

УДК 517.97

## **Генетические алгоритмы поиска оптимального управления непрерывными детерминированными системами**

Д.В. Метлицкая

Научный руководитель доктор физико-математических наук,  
профессор А.В. Пантелеев

### **Аннотация**

Предложена методика применения генетических алгоритмов поиска условного глобального экстремума с бинарным и вещественным кодированием для решения задачи оптимального управления нелинейными непрерывными детерминированными системами. Сформирован детальный алгоритм решения поставленной задачи, на основе которого создано соответствующее программное обеспечение. Приведен пример решения прикладной задачи и проведено сравнение полученных результатов с другими методами, иллюстрирующее эффективность сформированных алгоритмов.

### **Ключевые слова:**

генетические алгоритмы; бинарное кодирование; вещественное кодирование; непрерывная система управления; оптимальное управление.

### **Введение**

На различных этапах проектирования сложных авиационно-космических комплексов требуется решение различных задач оптимального управления динамическими объектами, поведение которых описывается обыкновенными дифференциальными уравнениями. Применение для их решения численных методов, реализующих соответствующие

необходимые или достаточные условия оптимальности, нередко связано с большими вычислительными затратами, трудностями в достижении сходимости процесса, неэффективностью алгоритмов при увеличении размерности вектора состояния объекта. Поэтому актуальной проблемой является развитие метаэвристических методов [1], которые позволяют получать приемлемые с точки зрения проектировщика решения за разумное время.

В данной работе рассмотрено применение генетических алгоритмов (ГА) поиска условного глобального экстремума с бинарным и вещественным кодированием [1-3] к задаче нахождения оптимального программного управления непрерывными детерминированными системами [4]. Предложен детальный алгоритм решения поставленной задачи, на основе которого сформирован комплекс программных средств. Комплекс имеет удобный интерфейс, с помощью которого можно легко менять параметры алгоритма, а также анализировать эффективность его работы. Решена прикладная задача, на которой продемонстрировано, что предложенные в данной работе ГА работают не хуже эволюционных алгоритмов, описанных в [5].

### Постановка задачи

Поведение модели объекта управления описывается дифференциальным уравнением  $\dot{x} = f(t, x(t), u(t))$ ,  $t \in [t_0, t_N]$ , 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;  $f(t, x, u) = (f_1(t, x, u), \dots, f_n(t, x, u))^T$  – непрерывная вектор-функция.

Начальное состояние системы задано:  $x(t_0) = x_0$ . Правый конец траектории свободен.

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 управление  $u(t)$ ,  $u(t) \in U$ , удовлетворяющих дифференциальному уравнению и начальному условию  $x(t_0) = x_0$ .

На множестве допустимых процессов определен функционал качества управления  $I(d) = F(x(t_N))$ , где  $F(x)$  – заданная непрерывная функция.

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

### Стратегия поиска решения

Основная идея ГА состоит в том, что они имитируют в своей работе природные способы оптимизации: генетическое наследование и естественный отбор. При этом в упрощенном виде сохраняется биологическая терминология. Суть механизма естественного отбора заключается в том, что в процессе эволюции выживают и размножаются наиболее приспособленные особи. Благодаря механизму генетического наследования их потомки сохраняют основные качества родителей, и, подвергаясь случайным мутациям, приобретают ряд новых свойств. Если новые свойства полезны, то они сохраняются и наследуются.

Классические ГА применимы к решению задачи оптимизации целевой функции  $f(x) = f(x_1, x_2, \dots, x_n)$ , определенной на множестве допустимых решений  $D \subseteq R^n$ , и позволяют найти ее условный глобальный максимум на заданном множестве, т.е. такую точку  $x^* \in D$ , что  $f(x^*) = \max_{x \in D} f(x)$ . Задача поиска минимума функции  $f(x)$  сводится к задаче поиска максимума путем замены знака перед функцией на противоположный:  $f(x^*) = \min_{x \in D} f(x) = -\max_{x \in D} [-f(x)]$ .

Рассматриваемая целевая функция  $f(x)$  эквивалентна понятию приспособленности живого организма. Вектор параметров  $x = (x_1, x_2, \dots, x_n)^T$  целевой функции называется фенотипом, а отдельные его параметры  $x_i$  – признаками,  $i = 1, 2, \dots, n$ .

