

Условия отсутствия свойств унимодальности функционала в задаче оптимального управления с фазовыми ограничениями¹

А. И. Дивеев

*Федеральный исследовательский центр «Информатика и управление» РАН
119333, Москва, ул. Вавилова, 44, корп. 2*

e-mail: aidiveev@mail.ru

Аннотация. Статья посвящена исследованию свойств унимодальности функционала в задаче оптимального управления. В статье доказано, что при определенных условиях фазовые ограничения, включенные в функционал в виде штрафной функции, делают функционал не унимодальным. Отсутствие унимодальности функционала указывает на малую эффективность использования в этих задачах градиентных методов. Предложено использовать для численного решения задач оптимального управления с фазовыми ограничениями эволюционные алгоритмы. Приведено доказательство теоремы о том, что при выполнении определенных условий эволюционные алгоритмы сходятся быстрее, чем случайный поиск. Приведены результаты вычислительного эксперимента, в котором сравнивались алгоритмы решения задачи оптимального управления двумя мобильными роботами с фазовыми ограничениями. Результаты эксперимента показали эффективность применения эволюционных алгоритмов для численного решения задач оптимального управления с фазовыми ограничениями по сравнению с градиентными методами и случайным поиском.

Ключевые слова: задача оптимального управления с фазовыми ограничениями, управление группой роботов, унимодальность функционала, скорость сходимости эволюционного алгоритма.

1. Введение

Наиболее общий прием решения задачи оптимального управления — это редукция задачи оптимального управления к задаче нелинейного программирования и ее решение классическими или современными численными методами [1].

Как показывают исследования, переход от задачи оптимального управления к задаче нелинейного программирования выполнить не сложно, но в результате мы получаем задачу нелинейного программирования высокой размерности, и, что более важно, в большинстве случаев с неунимодальной целевой функцией.

¹ Работа выполнена при финансовой поддержке грантов РФФИ 16-29-04224-офи_м и 17-08-01203-а.

Высокая размерность задачи оптимизации с неунимодальной целевой функцией не позволяет применять к ее решению точные методы глобальной оптимизации. Например, для использования метода неравномерного покрытия [2] необходимо вычислять оценки целевой функции для каждой из областей, на которые разбивается пространство поиска. Количество областей для оценки функционала при этом превышает величину 2^r , где r — размерность пространства поиска. Заметим, что при редукции задачи оптимального управления к задаче нелинейного программирования ситуация, когда $r \geq 100$, является типичной, например, если объект содержит m управлений и мы разбиваем время управления на k интервалов, то получаем $r = mk$, причем, чем больше k , тем точнее решение задачи. Данная проблема была бы легко решена при использовании современных методов решения задачи нелинейного программирования, например, методов стохастического градиентного поиска, которые используются сегодня успешно для обучения нейронных сетей, но в этом случае мы должны быть уверены, что целевая функция является унимодальной. К сожалению, большинство прикладных задач оптимального управления имеют не унимодальный функционал, особенно задачи с фазовыми ограничениями, что часто встречается при управлении роботами или группой роботов, где каждый объект является фазовым ограничением для других объектов. Этим обстоятельством вызвана большая популярность в последнее время эволюционных алгоритмов, которые можно отнести к методам адаптивного случайного поиска. Данные методы малочувствительны к размерности задачи и форме функционала, но почти всегда при использовании этих методов достаточно трудно определить, как далеко расположено найденное ими решение от оптимального решения.

В настоящей работе рассматриваем некоторые условия, при которых функционал в задаче оптимального управления теряет свойство унимодальности. Мы вводим меру, независимую от времени, в пространстве решений дифференциальных уравнений, и доказываем теорему, что при определенных условиях наличие фазовых ограничений приводит к неунимодальности функционала. Далее рассматриваем задачу управления группой объектов и показываем, что с мерой, зависящей от времени, каждый объект является фазовым ограничением для других объектов, а для специальных симметричных систем при определенных начальных и терминальных условиях функционал всегда является неунимодальным.

2. Решения системы дифференциальных уравнений в задаче оптимального управления

Рассмотрим систему из n обыкновенных дифференциальных уравнений в форме Коши со свободным вектором \mathbf{u} управления в правой части.

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

где $\mathbf{x} \in \mathbf{R}^n$, $\mathbf{u} \in \mathbf{R}^m$.

Пусть значения вектора управления \mathbf{u} определены функциями времени

$$\mathbf{u} = \mathbf{v}(t), \quad (2)$$

где $\mathbf{u} = [u_1 \dots u_m]^T$, $\mathbf{v} = [v_1(t) \dots v_m(t)]^T$.

Рассмотрим множество таких функций

$$\mathbf{V} = \{\mathbf{v}^i(t) : i = 1, \dots, M\}, \quad (3)$$

при которых решение $\mathbf{x}(t)$ системы дифференциальных уравнений (1),

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{v}^i(t)), \mathbf{v}^i(t) \in \mathbf{V}, \quad (4)$$

если будет начинаться в момент $t_0 = 0$ в точке

$$\mathbf{x}^0 = [x_1^0 \dots x_n^0]^T, \quad (5)$$

то в момент t_i будет заканчиваться в точке

$$\mathbf{x}^f = [x_1^f \dots x_n^f]^T, \mathbf{x}^f \neq \mathbf{x}^0. \quad (6)$$

Пусть выполняется условие

$$t_i \leq t^+, i = 1, \dots, M. \quad (7)$$

Дополним все решения системы условиями терминальной устойчивости

$$\tilde{\mathbf{x}}^i(t) = \begin{cases} \mathbf{x}^i(t), & \text{если } t < t_i \\ \mathbf{x}^f & \text{иначе} \end{cases}, \quad (8)$$

где $\mathbf{x}^i(t)$ — решение системы (1) при управлении $\mathbf{u} = \mathbf{v}^i(t)$.

