

УДК 517.977.58

РЕШЕНИЕ ЗАДАЧИ О ПЕРЕВОДЕ СПУТНИКА МЕЖДУ ОРБИТАМИ С ПОМОЩЬЮ ГИБРИДНОГО МЕМЕТИЧЕСКОГО АЛГОРИТМА¹

А.В. ПАНТЕЛЕЕВ, В.А. ПИСЬМЕННАЯ

В работе предложен гибридный меметический алгоритм (МА) для решения задачи поиска оптимального программного управления нелинейными непрерывными детерминированными системами, основанный на концепции мема, которым является одно из перспективных решений, полученных в ходе реализации процедуры поиска экстремума. На основе предложенного алгоритма сформирован комплекс программ на языке C#. Приведено решение задачи о переводе спутника с одной орбиты на другую.

Ключевые слова: оптимальное управление, глобальный экстремум, нелинейные непрерывные детерминированные системы, меметический алгоритм, перевод спутника.

Введение

Термин «меметические алгоритмы» (МА) широко используется в качестве обозначения взаимодействия эволюционного или другого подхода, основанного на понятии популяции, и индивидуального обучения особей либо другой локальной процедуры улучшения решения для задач поиска глобального оптимума.

Теория «универсального дарвинизма», разработанная в [1], полагает, что понятие эволюции применимо не только к биологическим системам, но и к любой сложной системе, которой присущи принципы наследования, изменения и селекции, т.е. все принципы развития. Таким образом, наука меметика представляет собой аналог генетики в развитии культуры. Термин «мем» был введён Р. Докинзом как «единица передачи культурной информации, распространяемая от одной особи к другой посредством имитации, научения и др.» [2].

Термин «меметический алгоритм» был впервые предложен в [3], где он рассматривался как гибрид генетического алгоритма и процедуры индивидуального обучения для уточнения решения задачи. На этапе индивидуального обучения решение (особь или её генотип) заменяется новым (обученным) решением в случае, если новое решение имеет большую приспособленность, независимо от остальной части популяции. Таким образом, происходит так называемое культурное развитие особи, которое затем передаётся её потомкам в течение последующих поколений [4].

К первому поколению МА относятся алгоритмы-гибриды эволюционного подхода, используемого для глобального поиска, и культурной эволюционной составляющей.

Второе поколение МА использует принципы меметической трансмиссии и селекции в своих реализациях. В таких МА меметический материал кодируется как часть генотипа. Далее декодированный мем каждой особи используется для локального уточнения решения. Меметический материал передаётся с помощью простого механизма наследования от родителей к потомкам.

В качестве третьего поколения МА выделяют самогенерирующиеся МА. В отличие от второго поколения МА, в которых предполагается, что используемые мемы известны изначально, третье поколение МА использует локальный поиск, основанный на некоторых правилах, для пополнения множества кандидатов в решение в рамках эволюционной системы, тем самым учитывая регулярно повторяющиеся структуры и черты данной области задач.

В разработанном алгоритме культурная эволюционная составляющая реализуется в ходе решения подзадачи оптимизации любым из двух методов: с помощью метода муравьиных колоний [5; 10] или методом имитации отжига [6; 10]. В ходе культурной эволюции информация о мемах используется для генерации более совершенной в терминах решаемой задачи особи. В качестве процедур улучшения решения используются метод покоординатного поиска и перекомпоновки пути (path-relinking).

¹ Работа выполнена при финансовой поддержке РФФИ (проект № 12–08–00892–а).

1. Постановка задачи

Поведение модели объекта управления описывается дифференциальным уравнением $\dot{x} = f(t, x(t), u(t))$, где x – вектор состояния системы, $x \in R^n$; u – вектор управления, $u = (u_1, \dots, u_q)^T \in U \subseteq R^q$; U – множество допустимых значений управления, представляющее собой прямое произведение отрезков $[a_i, b_i]$, $i = 1, 2, \dots, q$; t – непрерывное время, $t \in T = [t_0, t_N]$, начальный t_0 и конечный t_N моменты времени заданы; $f(t, x, u) = (f_1(t, x, u), \dots, f_n(t, x, u))^T$ – непрерывно-дифференцируемая вектор-функция.

Начальное состояние системы задано: $x(t_0) = x_0$. Ограничения на конечное состояние: $x_j(t_N) = x_{j,N}$, где $j = 1, \dots, l$; $0 \leq l \leq n$. Предполагается, что при управлении используется информация только о непрерывном времени t , т.е. применяется так называемое программное управление.