В ГА каждую координату  $x_i$  вектора  $x = (x_1, x_2, \dots, x_n)^T \in D$  представляют в некоторой форме  $s_i$ , называемой геном. Для этого вектор параметров  $x = (x_1, x_2, \dots, x_n)^T \in D$  преобразуют в некоторую структуру  $s = (s_1, s_2, \dots, s_n)^T \in S$ , называемую хромосомой:  $D \xrightarrow{e} S$ , где  $e$  – функция кодирования,  $S$  – пространство представлений. Восстанавливает решение обратное преобразование:  $S \xrightarrow{e^{-1}} D$ , где  $e^{-1}$  – функция декодирования. В пространстве  $S$  вводится функция приспособленности  $\mu(s)$ :  $S \xrightarrow{\mu} R$ , аналогичная целевой функции  $f(x)$  на множестве  $D$  и удовлетворяющая

условию:  $\forall x^1, x^2 \in D : s^1 = e(x^1), s^2 = e(x^2), s^1 \neq s^2$ , если  $f(x^1) > f(x^2)$ , то  $\mu(s^1) > \mu(s^2)$ .

Таким образом, решение исходной оптимизационной задачи  $f(x^*) = \max_{x \in D} f(x)$  сводится к поиску решения  $s^*$  другой задачи оптимизации:  $\mu(s^*) = \max_{s \in S} \mu(s)$ . Затем осуществляется обратное преобразование:  $x^* = e^{-1}(s^*)$ .

**Операции кодирования и декодирования.** При бинарном кодировании в ГА наиболее часто применяются битовые строки. Каждой координате  $x_i \in [a_i, b_i]$  вектора  $x = (x_1, x_2, \dots, x_n)^T \in D$  ставится в соответствие ген – битовая строка  $s_i$  длины  $l_i$ . Разобьем интервал  $[a_i, b_i]$  на  $k_i - 1$  подынтервалов:  $h_i = \frac{b_i - a_i}{k_i - 1}$ . Координате  $x_i$  ставится в соответствие целое число  $\beta_i \in \{0, 1, \dots, k_i\}$ :

$$\beta_i = \begin{cases} 0, & x_i = a_i, \\ \left[ \frac{x_i - a_i}{h_i} \right] + 1, & x_i \in (a_i, b_i), \\ k_i, & x_i = b_i. \end{cases}$$

Далее каждому номеру  $\{0, 1, \dots, k_i\}$  ставится в соответствие уникальная бинарная комбинация длины  $q_i$ , которая определяется из условия:  $k_i = 2^{q_i} - 1$ . Преобразование в битовую строку осуществляется с помощью рефлексивного кода Грея.

В процессе декодирования ген  $s_i$  переводится в десятичное представление  $\gamma_i$ . Значение координаты  $x_i$  определяется следующим образом:

$$x_i = \begin{cases} a_i, & \gamma_i = 0, \\ \frac{(a_i + (\gamma_i - 1)h_i) + (a_i + \gamma_i h_i)}{2}, & \gamma_i = 1, 2, \dots, k_i - 1, \\ b_i, & \gamma_i = k_i. \end{cases}$$

В ГА с вещественным кодированием функцией приспособленности является целевая функция  $f(x)$ , а хромосомой – вектор  $x = (x_1, x_2, \dots, x_n)^T \in D$ . Таким образом, хромосома в ГА с вещественным кодированием – это набор вещественных чисел. Каждая координата  $x_i$  вектора  $x$  является геном. Операции кодирования и декодирования в ГА с вещественным кодированием отсутствуют. Далее, для удобства изложения, хромосому в ГА с

вещественным кодированием  $x = (x_1, x_2, \dots, x_n)^T$ , пространство, которому она принадлежит  $D$ , и функцию приспособленности  $f(x)$  будем обозначать так же, как и в ГА с бинарным кодированием:  $s = (s_1, s_2, \dots, s_n)^T$ ,  $S$  и  $\mu(s)$ , соответственно.

При решении задачи используются наборы  $I = \{s^k = (s_1^k, s_2^k, \dots, s_n^k)^T, k = 1, 2, \dots, m\} \subset S$  возможных решений, называемые популяциями, где  $s^k$  – хромосома с номером  $k$ ,  $m$  – размер популяции,  $s_i^k$  – ген с номером  $i$ . Применение ГА сводится к исследованию множества  $S$ . Чем больше значение  $\mu(s^k)$ , тем более подходит хромосома  $s^k$  в качестве решения.

ГА имитирует эволюцию начальной популяции как циклический итерационный процесс, во время которого к популяции применяются основные биологические операторы: *селекция, скрещивание, мутация*. В результате происходит ее смена на новую, к которой, если решение не найдено, опять применяются биологические операторы, и так далее, до выполнения условия окончания работы алгоритма. Если условие окончания выполнено, то в качестве приближенного решения задачи  $\mu(s^*) = \max_{s \in S} \mu(s)$  из последней популяции выбирается особь с наибольшим значением функции приспособленности, а затем находится приближенное решение исходной задачи  $f(x^*) = \max_{x \in D} f(x)$ . В ГА с бинарным кодированием для этого необходимо применить операцию декодирования.