Пусть

$$\tilde{\mathbf{X}} = \{\tilde{\mathbf{x}}^i(t) : i = 1, \dots, M\} \quad (9)$$

множество решений системы (1) дополненных условиями терминальной устойчивости (8).

Введем расстояние между двумя решениями из $\tilde{\mathbf{X}}$

$$\Delta(\tilde{\mathbf{x}}^i, \tilde{\mathbf{x}}^j) = \max \left\{ \max_{\alpha \in [0; t^+]} \min_{\beta \in [0; t^+]} \|\tilde{\mathbf{x}}^i(\alpha) - \tilde{\mathbf{x}}^j(\beta)\|, \right. \\ \left. \max_{\alpha \in [0; t^+]} \min_{\beta \in [0; t^+]} \|\tilde{\mathbf{x}}^i(\beta) - \tilde{\mathbf{x}}^j(\alpha)\| \right\}, \quad (10)$$

где $\|\mathbf{x}\|$ — любая выпуклая норма вектора в \mathbf{R}^n , например,

$$\|\mathbf{x}\|_\infty = \max\{|x_i| : i = 1, \dots, n\} \text{ или } \|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^n x_i^2}.$$

Определение 1. Фундаментальной последовательностью $F(\tilde{\mathbf{x}}^i, \tilde{\mathbf{x}}^j)$ решений называется множество решений $\{\tilde{\mathbf{x}}^k : k = 0, \dots, L+1, \tilde{\mathbf{x}}^0 = \tilde{\mathbf{x}}^i, \tilde{\mathbf{x}}^{L+1} = \tilde{\mathbf{x}}^j\} \subseteq \tilde{X}$, для которого выполняются условия:

$$\Delta(\tilde{\mathbf{x}}^k, \tilde{\mathbf{x}}^{k+1}) < \varepsilon, k = 0, \dots, L, \quad (11)$$

где ε — малое положительное число.

Определение 2. Множество \tilde{X} решений непрерывно, если для любых двух решений $\tilde{\mathbf{x}}^i, \tilde{\mathbf{x}}^j \in \tilde{X}$ и любого положительного малого числа ε всегда можно построить фундаментальную последовательность $F(\tilde{\mathbf{x}}^i, \tilde{\mathbf{x}}^j) \subseteq \tilde{X}$.

Определение 3. ε -окрестностью решения в непрерывном множестве решений называется множество всех решений, для которых выполняются условия

$$\Delta(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}^j) \leq \varepsilon.$$

Пусть на непрерывном множестве решений \tilde{X} задан неотрицательный функционал

$$J(\tilde{\mathbf{x}}(t)) : \mathbf{R}^n \rightarrow \mathbf{R}^1. \quad (12)$$

Пусть функционал обладает следующим свойством: для любого решения $\tilde{\mathbf{x}}^i \in \tilde{X}$ и заданной положительной величины δ всегда существует решение $\tilde{\mathbf{x}}^j \in \tilde{X}$ и величина $\varepsilon > 0$, такие, что выполняются условия

$$|J(\tilde{\mathbf{x}}^i(t)) - J(\tilde{\mathbf{x}}^j(t))| < \delta, \Delta(\tilde{\mathbf{x}}^i, \tilde{\mathbf{x}}^j) < \varepsilon. \quad (13)$$

Это означает, что функционал имеет непрерывные значения оценок на множестве решений \tilde{X} .

Теорема 1. Если на непрерывном множестве решений \tilde{X} функционал имеет унимодальный минимум для решения $\tilde{\mathbf{x}}^- \in \tilde{X}$, то для любого решения $\tilde{\mathbf{x}}^i \in \tilde{X}, \tilde{\mathbf{x}}^i \neq \tilde{\mathbf{x}}^-$ всегда можно построить фундаментальную последовательность

$$\{\tilde{\mathbf{x}}^k : k = 0, \dots, L+1, \tilde{\mathbf{x}}^0 = \tilde{\mathbf{x}}^i, \tilde{\mathbf{x}}^{L+1} = \tilde{\mathbf{x}}^i\} \subset \tilde{X}, \quad (14)$$

для которой выполняются условия

$$J(\tilde{\mathbf{x}}^k(t)) \leq J(\tilde{\mathbf{x}}^{k+1}(t)).$$

Доказательство. Рассмотрим решение $\tilde{\mathbf{x}}^i \in \tilde{X}$. В ее ε -окрестности находим решение $\tilde{\mathbf{x}}^h$, для которого функционал дает меньшее значение оценки $J(\tilde{\mathbf{x}}^h(t)) \leq J(\tilde{\mathbf{x}}^i(t))$. Если такого решения нет, то решение $\tilde{\mathbf{x}}^i \in \tilde{X}$ является локальным минимумом, а так как по условию оно не совпадает с минимумом $\tilde{\mathbf{x}}^- \in \tilde{X}$, то функционал не является унимодальным. Следовательно, такое решение существу-

ет. Рассмотрим теперь ε -окрестности решения $\tilde{\mathbf{x}}^i$ и вновь находим в ее окрестности решение $\tilde{\mathbf{x}}^{i_2}$, для которого функционал имеет меньшее значение, $J(\tilde{\mathbf{x}}^{i_2}(t)) \leq J(\tilde{\mathbf{x}}^i(t))$. Повторяем процесс до того момента, пока в окрестности одного из решений не будет найден минимум функционала $\tilde{\mathbf{x}}^- \in \tilde{X}$. Все найденные в ε -окрестностях решения, взятые в обратном порядке, будут составлять фундаментальную последовательность (14). Теорема доказана.

