

УДК 519.85 + 517.977.58

ББК 22.18 + 22.19

РАЗРАБОТКА ИНТЕРВАЛЬНЫХ МЕТАЭВРИСТИЧЕСКИХ МЕТОДОВ МИНИМИЗАЦИИ ДЛЯ ПОИСКА ОПТИМАЛЬНОГО ПРОГРАММНОГО УПРАВЛЕНИЯ¹

Пантелеев А. В.², Пановский В. Н.³

(Московский авиационный институт, Москва)

Разработаны алгоритмическое и программное обеспечение метаэвристических интервальных методов глобальной условной оптимизации (усредненных концов путей, стохастической сетки, разбросанного поиска) и методика их применения для решения задачи поиска оптимального программного управления нелинейными детерминированными системами. Приведено решение прикладной задачи трехмерного сближения с маневрирующим объектом, проведен сравнительный анализ эффективности.

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

1. Введение

В современной математике достаточно большое внимание уделяется решению задач глобальной оптимизации, которые возникают, например, в ходе выбора наилучших параметров конструкций самолетов, вертолетов, космических аппаратов

¹ Исследование выполнено при финансовой поддержке РФФИ в рамках научных проектов №16-31-00115 мол_а, №16-07-00419 А.

² Андрей Владимирович Пантелеев, доктор физико-математических наук, профессор (avpanteleev@inbox.ru).

³ Валентин Николаевич Пановский, аспирант (panovskiy.v@yandex.ru).

(веса, дальности полета, аэродинамических характеристик). Существующие численные методы используют разнообразные подходы, но их использование связано с большими вычислительными затратами, излишними требованиями к постановке задачи, трудностями в достижении сходимости метода [9, 12].

В статье предлагаются новые методы поиска глобального условного экстремума функций многих переменных, использующие аппарат интервального анализа [8, 10, 11, 13]. Существующие интервальные методы оптимизации можно разделить на методы безусловной оптимизации (алгоритмы Мура–Скелбоу, Ичиды–Фуджи, Дюсселя, интервальный алгоритм «имитации отжига», метод случайного интервального дробления и др. [15–17]) и условной оптимизации (методы Хансена, Мура и др. [11, 14]).

Предлагаемые интервальные методы глобальной условной оптимизации относятся к классу метаэвристических методов. Метаэвристическим алгоритмом называют алгоритм, не имеющий строгого обоснования, но, тем не менее, позволяющий найти приемлемое решение задачи в большинстве практически значимых случаев. Он не гарантирует нахождения лучшего решения, не гарантирует нахождения решения, даже если оно существует («проскакивание решения»). Однако существенным достоинством таких алгоритмов является их низкая вычислительная сложность, что позволяет их применять для решения задач повышенной трудности, не накладывая существенных ограничений на элементы постановок задач (дифференцируемость, выпуклость, ограниченность и т.д.). При этом необходимые и достаточные условия экстремума [4] в алгоритмах не используются. Метаэвристические методы объединяют в себе один или несколько эвристических методов (процедур), опирающихся на стратегию более высокого уровня.

Существующие численные методы поиска оптимального управления включают в себя достаточно большое количество методов, которые используют принцип максимума Понтрягина и уравнение Беллмана, а также прямые методы, например, градиентные методы (методы первого порядка), методы второго порядка, основывающиеся на тейлоровской аппроксимации

функции Кротова–Беллмана, разнообразные методы улучшения и др. [1, 7]

Несмотря на большое количество проведенных исследований, проблема численного решения нелинейных задач оптимального управления остается актуальной и требующей разработки новых подходов, к которым относится применение интервального анализа совместно с метаэвристическими стратегиями [2–6]. Предлагаемые в статье методы дают возможность нахождения глобального экстремума критерия оптимальности для достаточного широкого класса задач.

2. Используемые понятия и обозначения

Основной идеей интервального анализа является окружение вещественных чисел интервалами, а вещественных векторов – интервальными векторами, или параллелотопами [10]. Для обозначения интервала используются строчные латинские буквы, заключенные в квадратные скобки ($[a]$, $[b]$, $[c]$, ...) или привычное представление ($[a_i; a_u]$, $[b_i; b_u]$, $[c_i; c_u]$, ...), где указывается нижняя и верхняя границы. Для задания параллелотопа применяется то же обозначение с полужирным начертанием букв ($[X] = [x_1] \times \dots \times [x_n]$).

Для произвольного интервала $[x]$ определены: нижняя граница

$$lb([x]) = \underline{[x]} = \sup \{ \xi \in \mathbb{R} \cup \{-\infty, \infty\} \mid \forall \zeta \in [x]: \xi \leq \zeta \},$$

верхняя граница

$$ub([x]) = \overline{[x]} = \inf \{ \xi \in \mathbb{R} \cup \{-\infty, \infty\} \mid \forall \zeta \in [x]: \zeta \leq \xi \},$$

ширина непустого интервала

$$\omega([x]) = \overline{[x]} - \underline{[x]},$$

средняя точка ограниченного и непустого интервала

$$mid([x]) = \frac{\underline{[x]} + \overline{[x]}}{2}.$$

Те же параметры определены и для параллелотопов. Нижняя, верхняя границы и средняя точка становятся векторами,

ширина же рассчитывается как максимум из ширин всех образующих параллелотоп компонентов.

Интервальной оболочкой множества $X \subset \mathbb{R}^n$ называется параллелотоп с наименьшей шириной, который содержит X . Если множество берется в квадратные скобки, это значит, что рассматривается интервальная оболочка этого множества.

Пусть \circ – некоторая бинарная операция, тогда $[x] \circ [y] = [\{\xi_1 \circ \xi_2 \mid \xi_1 \in [x], \xi_2 \in [y]\}]$. Пусть f – некоторый унарный оператор, тогда $f([x]) = [\{f(\xi) \mid \xi \in [x]\}]$. Описанная бинарная операция позволяет определить арифметические операции над интервалами [10].