Для решения поставленной 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  $\{0, h, 2h, \dots, Nh\}$ . Непрерывному управлению  $u(t)$  будет соответствовать дискретное управление  $u(\cdot) = \{u(0), u(1), \dots, u(N-1)\}$ , где  $u(\mathcal{E}) = u(\mathcal{E}h)$ ,  $\mathcal{E} = 0, 1, \dots, N-1$ , а траектории  $x(t)$  - дискретный аналог:  $x(\cdot) = \{x(0), x(1), \dots, x(N)\}$ , где  $x(\mathcal{E}) = x(\mathcal{E}h)$ ,  $\mathcal{E} = 0, 1, \dots, N$ ;  $x(0) = x_0$ . Дифференциальные уравнения модели объекта управления будем приближенно заменять разностными с помощью численных методов [6],

формулы для которых приведены в табл.1. В результате получим задачу поиска оптимального программного управления дискретной детерминированной системой [4].

Таблица 1

Название метода	Расчетные формулы
Метод Эйлера	$x(\mathcal{E}+1) = x(\mathcal{E}) + \Delta x(\mathcal{E}) = x(\mathcal{E}) + h f(\mathcal{E}, x(\mathcal{E}), u(\mathcal{E}))$
Метод Эйлера-Коши	$\tilde{x}(\mathcal{E}+1) = x(\mathcal{E}) + h f(\mathcal{E}, x(\mathcal{E}), u(\mathcal{E})),$ $x(\mathcal{E}+1) = x(\mathcal{E}) + \Delta x(\mathcal{E}) = x(\mathcal{E}) + h \frac{(f(\mathcal{E}, x(\mathcal{E}), u(\mathcal{E})) + f((\mathcal{E}+1)h, \tilde{x}(\mathcal{E}+1), u(\mathcal{E})))}{2}$
Метод Рунге-Кутты 3-го порядка	$x(\mathcal{E}+1) = x(\mathcal{E}) + \Delta x(\mathcal{E}), \quad \Delta x(\mathcal{E}) = \frac{1}{4}(K_1^\mathcal{E} + 3K_3^\mathcal{E}),$ $K_1^\mathcal{E} = h f(\mathcal{E}, x(\mathcal{E}), u(\mathcal{E})),$ $K_2^\mathcal{E} = h f(\mathcal{E} + \frac{1}{3}h, x(\mathcal{E}) + \frac{1}{3}K_1^\mathcal{E}, u(\mathcal{E})),$ $K_3^\mathcal{E} = h f(\mathcal{E} + \frac{2}{3}h, x(\mathcal{E}) + \frac{2}{3}K_2^\mathcal{E}, u(\mathcal{E}))$
Метод Рунге-Кутты 4-го порядка	$x(\mathcal{E}+1) = x(\mathcal{E}) + \Delta x(\mathcal{E}),$ $\Delta x(\mathcal{E}) = \frac{1}{6}(K_1^\mathcal{E} + 2K_2^\mathcal{E} + 2K_3^\mathcal{E} + K_4^\mathcal{E}),$ $K_1^\mathcal{E} = h f(\mathcal{E}, x(\mathcal{E}), u(\mathcal{E})),$ $K_2^\mathcal{E} = h f(\mathcal{E} + \frac{1}{2}h, x(\mathcal{E}) + \frac{1}{2}K_1^\mathcal{E}, u(\mathcal{E})),$ $K_3^\mathcal{E} = h f(\mathcal{E} + \frac{1}{2}h, x(\mathcal{E}) + \frac{1}{2}K_2^\mathcal{E}, u(\mathcal{E})),$ $K_4^\mathcal{E} = h f(\mathcal{E} + h, x(\mathcal{E}) + K_3^\mathcal{E}, u(\mathcal{E}))$

При использовании ГА для решения поставленной задачи будем оптимизировать управление  $u(\cdot)$ . Целевой функцией, а также функцией приспособленности будет являться функционал  $I(d) = I(x(\cdot), u(\cdot))$ . Хромосома  $k$ -ой особи в популяции будет представлять собой вектор-строку  $u^k = (u^k(0), u^k(1), \dots, u^k(N-1))$ . Для того чтобы найти значение функционала  $I(d^k)$ , соответствующее паре  $d^k = (x^k, u^k)$ , необходимо вычислить траекторию системы  $x^k = (x(0), x^k(1), \dots, x^k(N))$ , соответствующую управлению  $u^k$ , из

уравнения дискретной системы с учетом начального условия. Формула для нахождения значения функционала имеет вид:  $I(d^k) = F(x^k(N))$ . Состояние  $x^k(\ell)$ ,  $\ell = 0, 1, \dots, N$ , определяется численным методом решения заданного дифференциального уравнения с начальным условием:  $x^k(\ell+1) = x^k(\ell) + \Delta x(\ell)$ ,  $\ell = 0, 1, \dots, N-1$ ,  $x^k(0) = x_0$ , где приращение  $\Delta x(\ell)$  определяется соотношениями выбранного численного метода (табл. 1).