Теорема 2. Пусть в непрерывном множестве решений \tilde{X} существуют такие два решения $\tilde{\mathbf{x}}^i, \tilde{\mathbf{x}}^j \in \tilde{X}$, что в любой фундаментальной последовательности $F(\tilde{\mathbf{x}}^i, \tilde{\mathbf{x}}^j) \subseteq \tilde{X}$ всегда существует решение $\tilde{\mathbf{x}}^k \in F(\tilde{\mathbf{x}}^i, \tilde{\mathbf{x}}^j)$, для которого выполняются условия

$$J(\tilde{\mathbf{x}}^k(t)) > J(\tilde{\mathbf{x}}^i(t)) \text{ и } J(\tilde{\mathbf{x}}^k(t)) > J(\tilde{\mathbf{x}}^j(t)). \quad (15)$$

Тогда функционал $J(\tilde{\mathbf{x}}(t))$ не является униmodalным в задаче минимума

$$J(\tilde{\mathbf{x}}(t)) \rightarrow \min_{\tilde{X}}.$$

Доказательство. Рассмотрим ε -окрестности каждого из решений $\tilde{\mathbf{x}}^i, \tilde{\mathbf{x}}^j \in \tilde{X}$. Найдем в этих окрестностях решения, которые дают меньшие значения оценок функционалов $J(\tilde{\mathbf{x}}^{i_1}(t)) \leq J(\tilde{\mathbf{x}}^i(t))$ и $J(\tilde{\mathbf{x}}^{j_1}(t)) \leq J(\tilde{\mathbf{x}}^j(t))$. Рассмотрим теперь ε -окрестности найденных решений $\tilde{\mathbf{x}}^{i_1}, \tilde{\mathbf{x}}^{j_1} \in \tilde{X}$. И снова в их ε -окрестностях найдем решения с меньшими значениями оценок функционала. Повторяем процесс до тех пор, пока не найдем минимумы $\tilde{\mathbf{x}}^{i_l}$ и $\tilde{\mathbf{x}}^{j_k}$, в ε -окрестностях которых уже нет решений с меньшими значениями оценок функционалов. Если эти минимумы совпадают, $\tilde{\mathbf{x}}^{i_l} = \tilde{\mathbf{x}}^{j_k}$, то это означает, что между решениями $\tilde{\mathbf{x}}^i, \tilde{\mathbf{x}}^j \in \tilde{X}$ можно построить фундаментальную последовательность, для которой не выполняются условия (15) теоремы. Следовательно, эти решения не совпадают $\tilde{\mathbf{x}}^{i_l} \neq \tilde{\mathbf{x}}^{j_k}$. Это означает, что функционал на множестве решений не является униmodalным. Теорема доказана.

Рассмотрим проекции решений в \mathbf{R}^2 . Из всех компонент решений $\tilde{\mathbf{x}} = [\tilde{x}_1 \dots \tilde{x}_n]^T$ нас интересуют только две компоненты. Пусть это будут компоненты \tilde{x}_1, \tilde{x}_2 . Таким образом мы рассматриваем подпространство \mathbf{R}^2 в \mathbf{R}^n с преобразованием

$$\mathbf{y} = \mathbf{C}\mathbf{x}, \quad (16)$$

где $\mathbf{x} \in \mathbf{R}^n, \mathbf{y} \in \mathbf{R}^2$,

$$C = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \end{bmatrix}.$$

Пусть в \mathbf{R}^2 задана замкнутая область $D \subseteq \mathbf{R}^2$. Размер области определим по ее диаметру.

Определение 4. Конструктивным диаметром замкнутой области называется диаметр максимальной сферы, вписанной в эту область.

Пусть S максимального размера окружность, вписанная в D

$$S(x_1, x_2) = (x_1 - x_1^*)^2 + (x_2 - x_2^*)^2 - R^2 = 0,$$

где x_1^*, x_2^* — координаты центра окружности.

Тогда диаметр области D равен $2R$ и $\forall x_1, x_2$, для которых $S(x_1, x_2) \leq 0$, $x_1, x_2 \in D$.

Рассмотрим множество решений в \mathbf{R}^2 из точки $\mathbf{y}^0 = \mathbf{C}\mathbf{x}^0$, в точку $\mathbf{y}^f = \mathbf{C}\mathbf{x}^f$.

Множество решений в \mathbf{R}^2 , которые начинаются из \mathbf{y}^0 в \mathbf{y}^f , дополненные условиями терминальной устойчивости (8), обозначим \tilde{Y} .

Аксиома 1. Если во множестве решений \tilde{Y} существуют два решения $\tilde{\mathbf{y}}^i$ и $\tilde{\mathbf{y}}^j$ с расстоянием $\Delta(\tilde{\mathbf{y}}^i, \tilde{\mathbf{y}}^j)$, то между этими решениями нельзя поместить область с диаметром $2R > \Delta(\tilde{\mathbf{y}}^i, \tilde{\mathbf{y}}^j)$ так, чтобы решения $\tilde{\mathbf{y}}^i$ и $\tilde{\mathbf{y}}^j$ не пересекали данную область.

Пусть в \mathbf{R}^2 задана область D с диаметром $2R$. Возьмем в области D точку (x_1^p, x_2^p) . Построим полярную систему координат $\{\rho, \phi\}$ с центром в точке (x_1^p, x_2^p) . Преобразование координат из декартовых координат в \mathbf{R}^2 в полярные осуществляем по формулам

$$\rho = \sqrt{(x_1 - x_1^p)^2 + (x_2 - x_2^p)^2}, \quad (17)$$

$$\phi = \begin{cases} \arctan\left(\frac{x_2 - x_2^p}{x_1 - x_1^p}\right) \pm 2\pi, & \text{если } x_1 - x_1^p > 0 \\ \operatorname{sgn}(x_2 - x_2^p)\pi - \arctan\left(\frac{x_2 - x_2^p}{x_1 - x_1^p}\right) \pm 2\pi, & \text{если } x_1 - x_1^p < 0 \\ \operatorname{sgn}(x_2 - x_2^p)\frac{\pi}{2} \pm 2\pi, & \text{если } x_1 - x_1^p = 0 \end{cases} \quad (18)$$

Все решения $\tilde{\mathbf{y}}^k(t) = [\tilde{x}_1^k(t) \quad \tilde{x}_2^k(t)]^T$ из \tilde{Y} преобразуем в полярные координаты $\{\rho, \phi\}$ и записываем в виде $\mathbf{p}^k(t) = [\tilde{\rho}^k(t) \quad \tilde{\phi}^k(t)]^T$.

