

ЭВОЛЮЦИОННЫЕ ВЫЧИСЛЕНИЯ ДЛЯ РЕШЕНИЯ ЗАДАЧИ ТЕРМИНАЛЬНОГО ОПТИМАЛЬНОГО УПРАВЛЕНИЯ

А.И. Дивеев

aidiveev@mail.ru

ФИЦ ИУ РАН, Москва, Российская Федерация

Аннотация

Рассмотрена проблема численного решения задачи терминального оптимального управления. Приведена общая постановка задачи терминального оптимального управления и краткий обзор методов ее решения. При прямом подходе и редукции задачи оптимального управления к задаче конечномерной оптимизации целевой функционал на пространстве искомых параметров независимо от вида аппроксимации функции управления может не обладать свойством унимодальности. Поэтому для решения задачи целесообразно использовать эволюционные алгоритмы. Представлен общий подход к решению задачи терминального оптимального управления эволюционными вычислительными алгоритмами. Представлено описание некоторых эволюционных алгоритмов, которые были отобраны как наиболее эффективные для решения задачи оптимального управления. Рассмотрен гибридный эволюционный алгоритм, построенный на основе комбинации нескольких эволюционных алгоритмов. В вычислительном эксперименте рассмотрены задачи терминального оптимального управления, для которых найдены оптимальные решения известными классическими численными методами, использующие при поиске градиент целевого функционала. Сравнение результатов, полученных классическими и эволюционными методами по значениям функционала и вычислительным затратам, позволяет сделать вывод, что эволюционные алгоритмы способны эффективно решать задачи терминального оптимального управления

Ключевые слова

Оптимальное управление, терминальное управление, эволюционные вычисления, кусочно-линейная аппроксимация, генетический алгоритм, гибридный эволюционный алгоритм, алгоритм «роя — частиц»

Поступила 21.10.2022

Принята 09.11.2022

© Автор(ы), 2023

Введение. Современные метаэвристические алгоритмы, которые часто называют эволюционными или популяционными [1], все чаще применяют для решения сложных задач оптимизации. Их преимущество — простота

реализации и очевидная зависимость процесса поиска оптимального решения от значений управляющих параметров алгоритмов. Все эволюционные алгоритмы используют множество возможных решений, которое, как правило, сначала генерируется случайно и называется популяцией. Затем в зависимости от значений целевой функции для каждого возможного решения выполняется трансформация этих возможных решений. Операции изменения возможных решений часто называют эволюционными преобразованиями. В большинстве случаев эти изменения возможных решений, согласно утверждениям авторов эволюционных алгоритмов, построены на основе модели поведения некоторых живых организмов, которые и определяют название алгоритма. Поскольку эволюционные алгоритмы используют в основном прикладные математики и инженеры, не являющиеся специалистами в области биологии, названия алгоритмов, порой очень неожиданные, дополнительных дискуссий не вызывают, но придают некую «романтику» этой области точных наук.

Основной недостаток эволюционных алгоритмов — отсутствие строгого доказательства их сходимости. Эволюционные алгоритмы находят решения, близкие к оптимальному, как правило, разные запуски алгоритмов могут привести к различным решениям, незначительно отличающимся по значению целевой функции. Применение эволюционных алгоритмов для решения задач управления стало называться машинным обучением управления [2, 3], чтобы по аналогии с искусственными нейронными сетями, где также для обучения применяются оптимизационные алгоритмы, избежать обоснования получения не оптимального, а близкого к оптимальному решения. Хотя, скорее всего, термин «машинное обучение» — исторически обоснованный этап научного развития, предвещающий открытия в области искусственного интеллекта.

Сформулированная в 1950-е годы задача оптимального управления [4] является вычислительно сложной. Попытки создания универсального численного метода для ее приближенного решения [5] вследствие различия функционалов, нелинейных моделей объектов управления, терминальных и начальных условий не увенчались успехом. Наиболее продвинутым оказался математический пакет OPTCON-A [6–8], который представляет собой совокупность классических методов нелинейного программирования, используемых при редукции задачи оптимального управления к задаче конечномерной оптимизации.

Из предположения, что задачи оптимального управления для мобильных роботов, движущихся по пространству с препятствиями, являются

не унимодальными, так как обход препятствия возможен с разных сторон, возникла идея использовать для решения таких задач методы глобальной оптимизации, в том числе эволюционные алгоритмы. Масштабное исследование различных эволюционных алгоритмов для задачи оптимального управления мобильным роботом и их сравнение с классическими методами нелинейного программирования с применением мультистарта так, чтобы число вычислений значения функционала для эволюционных алгоритмов и классических было приблизительно одинаковым, представлено в [9]. Среди классических алгоритмов также рассмотрен алгоритм стохастического градиента Adam, который сейчас широко используется для обучения искусственных нейронных сетей. Во всех случаях эволюционные алгоритмы показали лучшие результаты, чем классические алгоритмы как по средней величине функционала, вычисленной по значениям, которые получены в нескольких экспериментах, так и по среднеквадратическому отклонению этих значений. Отметим, что алгоритм Adam показал лучшие результаты среди классических градиентных методов. Результаты, представленные в [9], дали повод для широкого исследования применения эволюционных алгоритмов при решении задач оптимального управления. Исследования эволюционных алгоритмов для численного решения задач оптимального управления с фазовыми ограничениями продолжены в [10, 11]. Сравнение численных решений задачи оптимального управления эволюционными алгоритмами на основе прямого подхода и применения принципа максимума Понтрягина приведено в [12]. Сравнение эффективности решений задачи оптимального управления эволюционными алгоритмами с классическими алгоритмами случайного поиска рассмотрено в [13]. Во всех исследованиях эволюционные алгоритмы показали более эффективные и стабильные результаты при решении задач оптимального управления. В результате эволюционные алгоритмы используют для получения множества экстремалей для решения задачи общего синтеза управления на основе машинного обучения управления с учителем методами символьной регрессии [14].