В случае, когда необходимо найти пару  $d^* = (x^*(t), u^*(t))$ , соответствующую минимальному значению функционала, будет решаться задача  $I(d^*) = \min_{d \in D(t_0, x_0)} I(d) = - \max_{d \in D(t_0, x_0)} [-I(d)]$ . Функцией приспособленности будет являться функционал  $-I(d) = -I(x(\cdot), u(\cdot))$ . Вид хромосомы и способ нахождения значения функционала  $-I(d^k)$ , соответствующего паре  $d^k = (x^k, u^k)$ , аналогичны предыдущему случаю.

## Методика применения генетических алгоритмов

**Шаг 1.** Создание начальной популяции.

1. Задается номер популяции  $p=0$ , максимальное количество популяций  $p_{\max}$ , номер итерации  $j=1$ , число шагов  $N$ , размер популяции  $m$ , длина битовой строки  $r$ .

2. Случайным образом, используя равномерное распределение на множестве допустимых значений управления, формируется начальная популяция  $I_0$ . В результате получаем  $m$  начальных вектор-строк

$$u^k = (u_{1,1}^k, \dots, u_{1,q}^k, u_{2,1}^k, \dots, u_{2,q}^k, \dots, u_{N,1}^k, \dots, u_{N,q}^k) = (u_1^k, u_2^k, \dots, u_n^k),$$

где  $n = N \cdot q$ ,  $k = 1, 2, \dots, m$  (заметим, что здесь  $u_{i,j}^k = u_j^k(i-1)$ ).

В ГА с бинарным кодированием с помощью операции кодирования формируется  $m$  хромосом вида  $s^k = (s_{1,1}^k, \dots, s_{1,q}^k, s_{2,1}^k, \dots, s_{2,q}^k, \dots, s_{N,1}^k, \dots, s_{N,q}^k) = (s_1^k, s_2^k, \dots, s_n^k)$ ,  $n = N \cdot q$ . В результате описанного процесса сформирована начальная популяция  $I_0 = \{s^k = (s_1^k, s_2^k, \dots, s_n^k), k = 1, 2, \dots, m\} \subset S$ .

В ГА с вещественным кодированием операция кодирования не требуется. Как и ранее, в этом случае будем обозначать хромосому и функцию приспособленности таким же образом, как и в ГА с бинарным кодированием.

3. Вычисляется значение функции приспособленности для каждой особи  $s^k \in I_0$ :  $\mu_k = \mu(s^k)$ ,  $k = 1, 2, \dots, m$ . Для этого в ГА с бинарным кодированием необходимо с помощью

операции декодирования найти вектор-строку  $u^k$ , соответствующую хромосоме  $s^k$  и вычислить траекторию  $x^k$ . В ГА с вещественным кодированием операция декодирования не требуется. Также вычисляется приспособленность популяции  $I_0$  в целом:  $\mu = \sum_{k=1}^m \mu_k$ .

**Шаг 2. Селекция** – это операция, которая осуществляет отбор особей (хромосом)  $s^k$  в соответствии со значениями функции приспособленности  $\mu(s^k)$  для последующего их скрещивания.

В ГА с бинарным кодированием используются следующие операторы селекции:

1) «панмиксия» - случайный равновероятностный отбор, вероятность участия особи в селекции  $p_k = \frac{1}{m}$ ;

2) селективный отбор – «родителями» могут стать с равной вероятностью только те особи, для которых выполнено условие  $\mu(s^k) \geq \frac{\mu}{m}$ ;

3) рулетка - отбирает особей с помощью двух «запусков» рулетки, вероятность участия особи в селекции пропорциональна относительной приспособленности особи  $p_k = \frac{\mu(s^k)}{\mu}$ ;

4) турнирный отбор - реализует два турнира, в каждом из которых случайным образом выбирается два элемента из популяции, затем выбирается лучшая особь среди них.

В ГА с вещественным кодированием используются следующие операторы селекции:

1) «панмиксия» - случайный равновероятностный отбор, вероятность участия особи в селекции  $p_k = \frac{1}{m}$ ;

2) рулетка - отбирает особей с помощью двух «запусков» рулетки, вероятность участия особи в селекции пропорциональна относительной приспособленности особи  $p_k = \frac{I_k}{I}$ ;

3) турнирный отбор - реализует два турнира, в каждом из которых случайным образом выбирается два элемента из популяции, затем выбирается лучшая особь среди них.

В ГА может применяться *стратегия элитизма*. Ее суть в том, что небольшое количество особей  $s^k$  переходит в следующее поколение без изменений, не участвуя в селекции и последующем скрещивании.



Результатом шага 2 являются две особи  $s^1$  и  $s^2$ , выбранные в качестве родительской пары с помощью одного из операторов селекции.