Множество интервалов обозначается как \mathbb{IR} , интервальных векторов – \mathbb{IR}^n . Пусть имеется некоторая функция f , действующая из \mathbb{R}^n в \mathbb{R} . Функция $[f](\cdot)$ называется интервальной функцией включения для f , действующей из \mathbb{IR}^n в \mathbb{IR} , если $\{f(\xi) \mid \xi \in [\mathbf{x}]\} \subseteq [f]([\mathbf{x}])$, $\forall [\mathbf{x}] \in \mathbb{IR}^n$. Функция включения позволяет получить априорную оценку множества значений функции, даже если оно не является выпуклым или связным (если вместо переменных используются интервалы и соответствующие арифметические операции, то полученная оценка называется оценкой прямого образа функции).

3. Постановка интервальной задачи ε -минимизации

Классическая постановка задачи условной оптимизации имеет вид: пусть имеются параллелотоп $[\mathbf{s}]$, задающий множество допустимых решений, и целевая функция $f: \mathbb{R}^n \rightarrow \mathbb{R}$. Требуется найти точку x^* , такую что

$$(1) \quad f(x^*) = \min_{x \in [\mathbf{s}]} f(x) \Leftrightarrow \forall x \in [\mathbf{s}]: f(x^*) \leq f(x).$$

Приведем интервальный аналог поставленной задачи: пусть имеются параллелотоп $[\mathbf{s}]$, задающий множество допустимых решений, целевая функция $f: \mathbb{R}^n \rightarrow \mathbb{R}$, малое число $\varepsilon > 0$. Требуется найти параллелотоп $[\mathbf{p}]^*$, такой что

$$(2) \quad [\mathbf{p}]^* \subseteq [\mathbf{s}], \omega([\mathbf{p}]^*) \leq \varepsilon, \exists [\mathbf{x}] \subseteq [\mathbf{s}], \omega([\mathbf{x}]) \geq \varepsilon, \overline{[f]}([\mathbf{x}]) < \overline{[f]}([\mathbf{p}]^*).$$

4. Стратегии интервальных метаэвристических методов глобальной условной оптимизации

4.1. МЕТОД УСРЕДНЕННЫХ КОНЦОВ ПУТЕЙ

В основе метода лежат следующие процедуры:

- построение случайной сетки на области поиска (представляет собой разбиение целевого параллелопада $[\mathbf{s}]$ на множество $\{[\mathbf{n}]_i\}_{i=1}^N$ непересекающихся параллелопадов, таких что

$$[\mathbf{s}] = \bigcup_{i=1}^N [\mathbf{n}]_i);$$

- выбор случайного параллелопада из сетки;
- поиск пути (последовательности параллелопадов, в которой каждый следующий параллелопад имеет общую гиперплоскость с предыдущим) до «оптимального» решения по сетке;
- полученный в ходе выполнения описанных процедур «перспективный» параллелопад запоминается.

Процедуры повторяются несколько раз. Из полученных параллелопадов берутся их средние точки, на основе которых строится новый целевой параллелопад (он рассматривается как усредненный из «перспективных» параллелопадов, являющихся концами последовательностей, полученных в результате поиска, начинающегося с некоторого параллелопада, принадлежащего построенной сетке), к которому снова применяется метод. Так повторяется до тех пор, пока ширина целевого параллелопада не будет удовлетворять условию точности.

4.2. МЕТОД СТОХАСТИЧЕСКОЙ СЕТКИ

Стратегия данного метода заключается в построении на целевом параллелопаде сетки (множества непересекающихся параллелопадов, объединение которых дает искомый целевой параллелопад) случайным образом. Далее для некоторых элементов сетки ищется оценка прямого образа функции, выбирается наиболее «перспективный» параллелопад (с наименьшим

значением нижней грани оценки прямого образа). Такая процедура повторяется несколько раз. Средние точки выбранных параллелотопов определяют новый параллелотоп, к которому снова применяется алгоритм. Так повторяется до тех пор, пока ширина целевого параллелотопа не будет удовлетворять условию точности.

4.3. МЕТОД ИНТЕРВАЛЬНОГО РАЗБРОСАННОГО ПОИСКА

В основе точечного метода разбросанного поиска (scatter search) лежит обработка множества элементарных исходов, состоящих из найденных на предыдущем этапе «перспективных» решений. Фактически обработка заключается в комбинации, улучшении и обновлении множества элементарных исходов. В основе интервального метода лежат пять методов: метод генерации разнообразных решений, методы конструирования и обновления множества элементарных исходов, метод генерации подмножеств, метод комбинации решений, метод улучшения решений.

Процедура повторяется до тех пор, пока ширина целевого параллелотопа не будет удовлетворять условию точности.

5. Алгоритмы интервальных метаэвристических методов глобальной условной оптимизации

5.1. МЕТОД УСРЕДНЕННЫХ КОНЦОВ ПУТЕЙ

Шаг 1. Задать $[s]$ – область поиска; ε – параметр точности (отвечает за ширину итогового параллелотопа); w – «скорость» уменьшения параллелотопа; p_{amount} – количество точек для построения случайной сетки; $p_{attempts}$ – количество повторений; r_{max} – максимальное количество возвращений (используется во время поиска пути до «оптимального» решения). Положить целевой параллелотоп $[t] = [s]$.

Шаг 2. Если $\omega([t]) < \varepsilon$, то закончить работу алгоритма, вернув параллелотоп $[t]$. В противном случае положить множество концов путей (последний элемент в последовательности параллелотопов, представляющей путь) $E = \emptyset$ и перейти к шагу 3.

Шаг 3. Случайным образом, используя равномерный закон распределения, выбрать в целевом параллелотопе p_{amount} точек.

Шаг 4. Разбить целевой параллелотоп относительно выбранных на шаге 3 точек на множество параллелотопов (так как любой параллелотоп является n -мерным параллелепипедом (n – размерность задачи), то разбиение можно делать следующим образом: в каждой точке строятся n гиперплоскостей, каждая из которых параллельна одной паре сторон данного параллелепипеда; совокупность этих гиперплоскостей порождает сетку на параллелотопе).

