УДК 519.71

МУЛЬТИАГЕНТНЫЙ АЛГОРИТМ ПОИСКА ОПТИМАЛЬНОГО ПРОГРАММНОГО УПРАВЛЕНИЯ ОДНИМ КЛАССОМ ДЕТЕРМИНИРОВАННЫХ СИСТЕМ

А.В. Пантелеев, М.М.С. Каранэ

В работе предложен алгоритм, позволяющий находить оптимальное программное управление с помощью метаэвристических мультиагентных методов условной глобальной оптимизации [1]: метода, имитирующего поведение стаи рыб; метода, имитирующего популяцию криля; метода, имитирующего империалистическую конкуренцию, описанных в [2]. Эти методы основаны на процессах, происходящих в среде, имеющей множество агентов. Агенты имеют возможность обмениваться информацией для того, чтобы найти решение задачи. На основе данных алгоритмов разработано программное обеспечение в среде Microsoft Visual Studio. Оно позволяет находить оптимальное программное управление динамическим системами, линейными по ограниченному управлению, в релейном виде, а также соответствующие траектории и наилучшее значение критерия. По результатам решения специально подобранного набора тестовых задач о нахождении оптимального программного управления [3] можно сделать вывод о том, что рассмотренный алгоритм можно успешно применять и находить близкое к точному решение за приемлемое время

This paper considers the algorithm which allows finding the optimal open-loop control with multi-agents methods of conditional global optimization [1]: fish school search, krill herd, and imperialist competitive algorithm. These methods were described in [2]. They are based on processes taking place in an environment that has many agents. Agents have the opportunity to exchange information in order to find a solution of the problem. Based on these algorithms, software was developed in the Microsoft Visual Studio. It allows to find the optimal open-loop control by dynamic system, which linear on limited control, in relay form, also the trajectory and the optimal value of the criterion. According to the results of solving a specially selected set of test problems about finding optimal open-loop control [3], it can be concluded that the considered algorithm can be successfully applied and find a close to exact solution in a reasonable time.

КЛЮЧЕВЫЕ СЛОВА
Программное управление, оптимизация, мультиагентные алгоритмы, алгоритмическое обеспечение, программное обеспечение.

ДЛЯ ЦИТАТЫ
А.В. Пантелеев, М.М.С. Каранэ. Мультиагентный алгоритм поиска оптимального программного управления одним классом детерминированных систем // Моделирование и анализ данных. 2019. №3. С. 58-64.
A.V. Panteleev, M.M.S. Karane. Multi-agents algorithm of finding the optimal open-loop control for one class of deterministic systems. Modelirovaniye i analiz dannykh=Modelling and data analysis *(Russia)*. 2019, no.3, pp. 58-64.

1. ПОСТАНОВКА ЗАДАЧИ

Пусть поведение модели объекта управления описывается обыкновенным дифференциальным уравнением

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

где x – вектор состояния системы, $x = (x_1, \dots, x_n)^T \in R^n$; u – вектор управления, $u = (u_1, ..., u_q)^T \in U \subseteq R^q$, U – некоторое заданное множество допустимых значений управления, определяемое прямым произведением отрезков $[a_1, b_1] \times ... \times [a_q, b_q]$; $t \in T = [t_0, t_1]$ – промежуток времени функционирования системы; моменты начала процесса t_0 и окончания t_1 заданы; $f(t, x, u)$ – непрерывная вектор-функция; R^n – n-мерное евклидово пространство.

Начальное условие $x(t_0) = x_0$ задает начальное состояние системы.

Определим множество допустимых процессов $\boldsymbol{D}(t_0, x_0)$ как множество пар $d = (x(\cdot), u(\cdot))$, которые включают 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.

На множестве $\boldsymbol{D}(t_0, x_0)$ определим функционал качества управления

$$I(d) = F(x(t_1)), \qquad (2)$$

где $F(x)$ – заданная непрерывная функция.

Требуется найти такую пару $d^* = (x^*(\cdot), u^*(\cdot)) \in \boldsymbol{D}(t_0, x_0)$, что $I(d^*) = \min\limits_{d \in \boldsymbol{D}(t_0, x_0)} I(d)$.