**Шаг 3. Скрещивание** – это операция, при которой из нескольких, обычно двух хромосом (особей), называемых родителями, порождается одна или несколько новых, называемых потомками, путем обмена частями родительских хромосом. В простейшем случае скрещивание в ГА реализуется так же, как и в биологии: хромосомы разрезаются в случайной точке (точках) и обмениваются частями между собой.

В ГА с бинарным кодированием применяются следующие операторы скрещивания:

1) одноточечное скрещивание - случайным образом выбирается точка разрыва хромосомы, обе родительские структуры разрываются на два сегмента по этой точке и меняются соответствующими сегментами;

2) двухточечное скрещивание – выбираются две точки разрыва, и родительские хромосомы обмениваются сегментом, находящимся между этими точками;

3) равномерное скрещивание – первый потомок наследует каждый бит первого, либо второго родителя с равной вероятностью, при этом бит, не унаследованный первым потомком, наследуется вторым.

В ГА с вещественным кодированием применяются следующие операторы скрещивания:

1) плоский кроссовер - создается один потомок  $s^q = (s_1^q, \dots, s_n^q)^T$ , где  $s_i^q, i = 1, \dots, n$  – случайное число из промежутка  $[s_i^1, s_i^2]$ ,  $s_i^2 \geq s_i^1$  или из промежутка  $[s_i^2, s_i^1]$ ,  $s_i^2 < s_i^1$ ;

2) простейший кроссовер - из множества  $\{1, 2, \dots, n-1\}$  выбирается случайное число  $p$ , и генерируются два потомка:  $s^{q1} = (s_1^1, \dots, s_p^1, s_{p+1}^2, \dots, s_n^2)^T$  и  $s^{q2} = (s_1^2, \dots, s_p^2, s_{p+1}^1, \dots, s_n^1)^T$ ;

3) арифметический кроссовер - создаются два потомка  $s^{q1} = (s_1^{q1}, \dots, s_n^{q1})^T$ ,  $s^{q2} = (s_1^{q2}, \dots, s_n^{q2})^T$ , где  $s_i^{q1} = \eta s_i^1 + (1-\eta) s_i^2$ ,  $s_i^{q2} = \eta s_i^2 + (1-\eta) s_i^1$ ,  $i = 1, 2, \dots, n$ , где  $\eta \in (0, 1)$ ;

4) дискретный кроссовер - создается один потомок  $s^q = (s_1^q, \dots, s_n^q)^T$ , где  $s_i^q$  выбирается из двух значений  $s_i^1, s_i^2$  случайно с вероятностью 0,5 для всех  $i = 1, \dots, n$ ;

5) кроссовер, имитирующий двоичный - создаются два потомка, в которых

$$u_i^{q1} = \frac{1}{2}[(1-\eta^1) u_i^1 + (1+\eta^1) u_i^2], \quad u_i^{q2} = \frac{1}{2}[(1+\eta^2) u_i^1 + (1-\eta^2) u_i^2],$$

$$\text{где } \eta^j = \begin{cases} (2z)^{\frac{1}{C+1}}, & z \leq 0,5, \\ \left(\frac{1}{2(1-z)}\right)^{\frac{1}{C+1}}, & z > 0,5, \end{cases} \quad j=1,2; \quad z - \text{случайное число, равномерно}$$

распределенное на  $(0; 1)$ ,  $C \in [2; 5]$  – параметр оператора.

Результатом шага 3 являются два потомка  $s^{ch1}$  и  $s^{ch2}$ , полученные путем применения одного из операторов скрещивания к хромосомам - родителям  $s^1$  и  $s^2$ .

**Шаг 4. Мутация** – это преобразование хромосомы, случайно изменяющее обычно один (реже несколько) из ее генов. Оператор мутации предназначен для того, чтобы поддерживать разнообразие особей в популяции.

В ГА с бинарным кодированием используются следующие операторы мутации:

1) обыкновенная мутация – каждый бит в хромосоме с определенной вероятностью  $p \in (0,1)$  инвертируется;

2) инверсия – хромосома делится на две части в случайной точке разрыва, полученные части меняются местами.

В ГА с вещественным кодированием применяются следующие операторы мутации:

1) случайная мутация. Поочередно рассматривается каждый потомок, полученный в результате скрещивания. Среди генов  $s_1, \dots, s_n$  случайно (с вероятностью  $\frac{1}{n}$ ) выбирается один с номером  $p \in \{1, \dots, n\}$ , подлежащий замене. Его новое значение  $u_p^M$  случайным образом выбирается из промежутка  $[a_p(t), b_p(t)]$  изменения выбранной координаты  $u_p$  при заданном времени  $t$ ;

2) неравномерная мутация – поочередно рассматривается каждый потомок, среди генов  $u_1, \dots, u_n$  случайно выбирается один, подлежащий замене:

$u_i^M = \begin{cases} u_i + \delta(p, (b_i(t) - u_i)), & z = 0, \\ u_i - \delta(p, (u_i - a_i(t))), & z = 1, \end{cases}$  где  $z$  – случайное число, выбранное с вероятностью 0,5 из множества  $\{0; 1\}$ ;  $\delta(p, y)$  – функция, определяющая значение на промежутке  $[0, y]$ :

$\delta(p, y) = y \left( 1 - r \left( 1 - \frac{p}{p_{\max}} \right)^C \right)$ ,  $r$  – случайное на отрезке  $[0, 1]$  число,  $p$  – номер текущей популяции,  $C$  – параметр оператора.