Множество допустимых процессов $D(t_0, x_0)$ – это множество пар $d = (x(t), u(t))$, включающих траекторию $x(t)$ и кусочно-непрерывное управление $u(t)$, $u(t) \in U$, удовлетворяющих дифференциальному уравнению, начальному условию, а также ограничениям на конечное состояние.

На множестве допустимых процессов определен функционал качества управления $I(d) = \int_{t_0}^{t_N} f^0(t, x(t), u(t)) dt + F(x(t_N))$, где $f^0(t, x, u)$, $F(x)$ – заданные непрерывные функции.

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

2. Стратегия поиска решений

Для решения задачи поиска оптимального управления непрерывной системой предлагается осуществить переход к задаче поиска оптимального управления дискретной детерминированной системой, а затем построить решение исходной задачи путем интерполяции значений в узлах сетки. Предлагается искать приближенное решение задачи (управление) в виде кусочно-постоянных вектор-функций.

Выберем N точек из отрезка $[t_0, t_N]$, в которых будем искать оптимальное дискретное управление. Пусть $t_0 = 0$, тогда $h = \frac{t_N - t_0}{N}$ – величина шага. В результате получим равномерную сетку $\{0, h, 2h, \dots, Nh\}$. Управлению $u(t)$ будет соответствовать дискретное управление $u(\cdot) = \{u(0), u(1), \dots, u(N-1)\}$, где $u(\hat{t}) = u(\hat{t}h)$, $\hat{t} = 0, 1, \dots, N-1$. Дифференциальные уравнения модели объекта управления приближенно заменяются разностными с помощью численных методов [7].

Перейдем от функционала Больца к функционалу Майера. Для этого введем дополнительную компоненту вектора состояния следующим образом: $\frac{dx_{n+1}(t)}{dt} = f^0(t, x(t), u(t))$. Начальное условие для этой компоненты имеет вид $x_{n+1}(t_0) = 0$. Таким образом, получаем расширенный вектор состояния $x = (x_1, \dots, x_n, x_{n+1}) \in R^{n+1}$.

Кроме этого, сведем задачу с ограничениями на конечное состояние к задаче со свободным правым концом, используя штрафные функции.

Функционал качества управления исследуемой системой примет вид:

$$\tilde{I}(u(0), u(1), \dots, u(N-1)) = x_{n+1}(t_N) + F(x(t_N)) + \sum_{j=1}^l R_j [x_j(t_N) - x_{j,N}]^2,$$
 где R_1, \dots, R_l – параметры штрафа, $t_N = Nh$. Данный функционал будет являться целевой функцией в рассматриваемой задаче. Выбору подлежит вектор $u = (u(0), u(1), \dots, u(N-1))^T$, который соответствует дискретному управлению $u(\cdot) = \{u(0), u(1), \dots, u(N-1)\}$.

Для оптимизации критерия будем использовать меметический алгоритм, который применим к решению задачи оптимизации целевой функции $f(x) = f(x_1, x_2, \dots, x_n)$, определенной на множестве допустимых решений $D = \{x \mid x_i \in [a_i, b_i], i = 1, 2, \dots, n\} \subseteq \mathbf{R}^n$, и позволяет найти ее условный глобальный минимум на заданном множестве, т.е. такую точку $x^* \in D$, что $f(x^*) = \min_{x \in D} f(x)$.

Для удобства изложения заменим вектор (содержащий всю информацию об управлении) $u = (u(0), u(1), \dots, u(N-1))^T = (u_1(0), \dots, u_q(0); u_1(1), \dots, u_q(1); \dots; u_1(N-1), \dots, u_q(N-1))^T$ на вектор $\tilde{u} = (\tilde{u}_{1,1}, \dots, \tilde{u}_{1,q}, \tilde{u}_{2,1}, \dots, \tilde{u}_{2,q}, \dots, \tilde{u}_{N,1}, \dots, \tilde{u}_{N,q})^T = (\tilde{u}_1, \tilde{u}_2, \dots, \tilde{u}_{Nq})^T$ (заметим, что здесь $\tilde{u}_{i,j} = u_j(i-1) \in [a_j; b_j]$). Обозначим через \tilde{U} множество допустимых векторов \tilde{u} , координаты которых удовлетворяют приведенным ограничениям. Тогда задачу можно сформулировать следующим образом: требуется найти такой вектор \tilde{u}^* , что $\tilde{I}(\tilde{u}^*) = \min_{\tilde{u} \in \tilde{U}} \tilde{I}(\tilde{u})$.