Теорема 3. Пусть во множестве решений \tilde{Y} , которые начинаются в момент 0 в точке $\mathbf{y}^0 = [x_1^0 \ x_2^0]^T$, в момент t_f проходят точку $\mathbf{y}^f = [x_1^f \ x_2^f]^T$, есть два решения $\tilde{\mathbf{y}}^i, \tilde{\mathbf{y}}^j$, не пересекающие область D с конструктивным диаметром $2R$, т. е. $\forall t \in [0; t_f] S(x_1^i(t), x_2^i(t)) > 0$ и $S(x_1^j(t), x_2^j(t)) > 0$, где $S(x_1, x_2) = (x_1 - x_1^*)^2 + (x_2 - x_2^*)^2 - R^2 = 0$, $x_1^*, x_2^* \in D$ и для которых выполняются условия

$$|\tilde{\varphi}^i(t_f) - \tilde{\varphi}^j(t_f)| = 2\pi, \quad (19)$$

где $\tilde{\varphi}^i(t), \tilde{\varphi}^j(t)$ — угловые компоненты в представлении решений $\tilde{\mathbf{y}}^i, \tilde{\mathbf{y}}^j$ в полярных координатах $\tilde{\mathbf{p}}^i(t) = [\tilde{\rho}^i(t), \tilde{\varphi}^i(t)]^T$, $\tilde{\mathbf{p}}^j(t) = [\tilde{\rho}^j(t), \tilde{\varphi}^j(t)]^T$ по формулам (17), (18), то этого достаточно, чтобы расстояние между решениями $\tilde{\mathbf{y}}^i, \tilde{\mathbf{y}}^j$ было не менее $2R$:

$$\Delta(\tilde{\mathbf{y}}^i, \tilde{\mathbf{y}}^j) \geq 2R.$$

Доказательство. Рассмотрим рис. 1.

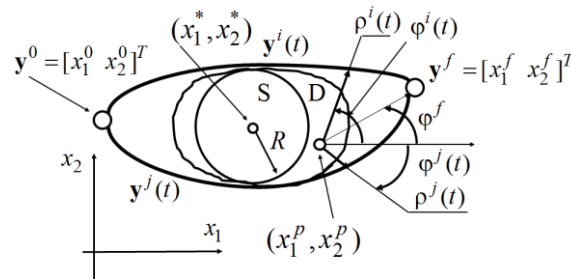


Рисунок 1. Решения в полярных координатах

На рисунке представлен случай, когда углы $\tilde{\varphi}^i(t_f)$ и $\tilde{\varphi}^j(t_f)$ отличаются на величину 2π . Решения в полярных координатах $\tilde{\mathbf{p}}^i(t) = [\tilde{\rho}^i(t), \tilde{\varphi}^i(t)]^T$, $\tilde{\mathbf{p}}^j(t) = [\tilde{\rho}^j(t), \tilde{\varphi}^j(t)]^T$ стартуют в момент $t = 0$ с одного угла φ^0 и завершаются на угле φ^f , при этом решение $\tilde{\mathbf{p}}^i(t)$ перемещается по часовой стрелке, а решение $\tilde{\mathbf{p}}^j(t)$ — против часовой стрелки. Так как $\varphi^0 > \varphi^f$, а решение $\tilde{\mathbf{p}}^j(t)$ движется в сторону увеличения угла, то $\tilde{\varphi}^j(0) = \varphi^0 - 2\pi$. Следовательно,

$$\tilde{\varphi}^j(t_f) = \varphi^0 + (\varphi^f - \varphi^0 + 2\pi),$$

$$\tilde{\varphi}^i(t_f) = \varphi^0 - (\varphi^f - \varphi^0).$$

Откуда получаем $|\tilde{\varphi}^i(t_f) - \tilde{\varphi}^j(t_f)| = 2\pi$. Согласно аксиоме 1, если расстояние между решениями меньше $2R$, то эти решения пересекут область D , т. е. не будут соответствовать условиям теоремы. Теорема доказана.

Введем на множестве решений функционал (12). Пусть в \mathbf{R}^2 определена область $D \subseteq \mathbf{R}^2$ такая, что, если решение $\tilde{\mathbf{y}}^k(t)$ попадает в область D , то значение функционала существенно увеличивается: $\forall \tilde{\mathbf{y}}(t) \in Y$. Справедливо условие $J(\tilde{\mathbf{y}}(t)) < J(\tilde{\mathbf{y}}^k(t))$, если $\forall t \in [0; t_f] \forall \tilde{\mathbf{y}}(t) \notin D$ и $\exists t' \in [0; t_f]$, при котором $\forall \tilde{\mathbf{y}}^k(t) \in D$.

Пусть во множестве решений \tilde{Y} существуют два решения $\tilde{\mathbf{y}}^i(t)$ и $\tilde{\mathbf{y}}^j(t)$, которые не пересекают область D и удовлетворяют условиям (16). Это означает, что любая фундаментальная последовательность, построенная между ними, будет содержать решение $\tilde{\mathbf{y}}^k(t)$, которое пересекает область D , поэтому $J(\tilde{\mathbf{y}}^k(t)) > J(\tilde{\mathbf{y}}^i(t))$ и $J(\tilde{\mathbf{y}}^k(t)) > J(\tilde{\mathbf{y}}^j(t))$. Следовательно, по теореме 2 функционал не будет унимодальным на множестве решений \tilde{Y} .