Результатом шага 4 являются два потомка - мутанта  $s^{M1}$  и  $s^{M2}$ , полученные путем применения оператора мутации или инверсии к хромосомам - потомкам  $s^{ch1}$  и  $s^{ch2}$ .

**Шаг 5. Формирование новой популяции.** С вероятностью 0,5 выбирается один из потомков  $s^{M1}$  и  $s^{M2}$ . Пусть это будет  $s^{M1}$ . Выбранный потомок добавляется в популяцию вместо хромосомы, которой соответствует наименьшее среди элементов популяции значение функции приспособленности, вычисляется значение функции приспособленности

$\mu_z = \mu(s^{M1})$  и приспособленность новой популяции в целом  $\mu = \sum_{k=1}^m \mu_k$ .

Проверяются условия: а) если  $j < m$ , то положить  $j = j + 1$  и перейти к шагу 2; б) если  $j = m$ , то положить  $p = p + 1$  и перейти к шагу 6.

**Шаг 6.** Проверка условия окончания работы ГА (сформировано заданное число популяций  $p = p_{\max}$ ): а) если условие не выполнено, то полагаем  $j = 1$  и переходим к шагу 2; б) если условие окончания работы выполнено, то в качестве решения (приближенного) задачи  $\mu(s^*) = \max_{s \in S} \mu(s)$  выбирается особь с лучшим значением функции приспособленности из текущей популяции:  $s^* \cong \mathfrak{s}^* = \operatorname{argmax}_{k=1, \dots, m} \mu(s^k)$ , а по нему в ГА с бинарным кодированием с помощью операции декодирования определяется приближенное решение поставленной задачи  $u^* \cong \tilde{u}^* = e^{-1}(\tilde{s}^*)$ .

Значение функционала (целевой функции) при этом будет равно  $I(d^*) = I(x^*, u^*)$ .

Более подробное описание типов операторов селекции, скрещивания и мутации в ГА с бинарным кодированием приведено в [1].

### Программное обеспечение

На основе изложенного алгоритма сформирован комплекс программных средств поиска оптимального программного управления дискретными детерминированными системами при помощи генетических алгоритмов условной оптимизации с бинарным и вещественным кодированием. Среда разработки Microsoft Visual Studio 2005, язык программирования C#.

Комплекс работает в режиме диалога с пользователем. Пользовательский интерфейс позволяет анализировать графическое изображение получаемых результатов. Работа программы начинается с выбора типа кодирования, ввода параметров алгоритма: характеристик популяции (размера, максимального количества особей, длины битовой строки), типа операторов селекции, скрещивания, мутации, количества элитных особей, а заканчивается предъявлением полученного приближенного решения: управления, соответствующей траектории и значения функционала качества.

**Пример.** Используя разработанное программное обеспечение, решим задачу оптимального управления химическим процессом, описанную в [5]. Модель нелинейной непрерывной детерминированной системы описывается дифференциальными уравнениями:

$$\dot{x}_1 = -(2+u)(x_1 + 0.25) + (x_2 + 0.5) \exp\left(\frac{25x_1}{x_1 + 2}\right);$$

$$\dot{x}_2 = 0.5 - x_2 - (x_2 + 0.5) \exp\left(\frac{25x_1}{x_1 + 2}\right);$$

$$\dot{x}_3 = x_1^2 + x_2^2 + 0.1u,$$

где  $0 \leq t \leq t_N = 0.78$ . Начальное условие:  $x(0) = [0.09, 0.09, 0]^T$ . Ограничения на управление:  $-10 \leq u \leq 10$ .

Критерий качества управления:  $I = x_3(t_N)$ . Необходимо минимизировать критерий при помощи выбора управления, удовлетворяющего заданным ограничениям.

В [5] приведены результаты решения поставленной задачи при помощи принципа максимума Л.С.Понтрягина:  $I = 0.133094$ , метода динамического программирования:  $I = 0.13336$ , алгоритма Нэлдера–Мида:  $I = 0.133366$  и эволюционных алгоритмов (табл. 2).