Рассматриваемая целевая функция $\tilde{I}(\tilde{u})$ эквивалентна природному понятию приспособленности живого организма. Чем меньше значение целевой функции, тем более приспособлен организм. Каждый допустимый вектор $\tilde{u} = (\tilde{u}_{1,1}, \dots, \tilde{u}_{1,q}, \tilde{u}_{2,1}, \dots, \tilde{u}_{2,q}, \dots, \tilde{u}_{N,1}, \dots, \tilde{u}_{N,q})^T = (\tilde{u}_1, \tilde{u}_2, \dots, \tilde{u}_{Nq})^T \in \tilde{U} \mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)^T \in D$ является возможным решением поставленной оптимизационной задачи и называется особью. Генерируемое на каждой итерации алгоритма конечное множество решений (особей) называется популяцией.

Стратегия поиска решения предлагаемым меметическим алгоритмом включает процедуры создания начальной популяции, формирования множества $Pool$, содержащего «хорошие» решения; реализацию локального поиска, механизм обновления множества $Pool$, процедуру path-relinking [8] для улучшения имеющихся решений путем выделения взаимосвязей между элементами множества $Pool$, а также процедуру покоординатного уточняющего поиска.

3. Алгоритм

Шаг 1. Задание параметров. Задать максимальное число итераций M_{\max} ; размер популяции \mathbf{m} ; параметр σ – некоторый порог расстояния между решениями; \mathbf{K} – характерное количество элементов множества $Pool$; P – количество направляющих решений процедуры path-relinking; IM – количество повторений в уточняющем поиске; q – количество удаляемых решений из множества $Pool$ на каждой итерации; область определения S коэффициентов c_1, \dots, c_k ; параметры метода муравьиных колоний или имитации отжига в зависимости от сделанного выбора. Положить $k = 0$ – количество элементов в множестве $Pool$; $M = 0$ – номер итерации.

Шаг 2. Формирование множества $Pool$.

Шаг 2.1. Случайным образом сформировать популяцию \mathbf{I}_M . Для этого с помощью равномерного распределения \mathbf{m} раз сгенерировать последовательность из $N \cdot q$ случайных значений $\{P_{ij}^{M,p}\}$, $i = 1, \dots, N$; $j = 1, \dots, q$; $p = 1, 2, \dots, m$ на отрезке $[0, 1]$. Используя линейное преобразова-

ние, каждое значение отображается на соответствующий ему промежуток множества допустимых значений управления $[a_j, b_j]$: $P_{ij}^p = [b_j - a_j]P_{ij}^{M \cdot p} + a_j$. Составляя векторы из элементов последовательности $\{P_{ij}^p\}$, $i=1, \dots, N$; $j=1, \dots, q$ при фиксированном значении p , получаем m начальных векторов $\tilde{u}^p = (\tilde{u}_1^p, \tilde{u}_2^p, \dots, \tilde{u}_{Nq}^p)^T$, $\tilde{u}_i^p = P_{ij}^p$, $i=1, 2, \dots, N$, $j=1, \dots, q$. Таким образом может быть сформирована начальная популяция $Y_M = \{\tilde{u}^p, p=1, 2, \dots, m \mid \tilde{u}^p = (\tilde{u}_1^p, \tilde{u}_2^p, \dots, \tilde{u}_{Nq}^p)^T \in \tilde{U}\}$.

Шаг 2.2. Вычислить значение функции приспособленности для каждой особи $\tilde{u}^p \in Y_M$: $\tilde{I}_p = \tilde{I}(\tilde{u}^p)$, $p=1, \dots, m$. В соответствии со значениями функции приспособленности \tilde{I}_p упорядочить решения \tilde{u}^p от лучшего (обеспечивающего наименьшее значение \tilde{I}_p) к худшему (обеспечивающему наибольшее значение \tilde{I}_p).

Шаг 2.3. Лучшее решение \tilde{u}^{best} поместить в множество $Pool$. Положить $\tilde{u}_{pool}^{k+1} = \tilde{u}^{best}$. Если $M = 0$, то перейти к шагу 2.4. Иначе перейти к шагу 3.