В настоящей работе продолжены исследования применения эволюционных алгоритмов и их сравнение с классическими методами при решении задачи терминального оптимального управления.

Задача терминального оптимального управления. Рассмотрим постановку задачи терминального оптимального управления. Задана математическая модель объекта управления

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

где $\mathbf{x} = [x_1 \dots x_n]^T \in \mathfrak{R}^n$ — вектор состояний; $\mathbf{u} = [u_1 \dots u_m]^T$ — вектор управления, $\mathbf{u} \in U \subseteq \mathfrak{R}^m$, U — компактное множество, которое определяет ограничения на значения управления.

Задано начальное состояние объекта управления

$$\mathbf{x}(0) = \mathbf{x}^0 = [x_1^0 \dots x_n^0]^T \quad (2)$$

и время процесса управления t_f .

В качестве критерия качества может быть задан интегральный функционал. Тогда должны быть заданы терминальное состояние объекта управления

$$\mathbf{x}(t_f) = \mathbf{x}^f = [x_1^f \dots x_n^f]^T \quad (3)$$

и интегральный функционал

$$J_1 = \int_0^{t_f} f_0(\mathbf{x}, \mathbf{u}) dt \rightarrow \min_{\mathbf{u}}. \quad (4)$$

В качестве критерия качества вместо (4) могут быть заданы терминальные условия

$$J_2 = F(\mathbf{x}(t_f)) \rightarrow \min_{\mathbf{u}}. \quad (5)$$

Рассмотрим общую схему численного решения задач (1)–(4) или (1), (2), (5) на основе прямого подхода и кусочно-линейной аппроксимации функции управления. Определим интервал времени Δt и разобьем ось времени на K отрезков:

$$K = \left\lfloor \frac{t_f}{\Delta t} \right\rfloor. \quad (6)$$

Для каждой компоненты управления определяем $K+1$ значений параметров на границах интервалов. Значения компонент управления вычисляем по формуле

$$u_i(t) = \begin{cases} u_i^+, & \text{если } \tilde{u}_i(t) \geq u_i^+; \\ u_i^-, & \text{если } \tilde{u}_i(t) \leq u_i^-; \\ \tilde{u}_i(t) & \text{иначе,} \end{cases} \quad i = 1, \dots, m;$$

где

$$\tilde{u}_i(t) = \frac{q_{j+1+(i-1)(K+1)} - q_{j+(i-1)(K+1)}}{\Delta t} (t - (j-1)\Delta t) + q_{j+(i-1)(K+1)}, \\ (j-1)\Delta t \leq t < j\Delta t, \quad j = 1, \dots, K, \quad i = 1, \dots, m.$$

Для решения задачи эволюционные алгоритмы должны найти вектор постоянных параметров

$$\mathbf{q} = [q_1 \dots q_{(K+1)m}]^T$$

для минимизации критерия $J_2(\mathbf{q})$ (5) или критерия (4) с включенным в него условием попадания в терминальное состояние (3):

$$J_3(\mathbf{q}) = p_1 \|\mathbf{x}^f - \mathbf{x}(t_f)\| + \int_0^{t_f} f_0(\mathbf{x}, \mathbf{u}) dt \rightarrow \min_{\mathbf{u}} \quad (7)$$

где p_1 — коэффициент штрафа за неточность попадания в терминальное состояние.

Эволюционные алгоритмы. В качестве эволюционного алгоритма используем гибридный алгоритм [15, 16], который включает в себя комбинацию трех эволюционных алгоритмов: 1) генетический алгоритм (GA) [17], 2) алгоритм «роя — частиц» (particle swarm optimization, PSO) [18], 3) алгоритм «серого волка» (Grey Wolf Optimizer, GWO) [19].

Рассмотрим этапы гибридного алгоритма. Генерируем популяцию возможных решений

$$Q = \{\mathbf{q}^1, \dots, \mathbf{q}^H\}, \quad (8)$$

где $\mathbf{q}^j = [q_1^j \dots q_p^j]^T$, $j = 1, \dots, H$, $q_i^j = \xi(q_i^+ - q_i^-) + q_i^-$, ξ — случайное число от 0 до 1, порождаемое генератором случайных чисел, $i = 1, \dots, p$.