Шаг 5. Из полученной сетки выбрать случайный параллелотоп $[p]$.

Шаг 6. Процедура поиска пути от параллелотопа $[p]$ к «оптимальному» параллелотопу (путь – последовательность параллелотопов $\{[w]_i\}_{i=1}^m$, таких, что все параллелотопы $[w]_i, [w]_{i+1}, i = 1, \dots, m - 1$, граничат, т.е. имеют общую гиперплоскость):

Шаг 6.1. Из параллелотопов сетки сформировать множество параллелотопов $N = \{[n]_i\}_{i=1}^l$, граничащих с параллелотопом $[p]$. Поместить в список параллелотопов, образующих путь, начальный элемент ($P = \{[p]\}$) и задать количество возвратов $r = 0$.

Шаг 6.2. Если $r \leq r_{max}$, то перейти к шагу 6.3. В противном случае перейти к шагу 6.6.

Шаг 6.3. Выбрать случайный параллелотоп $[n]_j \in N$, удалить его из N . Если

$$[f]([n]_j) \leq [f]([p]),$$

то перейти к шагу 6.4. В противном случае перейти к шагу 6.5.

Шаг 6.4. Если $[n]_j \in P$, то положить $r = r + 1$ и перейти к шагу 6.2. В противном случае добавить $[n]_j$ в P . Положить $[p] = [n]_j$ и вернуться к шагу 6.1.

Шаг 6.5. Если $N \neq \emptyset$, то перейти к шагу 6.3, в противном случае перейти к шагу 6.6.

Шаг 6.6. Параллелотоп $[p]$ добавить в множество концов путей E . Переход к шагу 7.

Шаг 7. Если в множестве E содержится $p_{attempts}$ элементов, то перейти к шагу 8. В противном случае перейти к шагу 3.

Шаг 8. Найти вектор

$$m = \frac{1}{p_{attempts}} \sum_{i=1}^{p_{attempts}} mid([\mathbf{e}]_i), \text{ где } [\mathbf{e}]_i \in E.$$

Заменить параллелотоп $[\mathbf{t}]$ на параллелотоп

$$\left[m_1 - \frac{w}{2} \cdot wid, m_1 + \frac{w}{2} \cdot wid \right] \times \dots \times \left[m_n - \frac{w}{2} \cdot wid, m_n + \frac{w}{2} \cdot wid \right] \cap [\mathbf{t}],$$

где $wid = \omega([\mathbf{t}])$. Перейти к шагу 2.

5.2. МЕТОД СТОХАСТИЧЕСКОЙ СЕТКИ

Шаг 1. Задать $[\mathbf{s}]$ – область поиска; ε – параметр точности (отвечает за ширину итогового параллелотопа); w – «скорость» уменьшения параллелотопа; p_{amount} – количество гиперплоскостей для построения случайной сетки; $p_{attempts}$ – количество повторений; $p_{analyze}$ – количество анализируемых параллелотопов. Положить целевой параллелотоп $[\mathbf{t}] = [\mathbf{s}]$.

Шаг 2. Если $\omega([\mathbf{t}]) < \varepsilon$, то закончить работу алгоритма, вернуть параллелотоп $[\mathbf{t}]$. В противном случае положить множество выбранных параллелотопов $E = \emptyset$ и перейти к шагу 3.

Шаг 3. Случайным образом, используя равномерный закон распределения, выбрать в целевом параллелотопе p_{amount} точек.

Шаг 4. Разбить целевой параллелотоп относительно выбранных на шаге 3 точек на множество параллелотопов $V = \{[\mathbf{b}]_j\}_{j=1}^l$. Так как любой параллелотоп является n -мерным параллелепипедом (n – размерность задачи), то разбиение можно реализовать следующим образом: в каждой точке строится гиперплоскость, параллельная одной (случайно выбранной) паре сторон данного параллелепипеда; совокупность этих гиперплоскостей порождает разбиение исходного параллелотопа. В полученном множестве следует удалять случайно выбранные параллелотопы до тех пор, пока количество элементов не станет равным $p_{analyze}$. В результате получится стохастическая сетка.

Шаг 5. Из полученной сетки выбрать параллелотоп

$$[\mathbf{p}] = \text{Arg} \min_{j=1, \dots, l} [f](\underline{[\mathbf{b}]_j})$$

и добавить его в множество E .

Шаг 6. Если в множестве E содержится $p_{attempts}$ элементов, то перейти к шагу 7. В противном случае перейти к шагу 3.

Шаг 7. Найти вектор

$$m = \frac{1}{p_{attempts}} \sum_{i=1}^{p_{attempts}} mid([\mathbf{e}]_i), \text{ где } [\mathbf{e}]_i \in E.$$

Заменить параллелотоп $[\mathbf{t}]$ на параллелотоп

$$\left[m_1 - \frac{w}{2} \cdot wid, m_1 + \frac{w}{2} \cdot wid \right] \times \dots \times \left[m_n - \frac{w}{2} \cdot wid, m_n + \frac{w}{2} \cdot wid \right] \cap [\mathbf{t}],$$

где $wid = \omega([\mathbf{t}])$. Перейти к шагу 2.

5.3. МЕТОД ИНТЕРВАЛЬНОГО РАЗБРОСАННОГО ПОИСКА

Шаг 1. Задать $[\mathbf{s}]$ – область поиска; ε – параметр точности (отвечает за ширину итогового параллелотопа); r_{gm} , r_{rsc} , r_{cs} , $r_{im} \in (0;1)$ – доли для генерации; p_{size} , rs_{size} , sub_{size} – размеры множеств P , RS , SUB соответственно; max_{im} – количество точек для улучшения; σ – величину влияния ширины.