3. Эволюционные алгоритмы

Эволюционные алгоритмы часто используются для решения задачи глобальной оптимизации. Общими свойствами эволюционных алгоритмов являются применение при поиске множества возможных решений и эволюционные преобразования элементов этого множества на основе их оценок по значениям функционала с целью получения новых возможных решений с более лучшими оценками. Эволюционные преобразования определяют различия между эволюционными алгоритмами. Часто авторы эволюционных алгоритмов обосновывают эволюционные преобразования поведением некоторых животных, названиями которых называют эти эволюционные алгоритмы. Во всех эволюционных алгоритмах при выполнении эволюционных преобразований используют генератор случайных чисел.

Сходимость эволюционных алгоритмов, как правило, рассматривается отдельно для каждого вида алгоритма и доказывается, что при увеличении числа эволюционных преобразований, которые не ухудшают оценки возможных решений, всегда эволюционный алгоритм приведет к ситуации, когда множество возможных решений содержит решение задачи. Сгенерируем новое возможное решение случайно без эволюционных преобразований. Оценим его по значению функционала и, если оно оказалось лучше, чем предшествующее, оставим его вместо предшествующего возможного решения. В результате получаем алгоритм случайного поиска, который также никогда не ухудшает оценку возможного решения и на бесконечном количестве итераций находит оптимальное решение задачи.

Поскольку алгоритм случайного поиска и все эволюционные алгоритмы при построении нового возможного решения используют генератор случайных чисел, то важным вопросом становится не сходимость эволюционного алгоритма, а эффективность эволюционных преобразований, которые должны увеличить вероятность нахождения оптимального решения по сравнению с алгоритмом случайного поиска. Покажем, что при определенных свойствах эволюционного алгоритма вероятность нахождения решения с более лучшей оценкой целевой функции у эволюционного алгоритма выше, чем у алгоритма случайного поиска.

Определение 5. Коэффициентом похожести двух кодов возможных решений S_α и S_β называется величина $\rho(S_\alpha, S_\beta)$, которая обладает следующими свойствами:

$$\rho(S_\alpha, S_\beta) = \rho(S_\beta, S_\alpha), \quad \forall S_\alpha \text{ и } \forall S_\beta, \quad 0 \leq \rho(S_\alpha, S_\beta) \leq 1,$$

если $\rho(S_\alpha, S_\beta) = 1$, то $S_\alpha = S_\beta$, если $\rho(S_\alpha, S_\beta) > \rho(S_\alpha, S_\gamma)$, то код S_β более похож на код S_α , чем S_γ .

Коэффициент похожести (resembling) для двоичных кодов можно вычислять как разность между единицей и отношением количества несовпадающих бит к общему количеству бит в коде

$$\rho(S_\alpha, S_\beta) = 1 - \left(\sum_{j=1}^L p_j \right)^{-1} \sum_{j=1}^L \sum_{i=1}^{p_j} |s_i^{\alpha,j} - s_i^{\beta,j}|, \quad (20)$$

где $S_\alpha = (\mathbf{s}^{\alpha,1}, \dots, \mathbf{s}^{\alpha,L})$, $S_\beta = (\mathbf{s}^{\beta,1}, \dots, \mathbf{s}^{\beta,L})$, $\mathbf{s}^{\alpha,j} = [s_1^{\alpha,j} \dots s_{p_j}^{\alpha,j}]^T$, $\mathbf{s}^{\beta,j} = [s_1^{\beta,j} \dots s_{p_j}^{\beta,j}]^T$, $s_i^{\alpha,j} \in \{0,1\}$, $s_i^{\beta,j} \in \{0,1\}$, $i = 1, \dots, p_j$, $j = 1, \dots, L$.

Для кодов числовых векторов, $S_\alpha = (\mathbf{s}^\alpha) = [s_1^\alpha \dots s_p^\alpha]^T$, $S_\beta = (\mathbf{s}^\beta) = [s_1^\beta \dots s_p^\beta]^T$, $s_i^\alpha, s_i^\beta \in \mathbf{R}^1$, $i = 1, \dots, p$, при условии, что известны ограничения на значения компонент векторов

$$s_i^- \leq s_i^\alpha \leq s_i^+, \quad s_i^- \leq s_i^\beta \leq s_i^+, \quad i = 1, \dots, p, \quad (21)$$

где s_i^- , s_i^+ , $i = 1, \dots, p$ — известные величины, коэффициент похожести можно вычислить по разности между единицей и средней нормой разности нормированных векторов

$$\rho(S_\alpha, S_\beta) = 1 - \frac{1}{p} \sum_{i=1}^p \frac{|s_i^\alpha - s_i^\beta|}{s_i^+ - s_i^-}. \quad (22)$$

В общем случае формулы для вычисления коэффициента похожести могут не совпадать с (21), (22) и зависеть от особенностей кодировки возможного решения и алгоритма поиска.

Определение 6. Множество кодов возможных решений $\mathbf{S} = \{S_1, \dots, S_H\}$ с заданной функцией оценки $F(S)$ обладает свойством преемственности (succession), если для определенной величины коэффициента похожести ρ_1 , существует величина $\delta < \delta_1(\rho_1)$, такая, что с вероятностью $\Pr > p_1(\rho_1)$ выполняется соотношение

$$|F(S_\alpha) - F(S_\beta)| < \delta, \quad (23)$$

где $S_\alpha, S_\beta \in \mathbf{S}$, $\rho(S_\alpha, S_\beta) > \rho_1$.

Согласно определению, похожие коды возможных решений в эволюционных алгоритмах с некоторой не нулевой вероятностью дают близкие значения оценок

$$\Pr[|F(S_\alpha) - F(S_\beta)| < \delta] > p_1(\rho_1). \quad (24)$$

Предположим, что оценки всех возможных кодов определенной длины лежат в ограниченном диапазоне значений

$$\forall S \quad f^- \leq F(S) \leq f^+. \quad (25)$$

Тогда вероятность генерации кода S_β возможного решения с оценкой $f_\beta = F(S_\beta)$ меньше заданного значения f_α пропорциональна величине

$$\Pr[f_\alpha > f_\beta] = (f_\alpha - f^-) / (f^+ - f^-). \quad (26)$$

Теорема 4. Если эволюционный алгоритм порождает новые возможные решения со свойством преемственности, то существует такая величина оценки f^* возможного решения, что после определенного количества K эволюционных преобразований вероятность нахождения решения с более лучшей оценкой $\Pr[f^* > f_\alpha^{(K+1)}]$ в эволюционном алгоритме больше вероятности возникновения этого же события в алгоритме случайного поиска.