Генерируем множество скоростей изменений возможных решений, которое на первом этапе состоит из нулевых векторов размерностью p :

$$V = \{\mathbf{v}^1 = [0 \dots 0]^T, \dots, \mathbf{v}^H = [0 \dots 0]^T\}.$$

Вычисляем значение целевой функции для каждого возможного решения из (8):

$$f_j = J(\mathbf{q}^j), \quad j = 1, \dots, H.$$

Находим лучшее возможное решение

$$f_{j^-} = J(\mathbf{q}^{j^-}) = \min_j \{f_j : j = 1, \dots, H\}.$$

Выбираем случайно тип эволюции $\gamma \in \{0, 1, 2\}$. Если $\gamma = 0$, то выполняем эволюцию возможного решения на основе алгоритма GWO. Отбираем L лучших на данный момент возможных решений

$$\{\mathbf{q}^{j^L}, \dots, \mathbf{q}^{j^L}\} \quad (9)$$

в соответствии со значением целевого функционала

$$J(\mathbf{q}^{j1}) \leq J(\mathbf{q}^{j2}) \leq \dots \leq J(\mathbf{q}^{jL}). \quad (10)$$

В классическом алгоритме GWO $L = 3$ [19]. Здесь предусматриваем выбор любого числа лучших решений.

Далее случайно отбираем какое-либо возможное решение $\mathbf{q}^j = [q_1^j \dots q_p^j]^T$ и модифицируем его каждую компоненту согласно алгоритму GWO. В результате получаем компоненты нового вектора

$$q_i^{H+1} = \frac{1}{L} \sum_{k=1}^L \left(q_i^{jk} - \left(2 - \frac{2g}{G} \right) (2\xi - 1) | 2\xi q_i^{jk} - q_i^j | \right), \quad (11)$$

где g — номер текущего поколения; G — общее число поколений, $i = 1, \dots, p$.

Если $\gamma = 1$, то модифицируем возможное решение согласно генетическому алгоритму. Отбираем два возможных решения-«родителя»:

$$\mathbf{q}^\alpha = [q_1^\alpha \dots q_p^\alpha]^T, \quad \mathbf{q}^\beta = [q_1^\beta \dots q_p^\beta]^T. \quad (12)$$

Находим для родителей компоненту скрещивания $c \in \{1, \dots, p\}$. Переводим вещественные значения этих компонент в коды Грея с заданным числом бит d под целую и дробную части:

$$\mathbf{y}^1 = [y_1^1 \dots y_d^1]^T, \quad \mathbf{y}^2 = [y_1^2 \dots y_d^2]^T. \quad (13)$$

Здесь

$$\mathbf{y}^1 = \text{Grey}(q_c^\alpha), \quad \mathbf{y}^2 = \text{Grey}(q_c^\beta) \quad (14)$$

и функция $\text{Grey}(a)$ преобразует вещественное число a в вектор двоичного кода Грея. При преобразовании вещественных чисел в коды Грея первоначально преобразуем число в двоичный код, а затем, используя операцию исключающего ИЛИ, преобразуем двоичный код в код Грея. При этом если вещественное число отрицательно, то двоичный код заменим его дополнением.

Находим точку скрещивания для кодов Грея $s \in \{1, \dots, d\}$ и обмениваем «хвосты» кодов после точки скрещивания. Получаем два новых кода Грея:

$$\mathbf{y}^3 = [y_1^1 \dots y_s^1 \ y_{s+1}^2 \dots y_d^2]^T, \quad \mathbf{y}^4 = [y_1^2 \dots y_s^2 \ y_{s+1}^1 \dots y_d^1]^T. \quad (15)$$

Далее переводим новые коды в вещественные числа:

$$q_c^{H+1} = \text{Grey}^{-1}(\mathbf{y}^3), \quad q_c^{H+2} = \text{Grey}^{-1}(\mathbf{y}^4). \quad (16)$$

Обмениваем остальные компоненты отобранных векторов родителей, получаем два новых возможных решения:

$$\begin{aligned} \mathbf{q}^{H+1} &= [q_1^\alpha \ \dots \ q_{c-1}^\alpha \ q_c^{H+1} \ q_{c+1}^\beta \ \dots \ q_p^\beta]^\top, \\ \mathbf{q}^{H+2} &= [q_1^\beta \ \dots \ q_{c-1}^\beta \ q_c^{H+2} \ q_{c+1}^\alpha \ \dots \ q_p^\alpha]^\top. \end{aligned} \quad (17)$$

Если $\gamma = 2$, то модифицируем возможное решение согласно алгоритму PSO. Находим лучшее решение среди D случайно отобранных возможных решений:

$$\begin{aligned} \mathbf{q}^* &= [q_1^* \ \dots \ q_p^*]^\top, \\ J(\mathbf{q}^*) &= \min_i \{J(\mathbf{q}^{j_i}) : i = 1, \dots, D\}. \end{aligned} \quad (18)$$

Случайно отбираем возможное решение $\mathbf{q}^j = [q_1^j \ \dots \ q_p^j]^\top$. Корректируем соответствующую скорость изменения отобранного возможного решения

$$v_i^j \leftarrow k_\alpha v_i^j + \xi k_\beta (q_i^* - q_i^j) + \xi k_\gamma (q_i^{j-} - q_i^j), \quad i = 1, \dots, p, \quad (20)$$

где k_α , k_β , k_γ — постоянные заданные коэффициенты алгоритма.