При решении поставленной задачи при помощи ГА с бинарным кодированием будем использовать следующие параметры: оператор селекции – турнирный отбор, количество элитных особей - 2, оператор скрещивания – равномерное скрещивание, оператор мутации – обыкновенная мутация с параметром 0.01, длина битовой строки – 11 ( $N = 40$ ) и 7 ( $N = 20$ ). Для численного решения дифференциального уравнения выберем метод Рунге–Кутты 4-го порядка. Решим задачу при различных размерах популяции и различном количестве популяций, а также при разном количестве интервалов разбиения при переходе к дискретной системе.

При решении поставленной задачи при помощи ГА с вещественным кодированием будем использовать следующие параметры: оператор селекции – турнирный отбор, количество элитных особей - 2, оператор скрещивания – кроссовер, имитирующий двоичный с параметром 2, оператор мутации – неравномерная мутация с параметром 1. Остальные параметры алгоритма выберем такими же, как и в ГА с бинарным кодированием.

Полученные при помощи ГА результаты сравним с результатами, приведенными в [5] для эволюционных алгоритмов (ЭА) (см. табл. 2). Результаты работы ГА с вещественным кодированием для  $N = 40, 60$  особей в популяции и 3000 популяций представлены на рис. 1 и 2.

Таблица 2

	Количество особей/популяций	ЭА	ГА с бинарным кодированием	ГА с вещественным кодированием
N=20	40/2000	0.134155	0.13423	0.134255
	60/3000	0.134155	0.134227	0.134238
	100/3000	0.134155	0.134217	0.134183
N=40	40/2000	---	0.133371	0.13444
	60/3000	0.13336	0.133364	0.13358
	100/3000	0.13336	0.133364	0.13341