Шаг 2.4. Решение \tilde{u}^j , следующее за \tilde{u}^{best} в упорядоченном списке и удовлетворяющее условию $d(\tilde{u}^{best}, \tilde{u}^j) = \sqrt{\sum_{i=1}^{Nq} (\tilde{u}_i^{best} - \tilde{u}_i^j)^2} > \sigma$, поместить в множество $Pool$. Положить $\tilde{u}_{pool}^{k+2} = \tilde{u}^j$.

Положить $k = k + 2$. Если такого решения нет, то перейти к шагу 2.1.

Шаг 3. Решение задачи локального поиска.

Шаг 3.1. Решить задачу $\tilde{I}(\tilde{u}^{new}) \rightarrow \min_{c_1, c_2, \dots, c_k}$, где $\tilde{u}^{new} = c_1 \tilde{u}_{pool}^1 + c_2 \tilde{u}_{pool}^2 + \dots + c_k \tilde{u}_{pool}^k$; $\tilde{u}^{new} \in \tilde{U}$; $c_1, \dots, c_k \in C$; $\tilde{u}_{pool}^1, \tilde{u}_{pool}^2, \dots, \tilde{u}_{pool}^k \in Pool$, $C_1, \dots, C_k \in C$. Для решения этой задачи применить метод муравьиных колоний или метод имитации отжига.

Шаг 3.2. Поместить \tilde{u}^{new} в множество $Pool$. Положить $\tilde{u}_{pool}^{k+1} = \tilde{u}^{new}$, $k = k + 1$.

Шаг 3.3. Если $k < K$, то перейти к шагу 3.1. Если $k = K$, перейти к шагу 4.

Шаг 4. Процедура перекомпоновки пути (path-relinking).

Шаг 4.1. Упорядочить решения, находящиеся в множестве $Pool$ в соответствии со значениями функции приспособленности $\tilde{I}(\tilde{u}_{pool}^j)$, $j=1, \dots, K$, от лучшего к худшему.

Шаг 4.2. Генерация подмножеств. Создать $T = \sum_{j=1}^{K-2} j$ подмножеств типа $\{\tilde{u}_{pool}^i, \tilde{u}_{pool}^j, \tilde{u}_{pool}^{j+1}\}$, где $i=1, \dots, K-2$, $j > i$.

Шаг 4.3. Для каждого из созданных на шаге 4.2 подмножеств $\{\tilde{u}_{pool}^i, \tilde{u}_{pool}^j, \tilde{u}_{pool}^{j+1}\}$ выполнить:

Шаг 4.3.1. Сгенерировать P решений $\{u_{pr}^1, \dots, u_{pr}^P\}$, где

$$u_{pr}^j = \tilde{u}_{pool}^i + \frac{j}{1+P} (\tilde{u}_{pool}^j - \tilde{u}_{pool}^i), \quad j=1, \dots, P.$$

Шаг 4.3.2. Из решений $\{u_{pr}^1, \dots, u_{pr}^P\}$ выбрать решение u_{pr}^* , которое соответствует наименьшему значению среди $\tilde{I}(u_{pr}^j)$, $j=1, \dots, P$.

Шаг 4.3.3. Сгенерировать P решений $\{\tilde{u}_{pr}^1, \dots, \tilde{u}_{pr}^P\}$, где

$$\tilde{u}_{pr}^j = u_{pr}^* + \frac{j}{1+P} (\tilde{u}_{pool}^{j+1} - u_{pr}^*), \quad j=1, \dots, P.$$

Шаг 4.3.4. Из решений $\{\tilde{u}_{pr}^1, \dots, \tilde{u}_{pr}^P\}$ выбрать решение \tilde{u}_{pr}^* , которое соответствует наименьшему значению из $\tilde{I}(\tilde{u}_{pr}^j), j = 1, \dots, P$.

Шаг 4.3.5. Добавить решение \tilde{u}_{pr}^* в множество $Pool$, положить $k = k + 1$.

Шаг 5. Процедура интенсивно-уточняющего поиска по координатам. Для каждого решения $\tilde{u}^j = (\tilde{u}_1^j, \dots, \tilde{u}_i^j, \dots, \tilde{u}_{Nq}^j)^T$ из множества $Pool$ выполнить следующие шаги.

Шаг 5.1. Задать номер цикла $w = 1$ и величину смещения $\Delta = (IM + 1) \cdot \sigma$.