Шаг 2. Метод генерации разнообразных решений. Сформировать множество P из p_{size} параллелотопов $[m_1 - \omega_1; m_1 + \omega_1] \times \dots \times [m_1 - \omega_1; m_1 + \omega_1] \cap [\mathbf{s}]$, где величина $\omega_k = 0,5 \cdot r_{gm} \cdot \omega([\mathbf{s}])$, m_k – случайная точка на отрезке, соответствующем k -й компоненте параллелотопа $[\mathbf{s}]$. Отсортировать множество P по возрастанию величины $[f](\underline{[\mathbf{p}]_i})$, где $[\mathbf{p}]_i \in P$.

Шаг 3. Метод конструирования множества элементарных исходов. Сформировать множество RS из $[r_{rsc} \cdot p_{size}]$ (здесь $[\cdot]$ означает целую часть от числа) первых элементов множества P . Выбранные элементы удалить из P . Создать множество

$$D = \left\{ d_i = \left[\sum_{j=1}^{[r_{rsc} \cdot p_{size}]} h_{\infty}([\mathbf{p}]_i, [\mathbf{rs}]_j) \right] \right\}_{i=1}^{p_{size} - [r_{rsc} \cdot p_{size}]},$$

где $h_{\infty}([a], [b])$ – метрика Хаусдорфа [10], $[\mathbf{rs}]_j \in RS$. Упорядочить множество D расстояний между параллелотопом $[\mathbf{p}]_i$ (оставшимся после удаления) и параллелотопами $[\mathbf{rs}]_j \in RS$ по убыва-

нию. В множество RS добавить $(rs_{size} - [r_{rsc} \cdot p_{size}])$ параллелотопов из P , которым соответствуют первые (наибольшие) значения из D (добавляются наиболее удаленные параллелотопы от имеющихся в RS для поддержания разнообразия).

Шаг 4. Отсортировать множество RS элементарных исходов по возрастанию величины $[f](\mathbf{rs}_i)$, где $\mathbf{rs}_i \in RS$.

Шаг 5. Если $\omega(\mathbf{rs}_1) \geq \varepsilon$, то перейти к шагу 6. В противном случае закончить работу алгоритма, вернув параллелотоп \mathbf{rs}_1 .

Шаг 6. Метод генерации подмножеств. Создать множество SUB , состоящее из всех подмножеств размера sub_{size} множества RS .

Шаг 7. Создать множество комбинированных решений $CS = \emptyset$.

Шаг 8. Метод комбинации решений. Для каждого подмножества

$$sub_i = \left\{ [s]_j^i \in RS, j = 1, \dots, sub_{size} \right\} \in SUB$$

выполнить: составить вектор

$$mw^i = \left(\min_{j=1, \dots, sub_{size}} \omega([s_1]_j^i), \dots, \min_{j=1, \dots, sub_{size}} \omega([s_n]_j^i) \right)^T,$$

создать параллелотоп $[\mathbf{u}]^i = \bigcup_{j=1}^{sub_{size}} [s]_j^i$, сгенерировать n случайных

точек $\left\{ p^k \in [\mathbf{u}]^i \right\}_{k=1}^n$. Построить n параллелотопов

$$\left[[p_1^k - \omega_1; p_1^k + \omega_1] \times \dots \times [p_n^k - \omega_n; p_n^k + \omega_n] \cap [\mathbf{u}]^i \right]_{k=1}^n$$

на сгенерированных точках, где $\omega_k = 0,5 \cdot r_{cs} \cdot mw_k^i$, и добавить их в множество CS .

Шаг 9. Создать множество улучшенных решений $IS = \emptyset$.

Шаг 10. Метод улучшения решений. Для каждого из параллелотопов $[\mathbf{cs}]_i \in CS$ сгенерировать случайным образом множество $\{p^k = (p_1^k, \dots, p_n^k)^T \in [\mathbf{cs}]_i\}_{k=1}^{\max_{im}}$ точек. Выбрать точку $p^* = \text{Arg} \min_{k=1, \dots, \max_{im}} f(p^k)$. Добавить в множество IS параллелотоп

$\left[p_1^* - \omega_1; p_1^* + \omega_1 \right] \times \dots \times \left[p_n^* - \omega_n; p_n^* + \omega_n \right] \cap [\mathbf{cs}]_i$, где величина $\omega_j = 0,5 \cdot r_{im} \cdot \omega([\mathbf{cs}]_i)$.

Шаг 11. Метод обновления множества элементарных исходов. Для каждого параллелопада $[\mathbf{is}]_i \in IS$, если есть такое число j , для которого выполнено соотношение

$$\underline{[f]}([\mathbf{is}]_i) + \sigma \cdot \omega([\mathbf{is}]_i) \leq \underline{[f]}([\mathbf{is}]_j) + \sigma \cdot \omega([\mathbf{is}]_j),$$

то вставить перед j -м параллелопадом в множестве RS параллелопад $[\mathbf{is}]_i$ и последний элемент множества RS удалить.

Шаг 12. Вернуться к шагу 5.

6. Поиск оптимального программного управления непрерывными системами

6.1. ПОСТАНОВКА ЗАДАЧИ

Поведение модели объекта управления описывается дифференциальным уравнением

$$(3) \quad \dot{x}(t) = f(t, x(t), u(t)),$$

где t – непрерывное время, $t \in T = [t_0; t_1]$ – заданный промежуток времени функционирования системы; $x \in \mathbb{R}^n$ – вектор состояния системы; $u \in [\mathbf{U}] \subset \mathbb{R}^q$ – вектор управления; $[\mathbf{U}]$ – множество допустимых значений управления, представляющее собой параллелопад; $f(t, x, u) = (f_1(t, x, u), \dots, f_n(t, x, u))^T$ – непрерывная вектор-функция.

Начальное состояние задано и равно

$$(4) \quad x(t_0) = x_0.$$

Конечное состояние $x(t_1)$ должно удовлетворять условиям

$$(5) \quad \Gamma_i(x(t_1)) = 0, \quad i = 1, \dots, l,$$

где $0 \leq l \leq n$, функции $\Gamma_i(x)$ – непрерывно дифференцируемы; система векторов $\{\partial \Gamma_i(x) / \partial x_1, \dots, \partial \Gamma_i(x) / \partial x_n\}$, $i = 1, \dots, l$, линейно независима $\forall x \in \mathbb{R}^n$.