Далее модифицируем все компоненты отобранного вектора и получаем компоненты нового вектора возможных решений:

$$q_i^{H+1} = q_i^j + k_\sigma v_i^j, \quad i = 1, \dots, p. \quad (21)$$

Здесь k_σ — заданный коэффициент алгоритма.

Для включения нового полученного вектора \mathbf{q}^{H+1} в популяцию возможных решений, а в случае генетической трансформации — еще и вектора \mathbf{q}^{H+2} , вычисляем значения функционала для новых полученных решений:

$$f_{H+1} = J(\mathbf{q}^{H+1}), \quad f_{H+2} = J(\mathbf{q}^{H+2}). \quad (22)$$

По значениям функционала для всех возможных решений находим наилучшее решение \mathbf{q}^{j^+} :

$$J(\mathbf{q}^{j^+}) = \max_j \{f_j : j = 1, \dots, H\}. \quad (23)$$

Если значение функционала для нового возможного решения лучше, чем наилучшее значение $f_{H+1} \leq J(\mathbf{q}^{j^+})$, то заменяем наилучшее возможное решение новым полученным возможным решением:

$$\mathbf{q}^{j+} \leftarrow \mathbf{q}^{H+1}, f_{j+} \leftarrow f_{H+1}. \quad (24)$$

То же выполняем и для нового возможного решения \mathbf{q}^{H+2} .

Повторяем шаги (9)–(24) заданное число раз G . Затем находим лучшее возможное решение, которое полагаем решением задачи оптимального управления.

Вычислительный эксперимент. Рассмотрим следующую задачу оптимального управления [8]:

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= u - \sin x_1, \\ \mathbf{x}(0) &= \mathbf{x}^0 = [5 \ 0]^T, \\ t_f &= 5, \\ -1 &\leq u \leq 1, \\ J_4 &= x_1^2(t_f) + x_2^2(t_f) \rightarrow \min_u. \end{aligned}$$

Согласно [8], функционал имеет два локальных минимума. Оптимальное управление представляет собой кусочно-постоянную функцию

$$u^* = \begin{cases} 1, & \text{если } t \in [0, 0; 0, 98244286) \cup [4, 55036911; 5, 0); \\ -1, & \text{если } t \in [0, 982442286; 4, 55036911). \end{cases} \quad (25)$$

Оптимальное значение функционала составляет $J_4 = 11,908013767$. Решение получено с использованием пакета OPTCON-A с применением метода интегрирования 8-го порядка точности.

Решим эту задачу эволюционным гибридным алгоритмом на основании кусочно-линейной аппроксимации функции управления. Введем интервал времени $\Delta t = 0,05$. В результате из (6) получим $K = 100$ интервалов возможного изменения управления и размерность вектора искомых параметров $p = K + 1 = 101$. Зададим ограничения на параметры, $q^+ = 4$, $q^- = -4$. Параметры алгоритма: число возможных решений в популяции $H = 1024$, число поколений $G = 512$, число эволюционных преобразований в одном поколении $R = 512$, число бит в двоичном коде Грея для генетического алгоритма $d = 16$, число лучших решений в алгоритме GWO $L = 4$, параметры для алгоритма PSO $k_\alpha = 0,729$, $k_\beta = 0,85$, $k_\gamma = 0,15$, $k_\sigma = 1,0$, $D = 4$. Шаг интегрирования составлял $\Delta = 0,0001$. В результате алгоритм нашел функцию управления, график которой представлен на рис. 1.

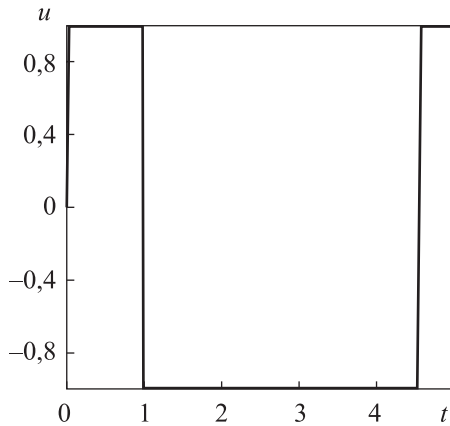


Рис. 1. График функции оптимального управления

Найденное управление практически совпало с оптимальным управлением (25). Значение функционала составило $J_4 = 11,90551$. Время вычислений на персональном компьютере с процессором Intel Core I7 2.8 GHz составило не более 25 мин.

Рассмотрим задачу оптимального управления электроприводом [7]:

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= u - x_2^2, \\ \mathbf{x}(0) &= \mathbf{x}^0 = [0 \ 0]^T, \end{aligned}$$

$$t_f = 4,85,$$

$$\mathbf{x}(t_f) = \mathbf{x}^f = [3,12 \ 0]^T, \tag{26}$$

$$J_5 = \int_0^{t_f} u^2 dt \rightarrow \min_u. \tag{27}$$

Оптимальное значение функционала, найденное в [7], составило $J_5 = 2,46546$. Решим задачу эволюционным алгоритмом. Введем временной интервал $\Delta t = 0,25$. В результате для решения задачи необходимо найти $p = \lfloor 4,85 / 0,25 \rfloor + 1 = 20$ значений параметров. Согласно схеме решения (7), добавим к интегральному функционалу (27) условия попадания в терминальное состояние (26):

$$J_6 = p_1 \|\mathbf{x}^f - \mathbf{x}(t_f)\| + \int_0^{t_f} u^2 dt \rightarrow \min_u, \tag{28}$$

где $p_1 = 25$.