Доказательство. Пусть в результате случайного поиска и эволюционных вычислений было найдено возможное решение с оценкой f^* . Вероятность нахождения более лучшего возможного решения \tilde{S} алгоритмом случайного поиска определяется соотношением (30).

$$\Pr[f^* > F(\tilde{S})] = (f^* - f^-) / (f^+ - f^-).$$

Пусть эволюционный алгоритм обладает свойством преемственности. Тогда для кода возможного решения \tilde{S} с величиной коэффициента похожести $\rho_1(S^*, \tilde{S})$ с кодом S^* наилучшего найденного к данному моменту возможного решения вероятность получения оценки, лучше f^* определяется соотношением (23)

$$\Pr[F(S^*) - F(\tilde{S}) < \delta] > 0.5 p_1(\rho_1).$$

Коэффициент 0.5 введен в формулу для учета только тех возможных решений S из интервала $|F(S^*) - F(S)| < \delta$, которые дают меньшее значение оценки $F(S) < F(S^*)$.

Для доказательства теоремы необходимо выполнение неравенства

$$\Pr[f^* > F(\tilde{S})] < \Pr[F(S^*) - F(\tilde{S}) < \delta]$$

или

$$\begin{aligned} (f^* - f^-)/(f^+ - f^-) &< 0.5 p_1(\rho_1), \\ f^* &< 0.5(f^+ - f^-) p_1(\rho_1) + f^-, \\ f^* &< 0.5 f^+ p_1(\rho_1) + (1 - 0.5 p_1(\rho_1)) f^-. \end{aligned}$$

Неравенство не выполнимо только при условии $p_1(\rho_1) = 0$. Следовательно, если в эволюционном алгоритме новые возможные решения создаются со свойством преемственности, то вероятность получения возможного решения с более лучшей оценкой у эволюционного алгоритма выше, чем у алгоритма случайного поиска. Теорема доказана.

4. Вычислительный эксперимент

Рассмотрим задачу оптимального управления группой объектов с фазовыми ограничениями.

Заданы математические модели мобильных роботов [3, 4]

$$\begin{aligned} \dot{x}_1^j &= 0.5(u_1^j + u_2^j) \cos(x_3^j), \\ \dot{x}_2^j &= 0.5(u_1^j + u_2^j) \sin(x_3^j), \\ \dot{x}_3^j &= 0.5(u_1^j - u_2^j), \end{aligned} \quad (27)$$

где j — номер робота в группе, $j = 1, \dots, N$; x_1, x_2 — координаты центра масс робота; x_3 — угол поворота оси робота; u_1, u_2 — управляющие сигналы на вращение колес робота.

Управление ограничено

$$u_i^- \leq u_i^j \leq u_i^+, \quad i = 1, 2, \quad j = 1, \dots, N. \quad (28)$$

Заданы начальные состояния роботов:

$$x_i^j(0) = x_i^{0,j}, \quad i = 1, 2, 3, \quad j = 1, \dots, N. \quad (29)$$

Заданы статические фазовые ограничения:

$$\beta(\mathbf{x}^j) = r^2 - (x_1^* - x_1^j)^2 - (x_2^* - x_2^j)^2 \leq 0, \quad j = 1, \dots, N, \quad (30)$$

где r — радиус статических ограничений

Заданы динамические фазовые ограничения как условия запрета сближения роботов между собой

$$\delta(\mathbf{x}^i, \mathbf{x}^{i+k}) = r_0^2 - (x_1^i - x_1^{i+k})^2 - (x_2^i - x_2^{i+k})^2 \leq 0, \quad (31)$$

где r_0 — предельное расстояние сближения роботов; $k = 1, \dots, N-i$; $i = 1, \dots, N-1$.

Заданы терминальные состояния роботов:

$$x_i^j - x_i^{f,j} = 0, \quad i = 1, 2, 3, \quad j = 1, \dots, N. \quad (32)$$

Задан критерий качества:

$$\tilde{J} = t_f \rightarrow \min, \quad (33)$$

где

$$t_f = \begin{cases} t, & \text{если } t < t^+ \text{ и } \max \left\{ \sqrt{\sum_{i=1}^3 (x_i^j(t) - x_i^J)^2} : j = 1, \dots, N \right\} < \varepsilon, \\ t^+ & \text{— иначе} \end{cases}$$

t^+ — максимальное предельное время процесса управления; ε — малая положительная величина.

Включим фазовые ограничения в критерий качества:

$$J = t_f + \sum_{j=1}^N \int_0^{t_f} \mathcal{G}(\beta(\mathbf{x}^j)) dt + \sum_{i=1}^{N-1} \sum_{k=i+1}^N \int_0^{t_f} \mathcal{G}(\delta(\mathbf{x}^i, \mathbf{x}^{i+k})) dt \rightarrow \min,$$

где $\beta(\mathbf{x}(t))$ и $\delta_k(\mathbf{x}(t))$ определяются по формулам (30) и (31) соответственно,

$$\mathcal{G}(A) = \begin{cases} 1, & \text{если } A \geq 1 \\ 0 & \text{— иначе} \end{cases}.$$

Мы используем следующие значения постоянных параметров модели: $n = 3$, $m = 2$, $N = 4$, $x_1^0 = 0$, $x_2^0 = 0$, $x_3^0 = 0$, $x_4^0 = 0$, $x_5^0 = 10$, $x_6^0 = 0$, $x_7^0 = 10$, $x_8^0 = 0$, $x_9^0 = 0$, $x_{10}^0 = 10$, $x_{11}^0 = 10$, $x_{12}^0 = 0$, $x_1^f = 10$, $x_2^f = 10$, $x_3^0 = 0$, $x_4^{10} = 10$, $x_5^0 = 0$, $x_6^0 = 0$, $x_7^0 = 0$, $x_8^0 = 10$, $x_9^0 = 0$, $x_{10}^0 = 0$, $x_{11}^0 = 0$, $x_{12}^0 = 0$, $t^+ = 2.8$, $\varepsilon = 0.01$, $u_1^- = -10$, $u_2^- = -10$, $u_1^+ = 10$, $u_2^+ = 10$.