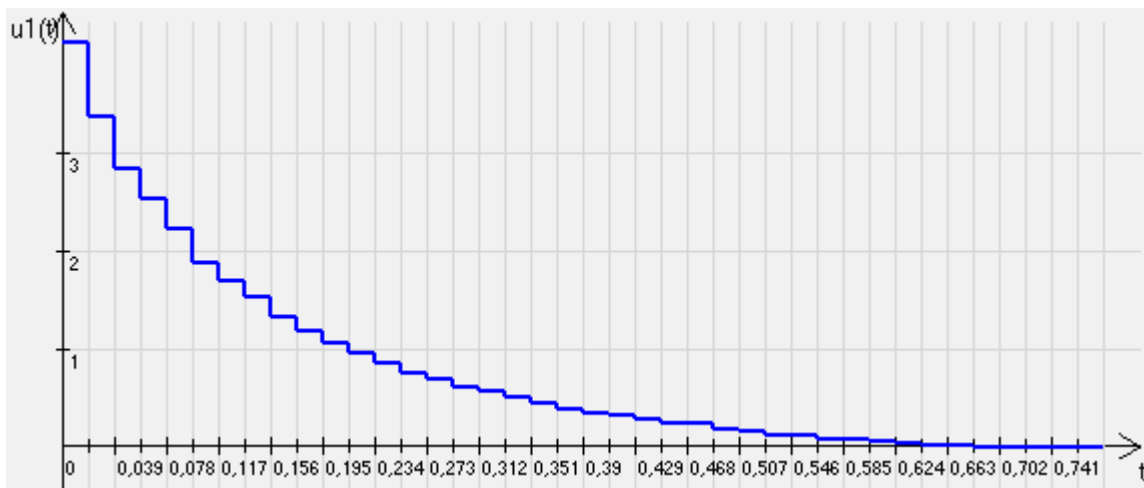


Рис. 1 Оптимальное управление

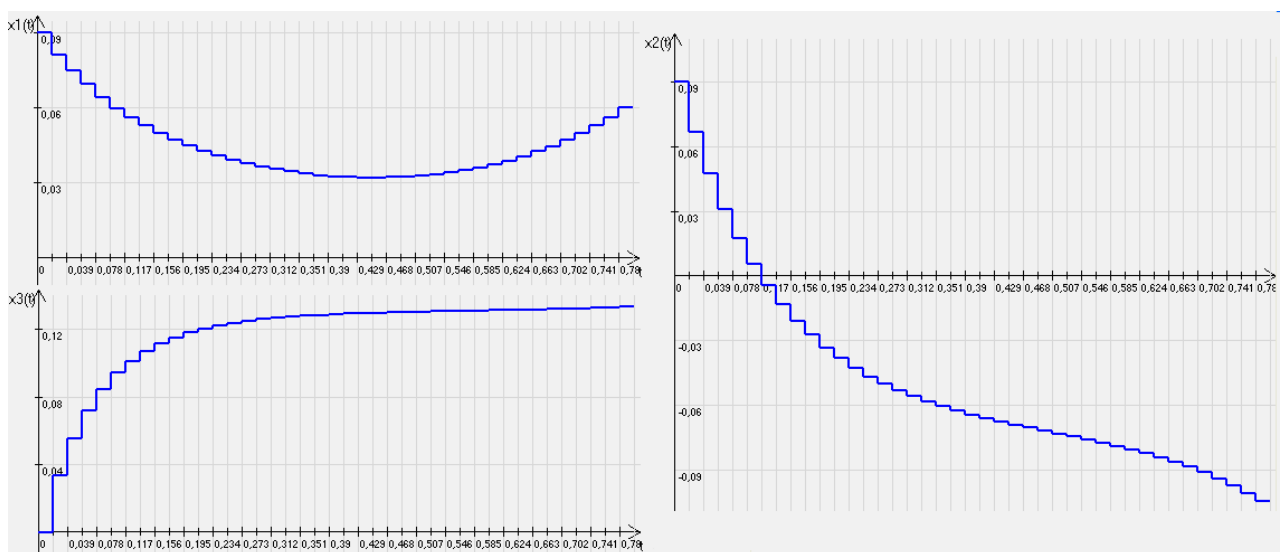


Рис. 2 Оптимальная траектория

На рис.3 представлено оптимальное управление, найденное при помощи эволюционных алгоритмов [5].

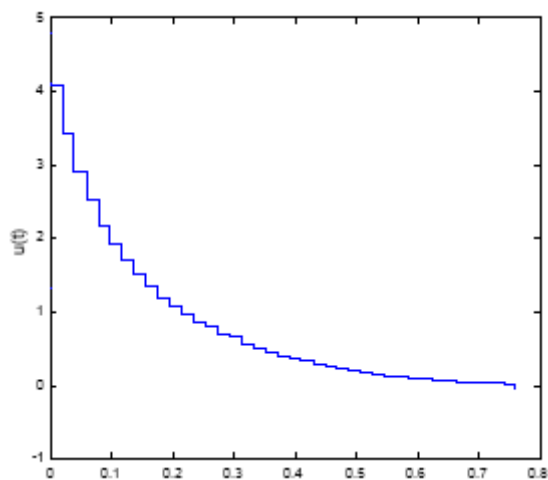


Рис. 3. Оптимальное управление (эволюционные алгоритмы)

По полученным результатам можно сделать вывод, что ГА с бинарным и вещественным кодированием находит решение данной задачи близкое к решению, найденному при помощи эволюционных алгоритмов. Сравнивая между собой ГА с бинарным кодированием и ГА с вещественным кодированием, можно заметить, что ГА с бинарным кодированием работают эффективней. Однако за счет отсутствия операций кодирования и декодирования ГА с вещественным кодированием работают быстрее, чем ГА с бинарным кодированием.

### Заключение

В данной работе предложен метод решения задачи поиска оптимального управления непрерывными детерминированными системами при помощи генетических алгоритмов условной оптимизации с бинарным и вещественным кодированием. Сформировано соответствующее программное обеспечение, с помощью которого была решена задача оптимального управления химическим процессом. Результаты решения сравнивались с результатами, полученными с использованием эволюционного алгоритма. В результате сравнения сделан вывод о том, что предложенные ГА с бинарным и вещественным кодированием эффективно работают при решении задач поиска оптимального управления непрерывными системами. В ходе исследования было установлено, что результаты сильно зависят от параметров ГА, однако широкие возможности их подбора позволяют получать приближенные решения хорошего качества.

## Библиографический список

1. Пантелеев А.В. Метлицкая Д.В. Генетические алгоритмы условной оптимизации с бинарным кодированием. Генетические алгоритмы условной оптимизации с вещественным кодированием.// Гл. 1 и 2 в кн.: Пантелеев А.В. Метаэвристические алгоритмы поиска глобального экстремума.- М.: Изд-во МАИ-ПРИНТ, 2009. С. 8-46.
2. Пантелеев А.В., Метлицкая Д.В. Применение генетических алгоритмов с бинарным и вещественным кодированием к задаче поиска условного экстремума функций // Теоретические вопросы вычислительной техники и программного обеспечения. Межвузовский сборник научных трудов, МИРЭА, 2010. С. 156-165.
3. Пантелеев А.В., Метлицкая Д.В. Комплекс программных средств «Генетические алгоритмы условной оптимизации с бинарным и вещественным кодированием»// Электронный журнал «Труды МАИ», 2010, №37, <http://www.mai.ru/science/trudy/published.php?ID=13421>.
4. Пантелеев А.В., Бортакровский А.С. Теория управления в примерах и задачах.- М.: Высш. шк., 2003.
5. Lopez Cruz I. L., Van Willigenburg L. G., Van Straten G. Evolutionary Algorithms for optimal control of chemical processes. Wageningen University, Netherlands.
6. Киреев В.И., Пантелеев А.В. Численные методы в примерах и задачах.- М.: Высш. шк., 2008.

## Сведения об авторе

Метлицкая Дарья Вадимовна студентка Московского авиационного института (государственного технического университета); e-mail: dashametlickaja@rambler.ru.