Установим ограничения на параметры $q^+ = 2$, $q^- = -2$. Остальные параметры эволюционного алгоритма те же, что и в предыдущем случае. Эволюционный алгоритм нашел следующее решение:

$$\begin{aligned} \mathbf{q} = & [1,24357 \ 0,93099 \ 0,82572 \ 0,79621 \ 0,72031 \ 0,78965 \ 0,74485 \\ & 0,71048 \ 0,72261 \ 0,60295 \ 0,71005 \ 0,56024 \ 0,77832 \ 0,61792 \\ & 0,49907 \ 0,32047 \ 0,04373 \ -0,23830 \ -0,69729 \ -1,17350]^T. \end{aligned} \tag{29}$$

Для этого решения значение функционала (28) составило $J_6 = 2,4659$. Точность попадания в терминальное состояние 10^{-6} . Графики переменных и функции управления для полученного решения (29) приведены на рис. 2.

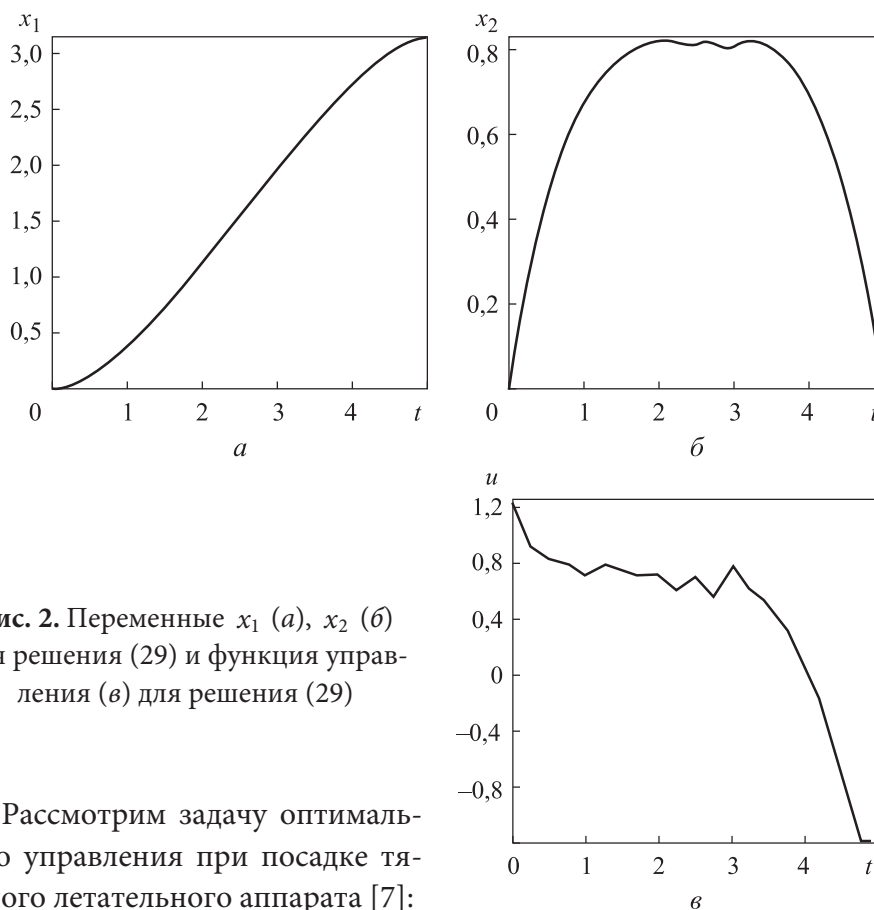


Рис. 2. Переменные x_1 (а), x_2 (б) для решения (29) и функция управления (в) для решения (29)

Рассмотрим задачу оптимального управления при посадке тяжелого летательного аппарата [7]:

$$\dot{x}_1 = \frac{\cos x_3}{ac(u)x_2^2},$$

$$\varepsilon \dot{x}_2 = -\frac{1}{x_2} - \frac{\sin x_3}{ac(u)x_2^3},$$

$$\varepsilon \dot{x}_3 = \frac{u}{c(u)x_2^2} - \frac{\cos x_3}{ac(u)x_2^4},$$

где $a = 1,6208$; $c(u) = 0,01 + 0,3u^2$; $\varepsilon = 0,01$; $t_f = 0,2185$;

$$\mathbf{x}(0) = [0 \ 2,19901 \ 0]^T;$$

$$0,08 \leq u \leq 0,417;$$

$$x_2(t_f) = x_2^f = 1,009905; \tag{30}$$

$$x_3(t_f) = x_3^f = 0; \tag{31}$$

$$J_7 = -x_1(t_f) \rightarrow \min_u. \tag{32}$$

Согласно [7], найденное оптимальное значение функционала (30) $J_7 = -1,98554$.