5. Трансформация к задаче нелинейного программирования

Для решения задачи оптимального управления (27)–(33) бесконечномерной размерности трансформируем ее в задачу нелинейного программирования конечномерной размерности. Выберем интервал Δt и разобьем время управления на L интервалов

$$L = \left\lfloor \frac{t^+}{\Delta t} \right\rfloor + 1.$$

В каждом интервале мы аппроксимируем управление функцией, зависящей от конечного числа параметров.

Заменяем управление полученной аппроксимирующей функцией и обрезаем эти функции при нарушении ограничений на управление:

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

где $\tilde{u}_i(t)$ — аппроксимирующая функция.

Для аппроксимации мы используем полиномы Эрмита четвертого порядка

$$\begin{aligned} \tilde{u}_i(t) = & q_{4(i-1+Nml)+1} 2(t-l\Delta t) + q_{4(i-1+Nml)+2} (4(t-l\Delta t)^2 - 2) + \\ & + q_{4(i-1+Nml)+3} (8(t-l\Delta t)^3 - 12t) + q_{4(i-1+Nml)+4} (16(t-l\Delta t)^4 - 48(t-l\Delta t)^2 - 12t), \end{aligned}$$

где $i=1, \dots, Nm$, $t \in [l\Delta t, (l+1)\Delta t)$, $l=0, \dots, L-1$.

Находим значения $4NmL$ параметров:

$$\mathbf{q} = [q_1 \quad \dots \quad q_{NmL}]^T.$$

На значения параметров задаем ограничения

$$-2 \leq q_i \leq 2, \quad i=1, \dots, 4NmL.$$

В эксперименте значения $\Delta t = 1.4$, $L = 2$.

Размерность искомого вектора параметров составила

$$4NmL = 4 \cdot 4 \cdot 2 \cdot 2 = 64.$$

Следующий тип аппроксимации с помощью кривых Безье [5]:

$$\begin{aligned} \tilde{u}_i(t) = & q_{4(i-1+Nml)+1} (1 - (t-l\Delta t)^3) + 3q_{4(i-1+Nml)+2} (1 - (t-l\Delta t)^2)(t-l\Delta t) + \\ & + 3q_{4(i-1+Nml)+3} (1 - (t-l\Delta t))(t-l\Delta t)^2 + q_{4(i-1+Nml)+4} (t-l\Delta t)^3, \end{aligned}$$

где $i=1, \dots, Nm$; $t \in [l\Delta t, (l+1)\Delta t)$; $l=0, \dots, L-1$.

В эксперименте мы использовали следующие ограничения:

$$q_i^- = -8, \quad q_i^+ = 8, \quad i=1, \dots, 4NmL \text{ и } \Delta t = 1.4, \quad L = 2.$$

Размерность вектора параметров составила

$$4NmL = 4 \cdot 4 \cdot 2 \cdot 2 = 64.$$

Наиболее часто при аппроксимации функций управления используют кубические полиномы, которые позволяют получить непрерывные по значению, а также первой и второй производной функции. В данном случае ограничиваемся непрерывностью самой функции и по первой производной. На интервале $0, t \in [0; \Delta t)$ мы

ищем все четыре коэффициента, а на последующих интервалах по два коэффициента, так как два коэффициента определяются из равенства значений полиномов и их производных с обеих сторон границы интервала

$$\begin{aligned}\tilde{u}_i(t) &= q_{4(i-1)+1}t^3 + q_{4(i-1)+2}t^2 + q_{4(i-1)+3}t + q_{4(i-1)+4}, \quad t \in [0; \Delta t), \\ \tilde{u}_i(t) &= q_{4(i-1+Nm)+5}(t-\Delta t)^3 + q_{4(i-1+Nm)+6}(t-\Delta t)^2 + a_{3,l}(t-\Delta t) + a_{4,l}, \quad t \in [\Delta t; 2\Delta t), \\ a_{3,l} &= 3q_{4(i-1+Nm(l-1))+1}(\Delta t)^2 + 2q_{4(i-1+Nm(l-1))+2}\Delta t + q_{4(i-1+Nm(l-1))+3}, \\ a_{4,l} &= q_{4(i-1+Nm(l-1))+1}(\Delta t)^3 + q_{4(i-1+Nm(l-1))+2}(\Delta t)^2 + q_{4(i-1+Nm(l-1))+3}\Delta t + q_{4(i-1+Nm(l-1))+4}, \\ \tilde{u}_i(t) &= q_{4(i-1+Nm)+5}(t-l\Delta t)^3 + q_{4(i-1+Nm)+6}(t-l\Delta t)^2 + a_{3,l}(t-l\Delta t) + a_{4,l}, \\ & \quad t \in [l\Delta t; (l+1)\Delta t), \quad l = 2, \dots, L-1, \\ a_{3,l} &= 3q_{4(i-1+Nm(l-1))+5}(l\Delta t)^2 + 2q_{4(i-1+Nm(l-1))+6}l\Delta t + a_{3,l-1}, \\ a_{4,l} &= q_{4(i-1+Nm(l-1))+5}(l\Delta t)^3 + q_{4(i-1+Nm(l-1))+6}(l\Delta t)^2 + a_{3,l-1}l\Delta t + a_{4,l-1},\end{aligned}$$

где $i = 1, \dots, Nm$, $l = 0, \dots, L-1$.

