID=105759>
8. Rybakov K.A. Solving the nonlinear problems of estimation for navigation data processing using continuous particle filter // Gyroscopy and Navigation, 2019, no. 10(1), pp. 27–34. DOI:[10.1134/S2075108719010061](https://doi.org/10.1134/S2075108719010061)
9. Rybakov K.A. Spectral method of analysis and optimal estimation in linear stochastic systems // International Journal of Modeling Simulation and Scientific Computing, 2020, no. 11(3), pp. 2050022. DOI:[10.1142/S1793962320500221](https://doi.org/10.1142/S1793962320500221)
10. Урюпин И.В. Синтез оптимальных кусочно-гладких аппроксимаций траекторий движения летательных аппаратов // Труды МАИ. 2018. № 100. URL: <http://trudymai.ru/published.php?ID=93440>
11. Luus R. Iterative dynamic programming. Chapman and Hall/CRC, Boca Raton, USA, 2000, 344 p.
12. Mirjalili S., Mirjalili S.M., Lewis A. Grey wolf optimizer //Advances in Engineering Software, 2014, vol. 69, pp. 46–61. DOI:[10.1016/j.advengsoft.2013.12.007](https://doi.org/10.1016/j.advengsoft.2013.12.007)
13. Mittal N., Singh U., Sohi B.S. Modified grey wolf optimizer for global engineering optimization // Applied Computational Intelligence and Soft Computing, 2016. DOI:[10.1155/2016/7950348](https://doi.org/10.1155/2016/7950348)
14. Пантелеев А.В., Беляков И.А. Разработка программного обеспечения метода глобальной оптимизации, имитирующего поведение стаи серых волков //

Моделирование и анализ данных. 2021. Т. 11. № 2. С. 59–73. DOI:  
[10.17759/mda.2021110204](https://doi.org/10.17759/mda.2021110204)

15. Mirjalili S., Lewis A. The Whale Optimization Algorithm // *Advances in Engineering Software*, 2016, vol. 95, pp. 51-67. DOI:[10.1016/j.advengsoft.2016.01.008](https://doi.org/10.1016/j.advengsoft.2016.01.008)

16. Wang Y., Zhu M., Wei Y. Solar Sail Spacecraft Trajectory Optimization Based on Improved Imperialist Competitive Algorithm // *Proceedings of the 10th World Congress on Intelligent Control and Automation*, 2012, pp. 191–195. DOI:[10.1109/WCICA.2012.6357865](https://doi.org/10.1109/WCICA.2012.6357865)

17. Пантелеев А.В., Пановский В.Н. Прикладное применение интервального метода взрывов для поиска оптимального программного управления солнечным парусом // *Вестник НПО имени С.А. Лавочкина*. 2016. № 4(34). С. 110-117.

18. McInnes C.R., Hughes G.W. Low Cost Mercury Orbiter and Sample Return Missions Using Solar Sail Propulsion // *Aeronautical Journal*, 2003, no. 107(1074), pp. 469–478.

19. Hughes G.W., McDonald M., McInnes C.R. Terrestrial Planet Sample Return Using Solar Sail Propulsion // *Acta Astronautica*, 2006, no. 59(8-11). pp. 797–806. DOI:[10.1016/j.actaastro.2005.07.019](https://doi.org/10.1016/j.actaastro.2005.07.019)

20. Kamboj V.K., Bath S.K., Dhillon J.S. Solution of non-convex economic load dispatch problem using Grey Wolf Optimizer // *Neural Computing and Applications*, 2015, vol. 27, pp. 1301-1316. DOI:[10.1007/s00521-015-1934-8](https://doi.org/10.1007/s00521-015-1934-8)