Для систем (1) , линейных по управлению, структура оптимального программного управления согласно принципу максимума является релейной. Поэтому предлагается искать приближенное решение в параметрическом виде, определяемом числом моментов переключения управления и их значениями.

2. АЛГОРИТМ ПОИСКА ОПТИМАЛЬНОГО ПРОГРАММНОГО УПРАВЛЕНИЯ

Шаг 1. Инициализация. Выбрать метод из семейства мультиагентных алгоритмов (метод, имитирующий поведение стаи рыб; метод, имитирующий популяцию криля; метод, имитирующий империалистическую конкуренцию [2]) и задать его параметры. Задать количество переключений $p = 0$ в управлении $u(t)$; при этом $t_{\Pi_0} \in \{t_0, t_1\}$.

Шаг 2. Сгенерировать начальную популяцию (управления), состоящую из NP особей, на интервале изменения времени $t \in [t_0, t_1]$. Полученные $1, \dots, NP$ последовательности значений являются точками переключения $t_\Pi \in [t_0, t_1]$ в управлении $u(t)$.

Шаг 3. По сгенерированным значениям точек переключения сформировать управление

$$u_p^j(t) = a_p \chi(t - t_0) + (a_p - b_p) \sum_{k=0}^{p} (-1)^k \chi(t - t_{\Pi_k}),$$

где $\chi(t) = \begin{cases} 0, t \le 0 \\ 1, t > 0 \end{cases}$, $j \in \overline{1, NP}, \ \ p \in \overline{1, q}.$

Шаг 4. Проинтегрировать NP систем дифференциальных уравнений (1) с управлениями $u^1(t),\dots,u^{NP}(t)$ методом Рунге–Кутты 4-го порядка. Для каждой особи получить соответствующие траектории $x_1^1,\dots,x_1^{NP},\dots,x_n^1,\dots,x_n^{NP}$ и вычислить значения функционала $I^1,\dots,I^{NP}$.

Шаг 5. Выполнить очередную итерацию выбранного метода минимизации функционала (2). Получить новые положения особей $1',\dots,NP'$ (значения точек переключения). Перейти к шагу 3.

Шаг 6. Цикл (шаг 3 – шаг 5) заканчивается по достижению заданного числа итераций. Выбирается лучшая особь (набор точек переключения управления). Соответствующие ей управление и траектория, а также значение функционала I_p^* принимаются за приближенное решение задачи с числом переключений, равным p.

Шаг 7. Если $I_p^* < I_{p-1}^*$ (условие проверяется при $p \ge 1$), то положить $p = p+1$ и перейти к шагу 2. Если $I_p^* \ge I_{p-1}^*$ то процедура поиска оптимального программного управления завершается и в качестве ответа выбирается найденное управление с p переключениями.

3. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

На основе изложенного алгоритма было разработано программное обеспечение. Для его создания использовалась среда разработки Microsoft Visual Studio, язык программирования C#.

На главной форме (рис. 1) метода пользователь может выбрать задачу для нахождения оптимального программного управления, задать количество переключении в управлении, выбрать метод оптимизации и указать его параметры.

Результатом работы программы являются координаты точек $x_1(t_1)$, $x_2(t_1)$, оптимальное значение функционала I и координаты точек переключения. По окончании поиска оптимального управления программа изображает графики функции управления и траектории.