Решим эту задачу эволюционным алгоритмом. Введем интервал времени $\Delta t = 0,015$. В результате необходимо найти $p = 16$ параметров. Добавим в функционал (32) точность выполнения терминальных условий (30), (31):

$$J_8 = -x_1(t_f) + p_1 \sqrt{\left(x_2^f - x_2(t_f)\right)^2 + \left(x_3^f - x_3(t_f)\right)^2} \rightarrow \min_u$$

со значением $p_1 = 10$.

Установим ограничения для искомым параметров $q^+ = 0,8$, $q^- = 0$.

Эволюционный алгоритм нашел следующее решение:

$$\mathbf{q} = [0,13642 \ 0,19037 \ 0,18947 \ 0,18798 \ 0,18847 \ 0,18997 \ 0,17812 \ 0,18854 \ 0,18882 \ 0,17598 \ 0,18051 \ 0,18569 \ 0,24834 \ 0,14323 \ 0,23160 \ 0,57793]^T. \tag{33}$$

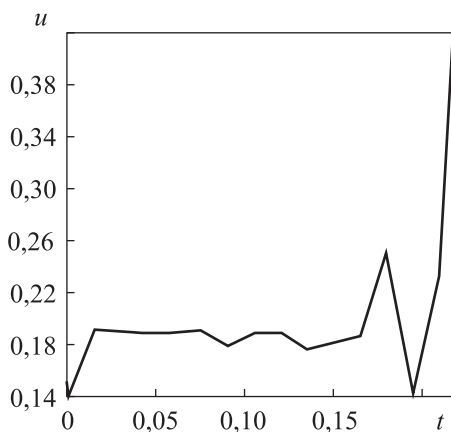


Рис. 3. Функция управления для решения (33)

Оптимальное значение функционала для найденного решения (33) $J_8 = -1,9689$. Точность попадания в терминальное состояние 10^{-5} . График функции управления для решения (33) приведен на рис. 3.

Заключение. Согласно результатам вычислительных экспериментов, полученные эволюционным алгоритмом решения незначительно уступают оптимальным по значению целевого функционала. Различие графиков функций управления вызвано выбором кусочно-линейной ап-

проксимации. Поскольку в задачах не оговаривался характер функции управления, ее кусочно-линейная аппроксимация выбрана как наиболее универсальная. Если бы требовалось искать функцию управления в классе непрерывно-дифференцируемых функций, то аппроксимация функции управления выполнялась бы в классе гладких полиномов. Сделан вывод, что эволюционные алгоритмы решают приближенно сложные задачи оптимального управления с приемлемым значением целевого функционала.

ЛИТЕРАТУРА

- [1] Карпенко А.П. Современные алгоритмы поисковой оптимизации. М., Изд-во МГТУ им. Н.Э. Баумана, 2014.
- [2] Duriez T., Brunton S.L., Noack B.R. Machine learning control — taming nonlinear dynamics and turbulence. Vol. 116. Cham, Springer, 2017.
DOI: <https://doi.org/10.1007/978-3-319-40624-4>
- [3] Diveev A., Shmalko E. Machine learning control by symbolic regression. Cham, Springer, 2021. DOI: <https://doi.org/10.1007/978-3-030-83213-1>
- [4] Понтрягин Л.С., Болтянский В.Г., Гамкрелидзе Р.В. и др. Математическая теория оптимальных процессов. М., Наука, 1983.
- [5] Федоренко Р.П. Приближенное решение задачи оптимального управления. М., Наука, 1978.
- [6] Горнов А.Ю., Тятюшкин А.И., Финкельштейн Е.А. Численные методы решения прикладных задач оптимального управления. *Журнал вычислительной математики и математической физики*, 2013, т. 53, № 12, с. 2014–2028.
DOI: <https://doi.org/10.7868/S0044466913120077>
- [7] Горнов А.Ю., Тятюшкин А.И., Финкельштейн Е.А. Численные методы для решения терминальных задач оптимального управления. *Журнал вычислительной математики и математической физики*, 2016, т. 56, № 2, с. 224–237.
DOI: <https://doi.org/10.7868/S0044466916020095>
- [8] Андрианов А.Н., Аникин А.С., Горнов А.Ю. Численное исследование задач оптимизации больших размерностей с использованием модификации метода Б.Т. Поляка. *Журнал вычислительной математики и математической физики*, 2021, т. 61, № 7, с. 1059–1069. DOI: <https://doi.org/10.31857/S0044466921070036>
- [9] Дивеев А.И., Константинов С.В. Исследование практической сходимости эволюционных алгоритмов оптимального программного управления колесным роботом. *Известия РАН. Теория и системы управления*, 2018, № 4, т. 57, с. 80–106.
DOI: <https://doi.org/10.31857/S000233880002513-3>
- [10] Diveev A.I., Sofronova E.A., Konstantinov S.V. Approaches to numerical solution of optimal control problem using evolutionary computations. *Appl. Sci.*, 2021, vol. 11, iss. 15, art. 7096. DOI: <https://doi.org/10.3390/app11157096>

[11] Дивеев А.И., Константинов С.В. Эволюционные алгоритмы для решения задачи оптимального управления. *Вестник РУДН. Сер. Инженерные исследования*, 2017, т. 18, № 2, с. 254–265.