## References

1. Panovskiy V.N., Panteleev A.V. Meta-heuristic interval methods of search of optimal in average control of nonlinear determinate systems with incomplete information about its parameters, *Journal of Computer and Systems Sciences International*, 2017, 56(1), pp. 52–63. DOI: [10.1134/s1064230717010117](https://doi.org/10.1134/s1064230717010117)
2. Panteleev A.V., Pis'mennaya V.A. Application of a memetic algorithm for the optimal control of bunches of trajectories of nonlinear deterministic systems with incomplete feedback, *Journal of Computer and Systems Sciences International*, 2018, 57(1), pp. 25–36. DOI: [10.1134/s1064230718010082](https://doi.org/10.1134/s1064230718010082)
3. Panteleev A.V., Karane M.M.S. *Trudy MAI*, 2021, no. 117. URL: <http://trudymai.ru/eng/published.php?ID=156249>. DOI [10.34759/trd-2021-117-10](https://doi.org/10.34759/trd-2021-117-10)
4. Fedorenko R.P. *Priblizhennoe reshenie zadach optimal'nogo upravleniya* (Approximate solution of optimal control problems), Moscow, Nauka, 1978, 488 p.
5. Kazmerchuk P.V., Vernigora L.V. *Trudy MAI*, 2020, no. 115. URL: <http://trudymai.ru/eng/published.php?ID=119924>. DOI: [10.34759/trd-2020-115-09](https://doi.org/10.34759/trd-2020-115-09)
6. Bortakovskii A.S., Uryupin I.V. *Trudy MAI*, 2020, no. 113. URL: <http://trudymai.ru/eng/published.php?ID=118185>. DOI: [10.34759/trd-2020-113-17](https://doi.org/10.34759/trd-2020-113-17)
7. Vernigora L.V., Kazmerchuk P.V. *Trudy MAI*, 2019, no. 106. URL: <http://trudymai.ru/eng/published.php?ID=105759>
8. Rybakov K.A. Solving the nonlinear problems of estimation for navigation data processing using continuous particle filter, *Gyroscopy and Navigation*, 2019, no. 10(1), pp. 27–34. DOI: [10.1134/S2075108719010061](https://doi.org/10.1134/S2075108719010061)

9. Rybakov K.A. Spectral method of analysis and optimal estimation in linear stochastic systems, *International Journal of Modeling Simulation and Scientific Computing*, 2020, no. 11(3), pp. 2050022. DOI:[10.1142/S1793962320500221](https://doi.org/10.1142/S1793962320500221)
10. Uryupin I.V. *Trudy MAI*, 2018, no. 100. URL: <http://trudymai.ru/eng/published.php?ID=93440>
11. Luus R. *Iterative dynamic programming*. Chapman and Hall/CRC, Boca Raton, USA, 2000, 344 p.
12. Mirjalili S., Mirjalili S.M., Lewis A. Grey wolf optimizer, *Advances in Engineering Software*, 2014, vol. 69, pp. 46–61. DOI:[10.1016/j.advengsoft.2013.12.007](https://doi.org/10.1016/j.advengsoft.2013.12.007)
13. Mittal N., Singh U., Sohi B.S. Modified grey wolf optimizer for global engineering optimization, *Applied Computational Intelligence and Soft Computing*, 2016. DOI:[10.1155/2016/7950348](https://doi.org/10.1155/2016/7950348)
14. Panteleev A.V., Belyakov I.A. *Modelirovanie i analiz dannykh*, 2021, vol. 11, no. 2, pp. 59–73. DOI: [10.17759/mda.2021110204](https://doi.org/10.17759/mda.2021110204)
15. Mirjalili S., Lewis A. The Whale Optimization Algorithm, *Advances in Engineering Software*, 2016, vol. 95, pp. 51-67. DOI:[10.1016/j.advengsoft.2016.01.008](https://doi.org/10.1016/j.advengsoft.2016.01.008)
16. Wang Y., Zhu M., Wei Y. Solar Sail Spacecraft Trajectory Optimization Based on Improved Imperialist Competitive Algorithm, *Proceedings of the 10th World Congress on Intelligent Control and Automation*, 2012, pp. 191–195. DOI:[10.1109/WCICA.2012.6357865](https://doi.org/10.1109/WCICA.2012.6357865)
17. Panteleev A.V., Panovskii V.N. *Vestnik NPO imeni S.A. Lavochkina*, 2016, no. 4(34), pp. 110-117.

18. McInnes C.R., Hughes G.W. Low Cost Mercury Orbiter and Sample Return Missions Using Solar Sail Propulsion, *Aeronautical Journal*, 2003, no. 107(1074), pp. 469–478.
19. Hughes G.W., McDonald M., McInnes C.R. Terrestrial Planet Sample Return Using Solar Sail Propulsion, *Acta Astronautica*, 2006, no. 59(8-11). pp. 797–806.  
DOI:[10.1016/j.actaastro.2005.07.019](https://doi.org/10.1016/j.actaastro.2005.07.019)
20. Kamboj V.K., Bath S.K., Dhillon J.S. Solution of non-convex economic load dispatch problem using Grey Wolf Optimizer, *Neural Computing and Applications*, 2015, vol. 27, pp. 1301-1316. DOI:[10.1007/s00521-015-1934-8](https://doi.org/10.1007/s00521-015-1934-8)

Статья поступила в редакцию 24.12.2021; одобрена после рецензирования 16.01.2022; принята к публикации 21.02.2022.

The article was submitted on 24.12.2021; approved after reviewing on 16.01.2022; accepted for publication on 21.02.2022.