При управлении используется информация только о времени t (применяется программное управление).

Множество допустимых процессов $D(t_0, x_0)$ определяется как множество пар $d = (x(\cdot), u(\cdot))$, включающих кусочно-дифференцируемую траекторию $x(\cdot)$ и кусочно-непрерывное управление $u(\cdot)$, где $u(t) \in [\mathbf{u}] \forall t \in T$, удовлетворяющих уравнению состояния (3), условиям (4) и (5).

На множестве допустимых процессов определен функционал качества управления

$$(6) \quad I(d) = \int_{t_0}^{t_1} f^0(t, x(t), u(t)) dt + F(x(t_1)),$$

где $f^0(t, x, u)$, $F(x)$ – заданные непрерывные функции.

Требуется найти пару $d^* = (x^*(\cdot), u^*(\cdot)) \in D(t_0, x_0)$, на которой достигается минимальное значение функционала (6) на множестве допустимых процессов.

6.2. СТРАТЕГИЯ ПОИСКА РЕШЕНИЯ

Основным элементом предлагаемой стратегии является последовательное сведение поставленной задачи к соответствующей задаче нелинейного программирования, а затем – к задаче интервальной ε -минимизации и решение последней с помощью разработанных интервальных методов.

Для более простого вычисления интегрального члена в критерии (6) к системе (3) добавляется уравнение $\dot{x}_{n+1}(t) = f^0(t, x(t), u(t))$ с начальным условием $x_{n+1}(t_0) = 0$, тогда значение функционала качества (6) определяется выражением

$$I = x_{n+1}(t_1) + F(x(t_1)).$$

Для сведения к задаче со свободным правым концом траектории к терминальному члену функционала добавляются либо классические штрафные функции, характеризующие степень выполнения условий (5), либо их интервальный аналог вида

$$(7) \quad \sum_{i=1}^l R_i h_\infty^2([\Gamma_i(x(t_1)); \Gamma_i(x(t_1))], [-\xi_i; \xi_i]),$$

где

$$h_\infty([a], [b]) = [\min(h_\infty^0([a], [b]), h_\infty^0([a], [b])), \max(h_\infty^0([a], [b]), h_\infty^0([a], [b]))]$$

– метрика Хаусдорфа;

$$h_{\infty}^0([a],[b]) = \inf \{r \mid [a] \subseteq [b] + [-r;r], r \geq 0\}$$

– мера близости интервалов; R_i – величины штрафов; $\xi_i > 0$ – величины допустимой погрешности выполнения конечного условия, которые задаются пользователем.

Искомое субоптимальное программное управление предлагается искать в двух различных классах функций: кусочно-постоянных и кусочно-линейных.

Поскольку ищется наилучшее управление в каждом из классов, не совпадающих с классом кусочно-непрерывных функций, то найденное управление является субоптимальным, рассматриваемым в качестве кандидата в решение задачи.

Так как при решении задачи применяется интервальный подход, то каждому из описанных управлений сопоставляется интервальный аналог, задаваемый интервальным вектором:

• кусочно-постоянное интервальное управление (рис. 1, а); для управления такого типа необходимо задать значения функции $u(t)$ в N моментах времени $\tau = t + (t_1 - t_0) \cdot i / N$, $i = 0, \dots, N - 1$, т.е. управлению можно однозначно сопоставить интервальный вектор

$$[u] = \underbrace{[u_1(\tau_0)] \times \dots \times [u_q(\tau_0)]}_{[u(\tau_0)]} \times \dots \times \underbrace{[u_1(\tau_{N-1})] \times \dots \times [u_q(\tau_{N-1})]}_{[u(\tau_{N-1})]};$$

соответствующее интервальное управление будет находиться по формуле

$$[u](t) \in [u(\tau_i)] = [\underline{u}(\tau_i); \bar{u}(\tau_i)], \tau_i \leq t < \tau_{i+1}, i = 0, \dots, N - 1;$$

• кусочно-линейное интервальное управление (рис. 1, б); для управления такого типа необходимо задать дополнительное значение управления $u(t)$ на последнем временном интервале, поэтому интервальный вектор, который ставится в соответствие управлению, имеет вид

$$[u] = \underbrace{[u_1(\tau_0)] \times \dots \times [u_q(\tau_0)]}_{[u(\tau_0)]} \times \dots \times \underbrace{[u_1(\tau_N)] \times \dots \times [u_q(\tau_N)]}_{[u(\tau_N)]};$$

ющее интервальное управление будет находиться по формуле

$$[u](t) \in \frac{\tau_{i+1} - t}{\tau_{i+1} - \tau_i} \cdot [u(\tau_i)] + \frac{t - \tau_i}{\tau_{i+1} - \tau_i} \cdot [u(\tau_{i+1})], \quad \tau_i \leq t \leq \tau_{i+1}, \quad i=0, \dots, N-1.$$

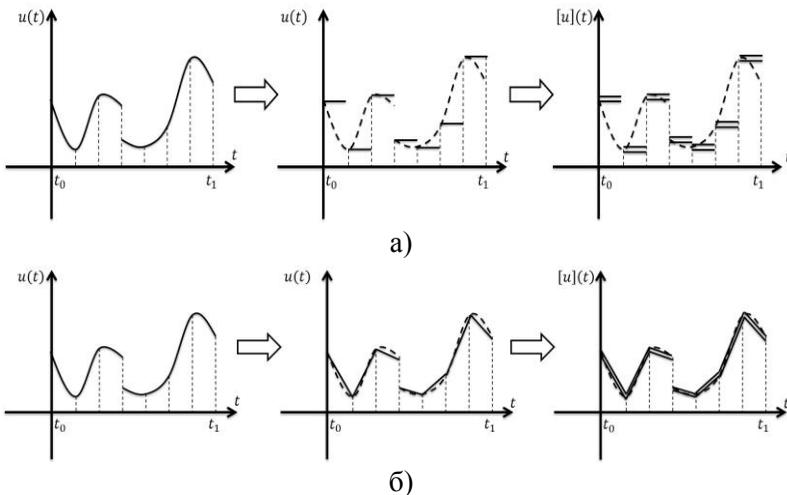


Рис. 1. Замена кусочно-непрерывного управления интервальным кусочно-постоянным («а») и кусочно-линейным («б»)

Интервальной траекторией называется решение уравнения (3) с начальным условием (4), соответствующее заданному интервальному управлению.