DOI: <https://doi.org/10.22363/2312-8143-2017-18-2-254-265>

[12] Diveev A., Shmalko E. Comparison of direct and indirect approaches for numerical solution of the optimal control problem by evolutionary methods. In: Jaćimović M., Khachay M., Malkova V., Posypkin M. (eds). *Optimization and Applications. OPTIMA 2019*. Cham, Springer, 2020, pp. 180–193.

DOI: https://doi.org/10.1007/978-3-030-38603-0_14

[13] Diveev A., Konstantinov S.V., Balandina G.I., et al. Comparative research of random search algorithms and evolutionary algorithms for the optimal control problem of the mobile robot. *Procedia Comput. Sci.*, 2019, vol. 150, pp. 462–470.

DOI: <https://doi.org/10.1016/j.procs.2019.02.080>

[14] Konstantinov S.V., Diveev A.I. Solving the problem of the optimal control system general synthesis based on approximation of a set of extremals using the symbol regression method. *Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение*, 2020, № 2 (131), с. 59–74. DOI: <http://dx.doi.org/10.18698/0236-3933-2020-2-59-74>

[15] Дивеев А.И. Гибридный эволюционный алгоритм для решения задачи оптимального управления. *Вопросы теории безопасности и устойчивости систем*, 2021, № 23, с. 3–12.

[16] Diveev A. Hybrid evolutionary algorithm for optimal control problem. In: Arai K. (eds). *Intelligent Systems and Applications. IntelliSys 2022. Lecture Notes in Networks and Systems*, vol. 543. Cham, Springer, 2023, pp. 726–738.

DOI: https://doi.org/10.1007/978-3-031-16078-3_50

[17] Davis L. Handbook of genetic algorithms. New York, Van Nostrand Reinhold, 1991.

[18] Eberhart R.K.E. Particle swarm optimization. *Proc. ICNN'95*, 1995, pp. 1942–1948.

DOI: <https://doi.org/10.1109/ICNN.1995.488968>

[19] Mirjalili S., Mirjalili S.M., Lewis A. Grey Wolf Optimizer. *Adv. Eng. Softw.*, 2014, vol. 69, pp. 46–61. DOI: <https://doi.org/10.1016/j.advengsoft.2013.12.007>

Дивеев Асхат Ибрагимович — д-р техн. наук, профессор, главный научный сотрудник ФИЦ ИУ РАН (Российская Федерация, 119333, Москва, ул. Вавилова, д. 44, корп. 2).

Просьба ссылаться на эту статью следующим образом:

Дивеев А.И. Эволюционные вычисления для решения задачи терминального оптимального управления. *Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение*, 2023, № 1 (142), с. 44–59.

DOI: <https://doi.org/10.18698/0236-3933-2023-1-44-59>

EVOLUTIONARY COMPUTATION FOR SOLVING THE TERMINAL OPTIMAL CONTROL PROBLEM

A.I. Diveev

aidiveev@mail.ru

Federal Research Center “Computer Science and Control”,
Russian Academy of Sciences, Moscow, Russian Federation

Abstract

The present article considers the problem of numerical solution of the terminal optimal control problem. The general statement of the terminal optimal control problem and a brief overview of its solving methods are presented. With a direct approach and reduction of the optimal control problem to the finite-dimensional optimization problem, the target functional on the space of desired parameters, regardless of the type of approximation of the control function, may not have the unimodal property. Therefore, it is advisable to use evolutionary algorithms to solve the problem. A general approach to solving the terminal optimal control problem of evolutionary computational algorithms is presented. The paper presents a description of some evolutionary algorithms that were selected as the most effective for solving the optimal control problem. A hybrid evolutionary algorithm based on a combination of several evolutionary algorithms is considered. The computational experiment considers the terminal optimal control problems, for which optimal solutions were found by known classical numerical methods that use the gradient of the target functionality when searching. Comparison of the results obtained by classical and evolutionary methods by functional values and computational costs allows us to conclude that evolutionary algorithms are able to effectively solve the terminal optimal control problems

Keywords

Optimal control, terminal control, evolutionary computation, piecewise linear approximation, genetic algorithm, hybrid algorithm, particle swarm optimization algorithm

Received 21.10.2022

Accepted 09.11.2022

© Author(s), 2023

REFERENCES

- [1] Karpenko A.P. *Sovremennye algoritmy poiskovoy optimizatsii* [Modern algorithms of search engine optimization]. Moscow, BMSTU Publ., 2014.
- [2] Duriez T., Brunton S.L., Noack B.R. *Machine learning control — taming nonlinear dynamics and turbulence*. Vol. 116. Cham, Springer, 2017.
DOI: <https://doi.org/10.1007/978-3-319-40624-4>