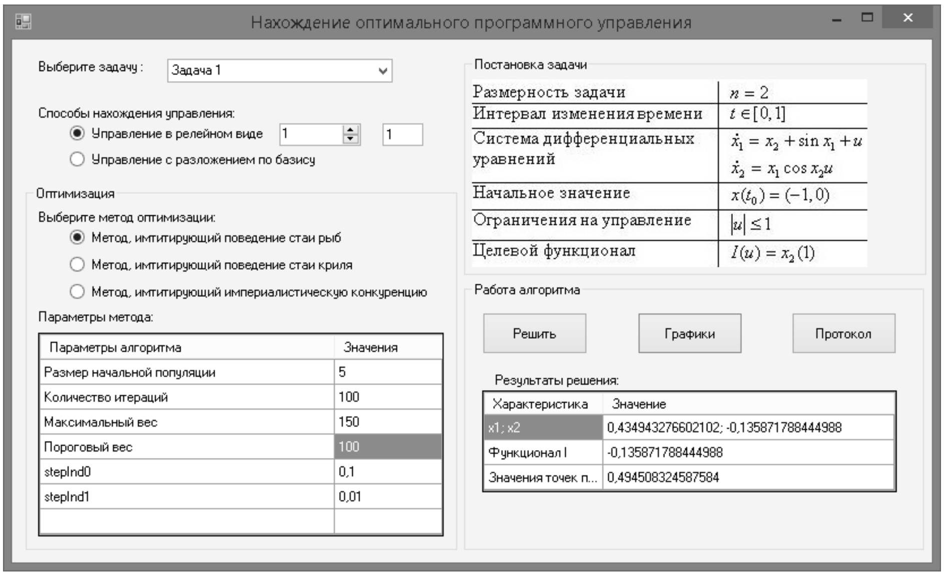

Рис. 1 Общий вид интерфейса программы

4. РЕШЕНИЕ ЗАДАЧ ПОИСКА ОПТИМАЛЬНОГО ПРОГРАММНОГО УПРАВЛЕНИЯ

Задача 1 [3]. Постановка задачи (табл. 1):

Таблица 1. Постановка задачи 1

Размерность вектора состояния	$n = 2$
Интервал времени	$t \in [0,1]$
Ограничения на управление	$-1 \le u \le 1$
Начальное значение	$x(0) = (0,0)$
Система дифференциальных уравнений	$\begin{cases} \dot{x}_1 = x_2 + \sin x_1 + u \\ \dot{x}_2 = x_1 \cos x_2 u \end{cases}$
Целевой функционал	$I(u) = x_2(1)$

Найденное наилучшее количество переключений: $p = 1$.

Метод оптимизации и его параметры: метод, имитирующий поведение стаи рыб ($NP = 5,\ ITER = 100,\ W_{scale} = 150,\ W = 100,\ step_{vol} = 0{,}1,\ step_{ind} = 0{,}01$); метод, имитирующий поведение стаи криля ($NP = 10,\ ITER = 100,\ N_{\max} = 0{,}01,\ mu = 0{,}5$, $V_f = 0{,}02$, $D_{\max} = 0{,}005,\ c_i = 0{,}2.$); метод, имитирующий империалистическую конкуренцию ($N_{pop} = 150,\ N_{imp} = 15$, $ITER = 500,\ \beta = 0{,}2$, $\gamma = 0{,}02$, $\xi = 0{,}01$).

Результаты решения задачи представлены в табл. 2.

Таблица 2. Результаты решения задачи 1

Метод	Координаты точек $(x_1(1), x_2(1))$	Координата точки переключения	Значение функционала I
Метод, имитирующий поведение стаи рыб	(0,43494; –0,13587)	0,49	–0,13587
Метод, имитирующий поведение стаи криля	(0,42459; –0,13478)	0,48	–0,13571
Метод, имитирующий империалистическую конкуренцию	(0,44665; –0,13598)	0,5	–0,13598
Известное решение [3]	(0,440804; –0,13593)	0,5	–0,13599

Графики оптимальных траекторий и управления представлены на рис. 2:

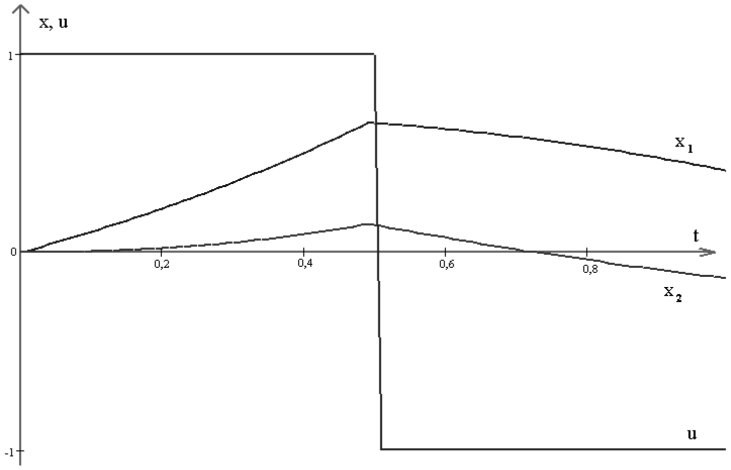

Рис. 2 Графики оптимальных траекторий и управления

Задача 2 [3]. Постановка задачи (табл. 3):