Шаг 5.2. Задать номер исследуемой координаты $y = 1$.

Шаг 5.3. Найти решения $\tilde{u}_-^j = (\tilde{u}_1^j, \dots, \tilde{u}_y^j - \Delta, \dots, \tilde{u}_{Nq}^j)^T$ и $\tilde{u}_+^j = (\tilde{u}_1^j, \dots, \tilde{u}_y^j + \Delta, \dots, \tilde{u}_{Nq}^j)^T$.

Шаг 5.4. Если $\tilde{I}(\tilde{u}_-^j) < \tilde{I}(\tilde{u}^j)$, то заменить \tilde{u}^j на \tilde{u}_-^j и перейти к шагу 5.3; если $\tilde{I}(\tilde{u}_-^j) \geq \tilde{I}(\tilde{u}^j)$, то перейти к шагу 5.5.

Шаг 5.5. Если $\tilde{I}(\tilde{u}_+^j) < \tilde{I}(\tilde{u}^j)$, то заменить \tilde{u}^j на \tilde{u}_+^j и перейти к шагу 5.3; если $\tilde{I}(\tilde{u}_+^j) \geq \tilde{I}(\tilde{u}^j)$, то перейти к шагу 5.6.

Шаг 5.6. Увеличить y на единицу. Если $y > Nq$, то увеличить w на единицу и перейти к шагу 5.7. В противном случае перейти к шагу 5.3.

Шаг 5.7. Если $w > IM$, то перейти к шагу 6. В противном случае положить $\Delta = \Delta - \sigma$ и перейти к шагу 5.2.

Шаг 6. Обновление множества $Pool$.

Шаг 6.1. Упорядочить решения, находящиеся в множестве $Pool$, в соответствии со значениями функции приспособленности $\tilde{I}(\tilde{u}_{pool}^j), j = 1, \dots, k$ от лучшего к худшему. Наилучшее решение записать на лист памяти.

Шаг 6.2. Положить $M = M + 1$. Если $M < M_{\max}$, то перейти к шагу 6.3. Иначе выбрать наилучшее решение из листа памяти и закончить выполнение алгоритма.

Шаг 6.3. Удалить $Q + T$ наихудших решений из $Pool$, положить $k = k - Q - T$.

Шаг 6.4. Удалить из множества $Pool$ худшее из двух решений $\tilde{u}_{pool}^{j_1}, \tilde{u}_{pool}^{j_2}$, находящихся слишком близко друг к другу, т.е. не удовлетворяющих условию:

$$d(\tilde{u}_{pool}^{j_1}, \tilde{u}_{pool}^{j_2}) = \sqrt{\sum_{i=1}^{Nq} (\tilde{u}_{pooli}^{j_1} - \tilde{u}_{pooli}^{j_2})^2} > \sigma, \forall \tilde{u}_{pool}^{j_1}, \tilde{u}_{pool}^{j_2}, \text{ каждый раз полагая } k = k - 1. \text{ Перейти к шагу 2.}$$

4. Программное обеспечение

На основе предложенного алгоритма разработан комплекс программ для поиска оптимального управления непрерывными детерминированными системами. Среда разработки – Microsoft Visual Studio, язык программирования – C#. С помощью программного обеспечения пользователь может вводить параметры постановки задачи, а также задавать параметры меметического алгоритма и метода решения задачи локального поиска. В программе имеется возможность выбора метода решения задачи локального поиска: муравьиных колоний (непрерывный вариант) либо имитации отжига (рис. 1).

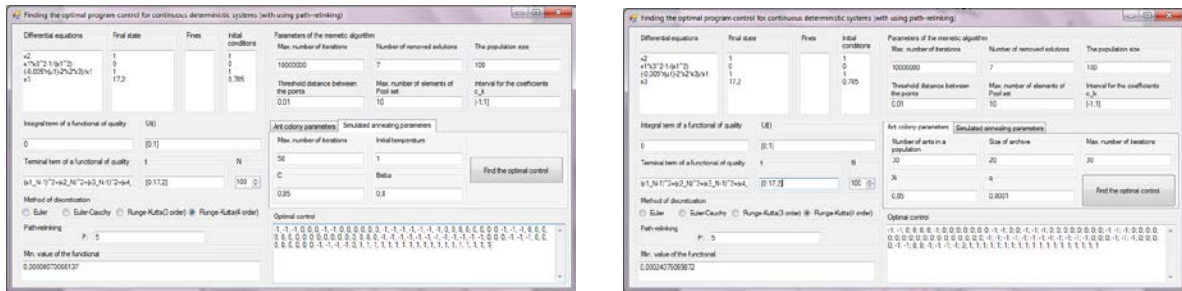


Рис. 1. Главное окно комплекса программ

5. Решение задачи о переводе спутника между орбитами

Пусть движение спутника описывается системой дифференциальных уравнений [9]:

$$\begin{aligned} \dot{x}_1 &= x_2, & \dot{x}_2 &= x_1 \cdot x_3^2 - \frac{1}{x_1^2}, \\ \dot{x}_3 &= \frac{-u - 2 \cdot x_2 \cdot x_3}{x_1}, & \dot{x}_4 &= x_3, \end{aligned}$$

где $x_1 = r$ – радиус-вектор от притягивающего центра; $x_2 = \dot{r}$ – скорость изменения радиус-вектора; $x_3 = \dot{\varphi}$ – скорость изменения угла поворота радиус-вектора; $x_4 = \varphi$ – угол поворота радиус-вектора.

Начальное состояние системы - $x(0) = (1; 0; 1; 0,785)^T$, конечное состояние - $x(17,2) = (1; 0; 1; 17,2)^T = x^*$, $t_N = 17,2$. Требуется перевести спутник из заданного начального состояния в заданное конечное при помощи выбора управления, удовлетворяющего ограничению $\forall t \ u(t) \in \{-0,005; 0; 0,005\}$, описывающему трехпозиционное реле.

Функционал качества управления терминальный $I = \sum_{k=1}^4 (x_k(t_N) - x_k^*)^2 \rightarrow \min$.

Для обеспечения возможности применения гибридного меметического алгоритма преобразуем систему дифференциальных уравнений к виду

$$\begin{cases} \dot{x}_1 = x_2, \\ \dot{x}_2 = x_1 \cdot x_3^2 - \frac{1}{x_1^2}, \\ \dot{x}_3 = \frac{-0,005 \cdot r(u) - 2 \cdot x_2 \cdot x_3}{x_1}, \\ \dot{x}_4 = x_3, \end{cases}$$

где $r(u)$ – релейная функция, определяемая следующим образом

$$r(u) = \begin{cases} -1, & 0 \leq u < \frac{1}{3}, \\ 0, & \frac{1}{3} \leq u < \frac{2}{3}, \\ 1, & \frac{2}{3} \leq u \leq 1. \end{cases}$$

В полученной задаче ограничение на управление выглядит следующим образом: $\forall t: u(t) \in [0;1]$. Управление и траектории системы, найденные в [9], изображены на рис. 2; величина функционала составляет 0,00024.

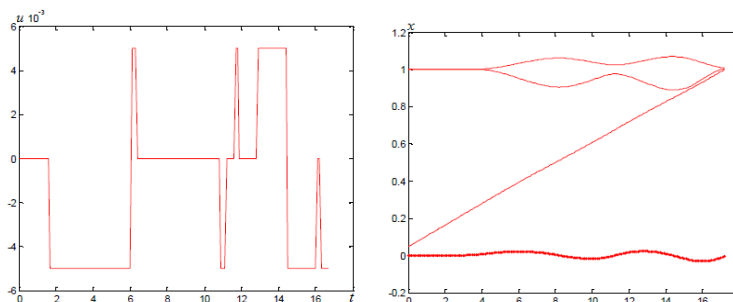


Рис. 2. Результаты решения задачи, полученные в [9]

Решение, полученное меметическим алгоритмом, представлено на рис. 3. Для нахождения решения использовались следующие параметры меметического алгоритма: максимальное число итераций $M_{\max} = 10^7$; размер популяции m ; порог расстояния между точками σ ; максимальное количество элементов множества **Pool K**; количество удаляемых решений из множества **Pool Q**; область определения коэффициентов c_1, \dots, c_k $C = [-1;1]$, $IM = 10$, $P = 5$.

Параметры метода имитации отжига: максимальное количество итераций $N = 50$, начальная температура $T_0 = 1$; параметр закона распределения Больцмана $C = 0,85$; параметр уменьшения температуры $\beta = 0,8$.