- [3] Diveev A., Shmalko E. Machine learning control by symbolic regression. Cham, Springer, 2021. DOI: <https://doi.org/10.1007/978-3-030-83213-1>
- [4] Pontryagin L.S., Boltyanskiy V.G., Gamkrelidze R.V., et al. Matematicheskaya teoriya optimalnykh protsessov [Mathematical theory of optimal processes]. Moscow, Nauka Publ., 1983.
- [5] Fedorenko R.P. Priblizhennoe reshenie zadachi optimalnogo upravleniya [Approximate solution of an optimal control problem]. Moscow, Nauka Publ., 1978.
- [6] Gornov A.Yu., Tyatyushkin A.I., Finkelshteyn E.A. Numerical methods for solving applied optimal control problems. *Comput. Math. and Math. Phys.*, 2013, vol. 53, no. 12, pp. 1825–1838. DOI: <https://doi.org/10.1134/S0965542513120063>
- [7] Gornov A.Yu., Tyatyushkin A.I., Finkelshteyn E.A. Numerical methods for solving terminal optimal control problems. *Comput. Math. and Math. Phys.*, 2016, vol. 56, no. 2, pp. 221–234. DOI: <https://doi.org/10.1134/S0965542516020093>
- [8] Andrianov A.N., Anikin A.S., Gornov A.Yu. Numerical study of high-dimensional optimization problems using a modification of Polyak's method. *J. Comput. Syst. Sci. Int.*, 2018, vol. 57, no. 4, pp. 561–580. DOI: <https://doi.org/10.1134/S106423071804007X>
- [9] Diveev A.I., Konstantinov S.V. Study of the practical convergence of evolutionary algorithms for the optimal program control of a wheeled robot. *J. Comput. Syst. Sci. Int.*, 2018, vol. 57, no. 4, pp. 561–580. DOI: <https://doi.org/10.1134/S106423071804007X>
- [10] Diveev A.I., Sofronova E.A., Konstantinov S.V. Approaches to numerical solution of optimal control problem using evolutionary computations. *Appl. Sci.*, 2021, vol. 11, iss. 15, art. 7096 (in Russ.). DOI: <https://doi.org/10.3390/app11157096>
- [11] Diveev A.I., Konstantinov S.V. Evolutionary algorithms for the problem of optimal control. *Vestnik RUDN. Ser. Inzhenernyye issledovaniya* [RUDN Journal of Engineering Research], 2017, vol. 18, no. 2, pp. 254–265 (in Russ.). DOI: <https://doi.org/10.22363/2312-8143-2017-18-2-254-265>
- [12] Diveev A., Shmalko E. Comparison of direct and indirect approaches for numerical solution of the optimal control problem by evolutionary methods. In: Jaćimović M., Khachay M., Malkova V., Posypkin M. (eds). *Optimization and Applications. OPTIMA 2019*. Cham, Springer, 2020, pp. 180–193. DOI: https://doi.org/10.1007/978-3-030-38603-0_14
- [13] Diveev A., Konstantinov S.V., Balandina G.I., et al. Comparative research of random search algorithms and evolutionary algorithms for the optimal control problem of the mobile robot. *Procedia Comput. Sci.*, 2019, vol. 150, pp. 462–470. DOI: <https://doi.org/10.1016/j.procs.2019.02.080>
- [14] Konstantinov S.V., Diveev A.I. Solving the problem of the optimal control system general synthesis based on approximation of a set of extremals using the symbol regression method. *Herald of the Bauman Moscow State Technical University, Series Instrument Engineering*, 2020, no. 2 (131), pp. 59–74. DOI: <http://dx.doi.org/10.18698/0236-3933-2020-2-59-74>

- [15] Diveev A.I. Hybrid evolutionary algorithm for optimal control. *Voprosy teorii bezopasnosti i ustoychivosti sistem*, 2021, no. 23, pp. 3–12.
- [16] Diveev A. Hybrid evolutionary algorithm for optimal control problem. In: Arai K. (eds). *Intelligent Systems and Applications. IntelliSys 2022. Lecture Notes in Networks and Systems*, vol. 543. Cham, Springer, 2023, pp. 726–738.
DOI: https://doi.org/10.1007/978-3-031-16078-3_50
- [17] Davis L. Handbook of genetic algorithms. New York, Van Nostrand Reinhold, 1991.
- [18] Eberhart R.K.E. Particle swarm optimization. *Proc. ICNN'95*, 1995, pp. 1942–1948.
DOI: <https://doi.org/10.1109/ICNN.1995.488968>
- [19] Mirjalili S., Mirjalili S.M., Lewis A. Grey Wolf Optimizer. *Adv. Eng. Softw.*, 2014, vol. 69, pp. 46–61. DOI: <https://doi.org/10.1016/j.advengsoft.2013.12.007>

Diveev A.I. — Dr. Sc. (Eng.), Professor, Chief Researcher, Federal Research Center “Computer Science and Control”, Russian Academy of Sciences (Vavilova ul. 44, korp. 2, Moscow, 119333 Russian Federation).

Please cite this article in English as:

Diveev A.I. Evolutionary computation for solving the terminal optimal control problem. *Herald of the Bauman Moscow State Technical University, Series Instrument Engineering*, 2023, no. 1 (142), pp. 44–59 (in Russ.).
DOI: <https://doi.org/10.18698/0236-3933-2023-1-44-59>