Если система (3), описывающая поведение модели объекта управления, нежесткая, для ее численного интегрирования применяются явные методы (методы Эйлера, Эйлера-Коши, методы Рунге–Кутты различных порядков и др.). Если же система (3) жесткая, применяются неявные методы численного интегрирования (неявный метод Эйлера, метод трапеций, Адамса–Моултона, Гира и др.). Вычисление правых частей системы (3) производится по правилам интервальной арифметики [10].

Заметим, что описанные процедуры могут быть использованы при других способах параметризации управления, напри-

мер, кусочно-полиномиальном или применении разложений по системам ортонормированных базисных функций

7. Программное обеспечение. Решение прикладного примера

На основе изложенных алгоритмов сформирован комплекс программ поиска оптимального программного управления. Среда разработки – Microsoft Visual Studio, язык программирования – C#.

В качестве примера рассмотрим решение задачи преследования в трехмерном пространстве, рассмотренной в [18].

Поведение модели объекта управления описывается системой дифференциальных уравнений:

$$(8) \quad \begin{aligned} \dot{x}(t) &= V_x(t), \dot{V}_x(t) = u_x(t), \dot{x}^t(t) = V_x^t(t), \dot{V}_x^t(t) = u_x^t(t), \\ \dot{y}(t) &= V_y(t), \dot{V}_y(t) = u_y(t), \dot{y}^t(t) = V_y^t(t), \dot{V}_y^t(t) = u_y^t(t), \\ \dot{z}(t) &= V_z(t), \dot{V}_z(t) = u_z(t), \dot{z}^t(t) = V_z^t(t), \dot{V}_z^t(t) = u_z^t(t), \end{aligned}$$

где $r = (x, y, z)^T$ и $r = (x^t, y^t, z^t)^T$ – радиус-векторы, описывающие положение перехватчика и цели соответственно; $V = (V_x, V_y, V_z)^T$ и $V^t = (V_x^t, V_y^t, V_z^t)^T$ – векторы скоростей перехватчика и цели соответственно; $u = (u_x, u_y, u_z)^T$ и $u^t = (u_x^t, u_y^t, u_z^t)^T$ – векторы ускорения перехватчика (управление) и цели соответственно.

Пусть цель движется с постоянным ускорением: $u^t = (-1, -2, 0, 1)^T$ м/с², начальное положение цели: $r^t = (500, -600, 500)^T$ м, скорость цели: $V^t = (100, 100, 10)^T$ м/с, начальное положение перехватчика: $r = (0, 0, 0)^T$ м, начальная скорость перехватчика: $V = (150, 40, 5)^T$ м/с.

Время, за которое должен быть осуществлен перехват (координаты цели и перехватчика совпадают): $t_1 = 50$ с.

Тогда можно задать функционал качества управления:

$$I = (x(t_1) - x^t(t_1))^2 + (y(t_1) - y^t(t_1))^2 + (z(t_1) - z^t(t_1))^2.$$

Для сглаживания искомого интервального управления в функционал добавим слагаемое

$$\Delta I = 0,001 \cdot \int_0^{t_1} (u_x^2(t) + u_y^2(t) + u_z^2(t)) dt.$$

В результате решается задача минимизации функционала $I + \Delta I$.

В качестве метода дискретизации были выбраны явная и неявная схемы Эйлера, область поиска

$$[\mathbf{s}] = \underbrace{[-2; 2] \times \dots \times [-2; 2]}_{3 \cdot N}$$

(для кусочно-постоянного управления) и

$$[\mathbf{s}] = \underbrace{[-2; 2] \times \dots \times [-2; 2]}_{3 \cdot (N+1)}$$

(для кусочно-линейного управления), $N = 2$.

Параметры метода усредненных концов путей: $\varepsilon = 0,001$; $w = 0,5$; $p_{amount} = 75$; $p_{attempts} = 5$; $r_{max} = 5$.

Параметры метода стохастической сетки: $\varepsilon = 0,001$; $w = 0,9$; $p_{amount} = 100$; $p_{attempts} = 100$; $p_{analyze} = 1000$.

Параметры метода интервального разбросанного поиска: $\varepsilon = 0,001$; $r_{gm} = 0,8$; $r_{rsc} = 0,25$; $r_{cs} = 0,9$; $r_{im} = 0,9$; $p_{size} = 100$; $rs_{size} = 10$; $sub_{size} = 2$; $max_{im} = 1000$; $\sigma = 2500$.

На рис. 3 и 4 приведены графики траекторий цели и перехватчика, полученные с помощью интервальных кусочно-постоянных и кусочно-линейных управлений. Их сравнение с рис. 2 показывает близость результатов с найденными при использовании алгоритма пропорциональной навигации [18].