Таблица 3. Постановка задачи 2

Размерность вектора состояния	$n=2$
Интервал времени	$t\in[0,2]$
Ограничения на управление	$-1\le u\le 2$
Начальное значение	$x(0)=(-1,0)^T$
Система дифференциальных уравнений	$\begin{cases}\dot{x}_1 = x_2^2+u\\ \dot{x}_2 = 8\sin x_1 + x_1 - x_2 - u\end{cases}$
Целевой функционал	$I(u)=-x_2(2)$

Найденное наилучшее количество переключений: $p=4$

Метод оптимизации и его параметры: метод, имитирующий поведение стаи рыб ($NP=30,\ ITER=500,\ W_{scale}=5000,\ W=4500,\ step_{vol}=0{,}1,\ step_{ind}=0{,}01$); метод, имитирующий поведение стаи криля ($NP=40,\ ITER=1000,\ N_{\max}=0{,}01$, $mu=0{,}5$, $V_f=0{,}02$, $D_{\max}=0{,}005$, $c_i=0{,}2.$); метод, имитирующий империалистическую конкуренцию ($N_{pop}=150,\ N_{imp}=15$, $ITER=500,\ \beta=0{,}2$, $\gamma=0{,}02$, $\xi=0{,}01$).

Результаты решения задачи представлены в табл. 4.

Таблица 4. Результаты решения задачи 2

Метод	Координаты точек $(x_1(2),x_2(2))$	Координаты точек переключения	Значение функционала I
Метод, имитирующий поведение стаи рыб	(16,50987; 6,37294)	(0,62;1,52;1,74;1,93)	–16, 50987
Метод, имитирующий поведение стаи криля	(15,93114; 6,18436)	(0,57;0,58;0,58;1,85)	–15,93114
Метод, имитирующий империалистическую конкуренцию	(16,67731; 6,55339)	(0,57;1,35;1,52;1,86)	–16,67731
Известное решение [3]	(16,76268; 6,35095)	(0,5;1,25;1,5;1,8)	–16,76268

Графики оптимальных траекторий и управления представлены на рис. 3:

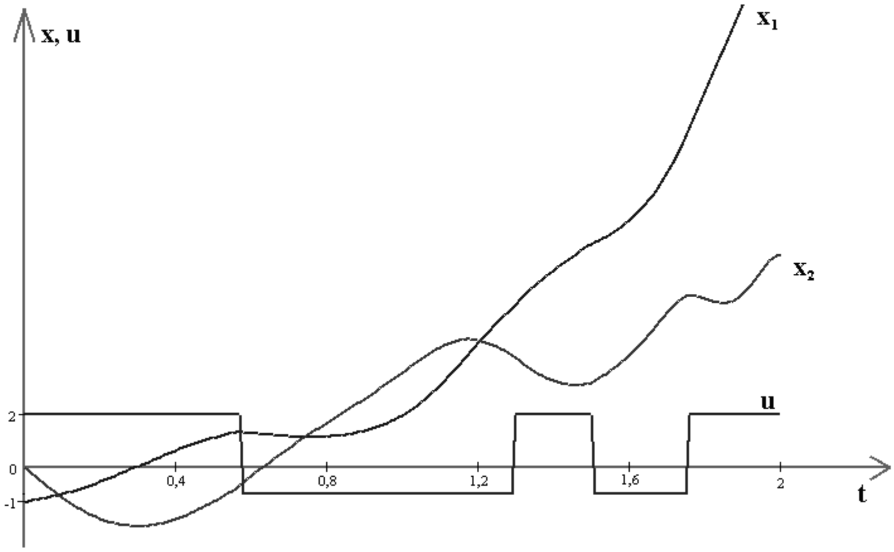

Рис. 3 Графики оптимальных траекторий и управления

Задача 3 [3]. Постановка задачи (табл. 5):

Таблица 5. Постановка задачи 3

Размерность вектора состояния	$n = 2$
Интервал времени	$t \in [0;1,6]$
Ограничения на управление	$-2 \le u \le 1$
Начальное значение	$x(0) = (1,0)^T$
Система дифференциальных уравнений	$\begin{cases} \dot{x}_1 = \dfrac{1}{\cos x_1 + 2} + 3\sin x_2 + u \\ \dot{x}_2 = x_1 + x_2 + u \end{cases}$
Целевой функционал	$I(u) = -x_1(1,6) + \dfrac{1}{2} x_2(1,6)$