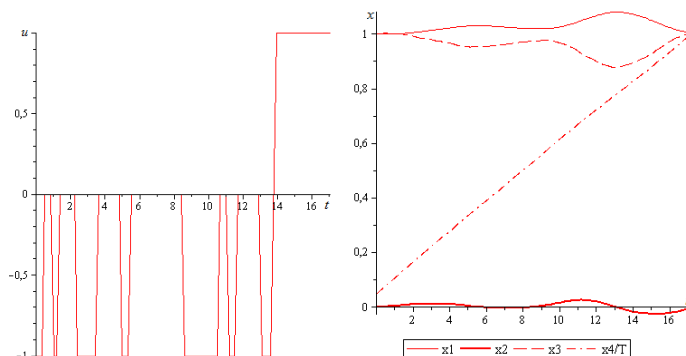


Рис. 3. Результаты решения задачи, полученные гибридным меметическим алгоритмом

Метод дискретизации – метод Эйлера, количество шагов дискретизации - 100, величина функционала $I = 0,000081$, значения координат в конечный момент: $x(17,2) = (1,005; -0,005; 1,002; 17,206)^T$.

Заключение

В работе предложен гибридный меметический алгоритм для решения задачи поиска оптимального программного управления нелинейными непрерывными детерминированными системами. Реализован комплекс программ, позволяющий применять разработанный алгоритм для различных систем. Точность решения может быть скорректирована путем подбора параметров метода. Работа алгоритма продемонстрирована на задаче перевода спутника с одной орбиты на другую. Сравнение результатов работы меметического алгоритма с результатами, полученными с помощью других методов, иллюстрирует эффективность применения разработанного метода.

ЛИТЕРАТУРА

1. **Dawkins R.** Universal Darwinism in D.S. Bendall (ed.), Evolution: From Molecules to Men, Cambridge University Press, Cambridge, 1983. - P. 403-425.
2. **Dawkins R.** The Selfish Gene. - Oxford University Press, 1976.
3. **Moscato P.** On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms // Caltech Concurrent Computation Program (report 826), 1989.
4. **Neri F., Cotta C., Moscato P.** Handbook of Memetic Algorithms // Studies in Computational Intelligence, V. 379, Springer-Verlag, Berlin Heidelberg, 2012.
5. **Dorigo M., Socha K.** Ant colony optimization for continuous domains // European Journal of Operational Research. V. 185, 2008. - P. 1155-1173.
6. **Kirkpatrick S., Gelatt C.D., Vecchi M.P.** Optimization by simulated annealing. Science. V. 220, 1983. - P. 671-680.
7. **Кирпеев В.И., Пантелеев А.В.** Численные методы в примерах и задачах. - М.: Высшая школа, 2008.
8. **Duarte A., Marti R., Gortazar F.** Path relinking for large scale global optimization. Soft Comput. 15(11), 2001. - P. 2257-2273.
9. **Охорзин В.А., Рыжиков И.С.** Решение терминальной задачи управления в виде многоуровневого реле // Перспективные инновации в науке, образовании, производстве и транспорте 2012: материалы междунар. науч.-практич. конф. - Вып. 2. - 2012. - Т. 2. - С. 89-93.
10. **Пантелеев А.В., Метлицкая Д.В., Алешина Е.А.** Методы глобальной оптимизации. Метаэвристические стратегии и алгоритмы. - М.: Вузовская книга, 2013.

**SOLUTION OF THE SATELLITE TRANSFER PROBLEM
WITH HYBRID MEMETIC ALGORITHM****Panteleyev A.V., Pismennaya V.A.**

This paper presents a hybrid memetic algorithm (MA) to solve the problem of finding the optimal program control of nonlinear continuous deterministic systems based on the concept of the meme, which is one of the promising solutions obtained in the course of implementing the procedure for searching the extremes. On the basis of the proposed algorithm the software complex is formed in C#. The solution of satellite transfer problem is presented.

Key words: optimal control, global extremes, nonlinear continuous deterministic systems, memetic algorithm, satellite transfer problem.

Сведения об авторах

Пантелеев Андрей Владимирович, 1955 г.р., окончил МГТУ им. Н.Э. Баумана (1978), профессор, доктор физико-математических наук, заведующий кафедрой математической кибернетики МАИ (Национального исследовательского университета), автор более 170 научных работ, область научных интересов – методы синтеза оптимальных нелинейных систем управления, методы оптимизации.

Письменная Виктория Александровна, окончила МАИ (2013), аспирантка МАИ, область научных интересов – методы оптимизации, численные методы, эвристические алгоритмы.