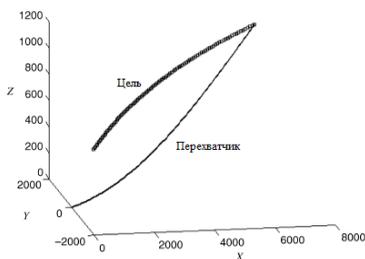


Рис. 2. График движения цели и перехватчика (при использовании алгоритма пропорциональной навигации [18])

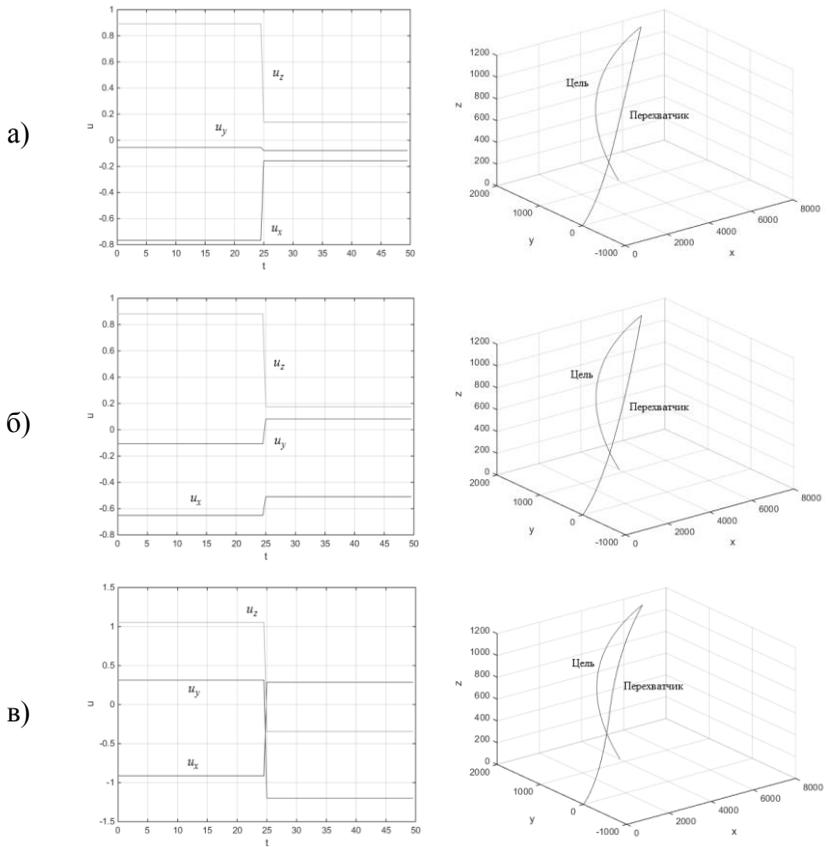


Рис. 3. Графики движения цели и перехватчика, законов изменения управления, найденные методами усредненных концов путей, стохастической сетки и интервального разброанного поиска (кусочно-постоянное управление)

В таблице 1 и 3 приведены интервалы значений критерия оптимальности, найденные в результате применения соответствующего метода интервальной минимизации в каждом из двух рассматриваемых классов управлений.

Таблица 1. Значение функционала качества управления (кусочно-постоянное управление)

Название метода	Значение функционала качества	
	Явная схема Эйлера	Неявная схема Эйлера
Метод усредненных концов путей	[0,0377; 0,0378]	[0,0371; 0,0372]
Метод стохастической сетки	[0,0359; 0,0360]	[0,0354; 0,0355]
Метод интервального разбросанного поиска	[0,0445; 0,0446]	[0,0437; 0,0438]

Таблица 2. Максимальные значения отклонения цели и перехватчика (кусочно-постоянное управление)

Название метода	Значение Δ_x ; Δ_y ; Δ_z	
	Явная схема Эйлера	Неявная схема Эйлера
Метод усредненных концов путей	0,1040; 0,0501; 0,0401	0,1035; 0,04993; 0,0409
Метод стохастической сетки	0,0632; 0,0735; 0,0053	0,0631; 0,0733; 0,0051
Метод интервального разбросанного поиска	0,0910; 0,0912; 0,0317	0,0908; 0,091; 0,0315

В таблицах 2, 4 величины отклонений вычисляются как $\Delta_x = \max |x(t_1) - x^t(t_1)|$, $\Delta_y = \max |y(t_1) - y^t(t_1)|$, $\Delta_z = \max |z(t_1) - z^t(t_1)|$.

Полученные данные о максимальных значениях отклонения являются приемлемыми с практической точки зрения и свидетельствуют об успешности применения разработанных алгоритмов. Наиболее точным оказался метод стохастической сетки, кусочно-линейное управление (таблица 3, 4) предпочтительнее по величине функционала, чем кусочно-постоянное (таблица 1, 2).

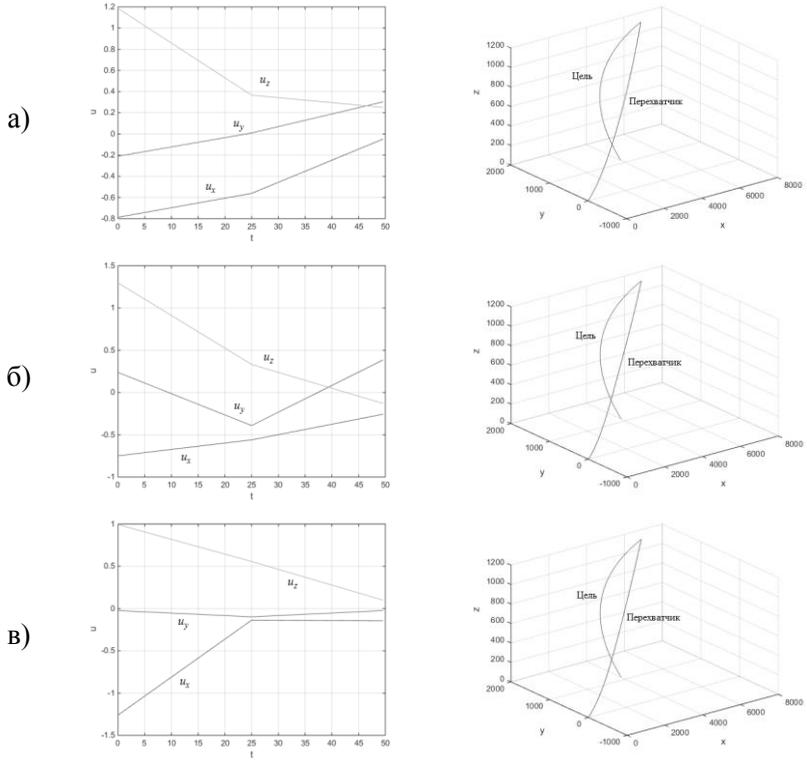


Рис. 4. Графики движения цели и перехватчика, законов изменения управления, найденные методами усредненных концов путей, стохастической сетки и интервального разброанного поиска (кусочно-линейное управление)