Найденное наилучшее количество переключений: $p = 1$

Метод оптимизации и его параметры: метод, имитирующий поведение стаи рыб ($NP = 5$, $ITER = 100$, $W_{scale} = 300$, $W = 250$, $step_{vol} = 0,1$, $step_{ind} = 0,01$); метод, имитирующий поведение стаи криля ($NP = 10$, $ITER = 100$, $N_{\max} = 0,01$, $mu = 0,5$, $V_f = 0,02$, $D_{\max} = 0,005$, $c_i = 0,2.$); метод, имитирующий империалистическую конкуренцию ($N_{pop} = 40$, $N_{imp} = 4$, $ITER = 500$, $\beta = 0,2$, $\gamma = 0,02$, $\xi = 0,01$).

Результаты решения задачи представлены в табл. 6.

Таблица 6. Результаты решения задачи 3

Метод	Координаты точек $(x_1(1,6), x_2(1,6))$	Координата точки переключения	Значение функционала I
Метод, имитирующий поведение стаи рыб	(3,43034; 12,81994)	1,24	–2,97963
Метод, имитирующий поведение стаи криля	(3,52562; 13.00372)	1,27	–2,97624
Метод, имитирующий империалистическую конкуренцию	(3,93412; 13,53938)	1,39	–2,83557
Известное решение [3]	(3,46114; 12,884)	1,26	–2,98086

Графики оптимальных траекторий и управления представлены на рис. 4:

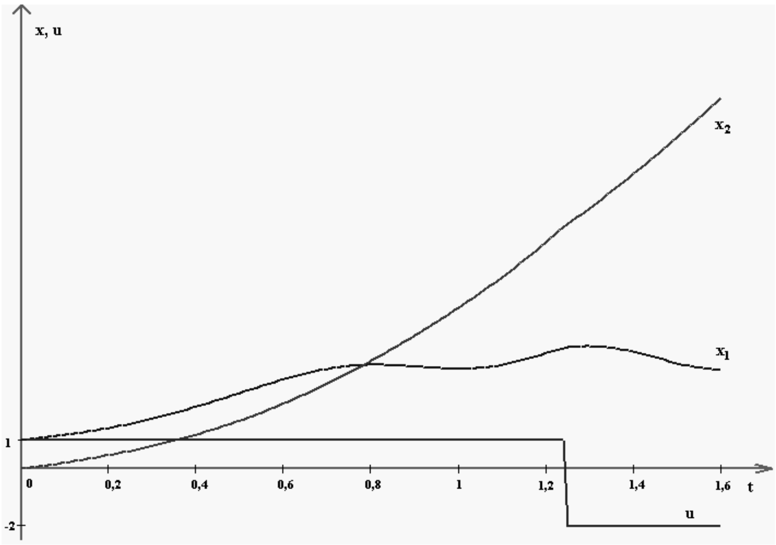

Рис. 4 Графики оптимальных траекторий и управления

ЗАКЛЮЧЕНИЕ

Разработанный алгоритм позволяет находить оптимальное программное управление с помощью мультиагентных метаэвристических методов условной глобальной оптимизации. Он позволяет решать задачи, в которых кроме глобального экстремума имеются один или несколько локальных, за приемлемое время.

ЛИТЕРАТУРА

1. Пантелеев А.В., Скавинская Д.В., Алешина Е.А. Метаэвристические алгоритмы поиска оптимального программного управления. М.: ИНФРА–М, 2016.
2. Каранэ М.М.С. Сравнительный анализ мультиагентных методов условной глобальной оптимизации // Материалы IV международной научно-практической конференции «Информатизация инженерного образования», Москва, 23–26 октября, 2018. – С. 128–133.
3. Финкельштейн Е.А. Вычислительные технологии аппроксимации множества достижимости управляемой системы: дисс. на соиск. учен. степ. канд. техн. наук (05.13.01). Институт динамики систем и теории управления, Иркутск, 2018.

Работа поступила 19.02.2019г.