Таблица 3. Значение функционала качества управления (кусочно-линейное управление)

Название метода	Значение функционала качества	
	Явная схема Эйлера	Неявная схема Эйлера
Метод усредненных концов путей	[0,0347; 0,0348]	[0,0343; 0,0344]
Метод стохастической сетки	[0,0341; 0,0341]	[0,0340; 0,0341]

Название метода	Значение функционала качества	
	Явная схема Эйлера	Неявная схема Эйлера
Метод интервального разбро- санного поиска	[0,0345; 0,0346]	[0,0339; 0,0340]

Таблица 4. Максимальные значения отклонения цели и перехватчика (кусочно-линейное управление)

Название метода	Значение Δ_x ; Δ_y ; Δ_z	
	Явная схема Эйлера	Неявная схема Эйлера
Метод усредненных концов путей	0,0607; 0,0698; 0,0060	0,0609; 0,0697; 0,00055
Метод стохастической сетки	0,0636; 0,0431; 0,0353	0,0632; 0,043; 0,0035
Метод интервального разбро- санного поиска	0,0432; 0,0935; 0,0053	0,0431; 0,0932; 0,0051

8. Заключение

Разработаны алгоритмическое и программное обеспечение трех метаэвристических интервальных методов оптимизации: усредненных концов путей, стохастической сетки и разбросанного поиска. Поставлена задача интервальной ϵ -оптимизации, к которой была сведена задача поиска оптимального программного управления нелинейными непрерывными детерминированными системами. Решена прикладная задача преследования в трехмерном пространстве. Анализ полученных результатов свидетельствует об эффективности предложенных методов.

Литература

1. ГУРМАН В.И., РАСИНА И.В., БЛИНОВ А.О. *Эволюция и перспективы приближенных методов оптимального управления* // Программные системы: теория и приложения. – 2001. – №2(6). – С. 11–29.
2. ПАНОВСКИЙ В.Н. *Прикладное применение интервального метода взрывов* // Электронный журнал «Труды МАИ». –

2014. – №73. – [Электронный ресурс] – URL: <http://www.mai.ru/science/trudy/published.php?ID=48451> (дата обращения: 19.03.2016.)
3. ПАНОВСКИЙ В.Н. *Применение аппарата интервального анализа для поиска глобального экстремума функций* // Электронный журнал «Труды МАИ». – 2012. – №51. – [Электронный ресурс] – URL: <http://www.mai.ru/science/trudy/published.php?ID=28948> (дата обращения: 19.03.2016.)
 4. ПАНТЕЛЕЕВ А.В. *Вариационное исчисление в примерах и задачах.* – М.: Высшая школа, 2006. – 272 с.
 5. ПАНТЕЛЕЕВ А.В., МЕТЛИЦКАЯ Д.В., АЛЕШИНА Е.А. *Методы глобальной оптимизации. Метаэвристические стратегии и алгоритмы.* – М.: Вузовская книга, 2013. – 244 с.
 6. ПАНТЕЛЕЕВ А.В. *Применение эволюционных методов глобальной оптимизации в задачах оптимального управления детерминированными системами.* – М.: Изд-во МАИ, 2013. – 160 с.
 7. ФЕДОРЕНКО Р.П. *Приближенное решение задач оптимального управления.* – М.: Наука, 1978. – 488 с.
 8. ШАРЫЙ С.П. *Конечномерный интервальный анализ.* – Новосибирск: XYZ, 2010. – 606 с.
 9. FLOUDAS C.A., PARDALOS P.M. *Encyclopedia of Optimization.* – London: Springer, 2009. – 4626 p.
 10. JAULIN L., KIEFFER M., DIDRIT O., et al. *Applied interval analysis.* – London: Springer-Verlag, 2001. – 391 p.
 11. HANSEN E. *Global optimization using interval analysis.* – New York: Marcel Dekker, 2004. – 492 p.
 12. MICHALEWICZ Z., FOGEL D. *How to solve it: Modern Heuristics.* – London: Springer, 2004. – 554 p.
 13. MOORE R.E. *Interval analysis.* – Englewood Cliffs: Prentice Hall, 1966. – 145 p.
 14. MOORE R.E. *Methods and applications of interval analysis.* – Philadelphia: SIAM, 1979. – 152 p.
 15. RATSCHЕК H., ROKNE J. *New computer methods for global optimization.* – Chichester: Horwood, 1988. – 236 p.

16. SHARY S.P. *A surprising approach in interval global optimization* // *Reliable Computing*. – 2001. – P. 497–505.
17. SHARY S.P. *Randomized algorithms in interval global optimization* // *Numerical Analysis and Applications*. – 2008. – Vol. 1. – P. 376–389.
18. TEWARI A. *Advanced control of aircraft, spacecraft and rockets*. – New York: A John Wiley & Sons, 2011. – 390 p.

DEVELOPMENT OF METAHEURISTIC INTERVAL MINIMIZATION METHODS FOR THE OPTIMAL PROGRAM CONTROL SYNTHESIS

Andrei Panteleev, Moscow Aviation Institute, Moscow, Doctor of Science, professor (avpanteleev@inbox.ru).

Valentin Panovskiy, Moscow Aviation Institute, Moscow, post-graduate student (panovskiy.v@yandex.ru).

Abstract: We propose metaheuristic interval methods of global constrained optimization and their software implementation. Three methods are considered: average path ending, stochastic grid and interval scatter search. The methods are applied to the problem of optimal program control of nonlinear deterministic discrete and continuous systems. The solution of the applied problem of 3-dimensional interception and a brief comparative analysis of the effectiveness are given.

Keywords: interval methods, metaheuristic, global constrained optimization, optimal control.

Статья представлена к публикации членом редакционной коллегии П.С. Щербаковым.

*Поступила в редакцию 29.10.2015.
Опубликована 31.03.2016.*