В эксперименте были заданы следующие значения ограничений:

$$q_i^- = -2.8, \quad q_i^+ = 2.8, \quad i = 1, \dots, (4 + 2(L-1))Nm$$

и значения $\Delta t = 0.95$, $L = 3$.

Размерность вектора искомым параметров составила

$$(4 + 2(L-1))Nm = (4 + 2 \cdot 2) \cdot 4 \cdot 2 = 64.$$

При кусочно-линейной аппроксимации ищем управления как линейные функции:

$$\tilde{u}_i(t) = \frac{q_{Nm+i+1} - q_{Nm+i}}{\Delta t}(t-l\Delta t) + q_{Nm+i},$$

где $i = 1, \dots, Nm$; $t \in [l\Delta t; (l+1)\Delta t)$; $l = 0, \dots, L-1$.

В эксперименте использовали следующие значения ограничений:

$$q_i^- = -20, \quad q_i^+ = 20, \quad i = 1, \dots, NLm$$

и значения $\Delta t = 0.255$, $L = 11$.

Размерность вектора параметров составила $Nm = 11 \cdot 4 \cdot 2 = 88$.

Наиболее простая аппроксимация с помощью кусочно-постоянной функции

$$\tilde{u}_i(t) = q_{Nm+i},$$

где $i = 1, \dots, Nm$; $t \in [l\Delta t; (l+1)\Delta t)$; $l = 0, \dots, L-1$.

В эксперименте использовали следующие ограничения

$$q_i^- = -20, \quad q_i^+ = 20, \quad i = 1, \dots, NLm$$

и значения $\Delta t = 0.255$, $L = 11$.

Размерность вектора управления составила $Nm = 11 \cdot 4 \cdot 2 = 88$.

6. Алгоритмы

Для решения задачи используем пять различных алгоритмов: известный алгоритм наискорейшего градиентного спуска (FGDA), алгоритм роя частиц (PSO) [6–8], генетический алгоритм (GA) [6, 7, 9], простой случайный поиск (RS) и стохастический градиентный алгоритм (ADAM) [6, 10]. Качество поиска эволюционных алгоритмов зависит от количества возможных решений в начальном множестве и числа эволюционных преобразований, поэтому оценивали эффективность алгоритмов по числу вычислений целевого функционала. Параметры алгоритмов были выбраны так, чтобы количество вычислений целевого функционала для каждого алгоритма было приблизительно одинаковым. Заметим, что для вычисления одного значения градиента вектора в алгоритме градиентного спуска количество вычислений значений равно размерности искомого вектора параметров. Результаты вычислительного эксперимента приведены в табл.

Таблица. Результаты эксперимента

PSO				
Cube	7.6864	8.917237	0.892737	2500511
Hermit	5.1408	9.686063	2.117146	2500511
Bezier	4.1660	8.023544	2.758620	2500511
Linear	3.4987	5.497326	1.427645	2501011
Constant	3.8021	5.437543	1.165168	2501011
RS				
Cube	12.0909	13.080324	0.868641	2500021
Hermit	14.0148	15.897309	1.245307	2500021
Bezier	9.2596	14.768950	2.760597	2500021
Linear	5.6273	12.238500	4.751106	2500021
Constant	9.4286	12.790501	2.640101	2500021
GA				
Cube	6.0272	9.299249	1.57050	2494205
Hermit	5.8111	7.282453	1.158785	2488467
Bezier	7.1604	10.165712	1.366374	2491811
Linear	4.8372	7.036774	1.704026	2496725
Constant	9.5319	11.932227	1.859405	2497143
FGDA				
Cube	18.5099	21.604358	3.912750	2594575
Hermit	18.4947	21.301057	1.686402	2591268
Bezier	22.3052	23.314141	1.006914	2606501
Linear	21.4992	22.366198	0.942382	2743742
Constant	23.5008	24.782348	0.779497	2715137
ADAM				
Cube	18.0045	18.743846	0.905551	2550251
Hermit	13.8628	16.149400	2.229859	2550261
Bezier	6.0134	8.905433	1.194444	2550261
Linear	7.6490	9.294152	1.144378	2522661
Constant	7.4528	8.841364	1.026492	2522661

В табл. приведены методы трансформации к задаче нелинейного программирования: Cube с помощью кубического полинома; Hermit — с помощью полиномов Эрмита; Bezier — на основе кривых Безье; Linear — с помощью кусочно-линейной аппроксимации; Constant — с помощью кусочно-постоянной аппроксимации. В колонке 4 указана величина среднеквадратичного отклонения. В пятом столбце показано количество вычислений целевого функционала.

Из таблицы видим, что эволюционные алгоритмы дают лучшие результаты, чем другие алгоритмы. Градиентный спуск работает хуже, чем случайный поиск. Лучшее решение было получено алгоритмом роя частиц. Значение функционала для лучшего решения составило 3.4987.

Лучшее решение имеет следующее значение вектора параметров

$$\tilde{\mathbf{q}} = [-19.386 \ -15.424 \ -19.758 \ -11.291 \ -19.995 \ -7.217 \ 15.213 \ 19.166 \ 14.471 \\ -17.018 \ -0.104 \ 19.665 \ 5.433 \ -19.988 \ -11.661 \ -16.911 \ 19.884 \ 19.914 \ 3.138 \\ -11.472 \ -19.991 \ -11.169 \ 13.684 \ -7.503 \ -14.478 \ -11.982 \ -17.202 \ 16.257 \ 19.916 \\ 19.959 \ 19.839 \ -19.877 \ 4.265 \ 18.173 \ -10.405 \ -19.970 \ -19.448 \ 8.587 \ 12.291 \ 18.394 \\ 14.360 \ 9.441 \ 19.842 \ -19.933 \ -18.911 \ -8.419 \ 4.644 \ 8.434 \ 11.161 \ 12.217 \ 15.492 \\ -7.754 \ -18.977 \ -10.800 \ -10.910 \ 18.196 \ -18.076 \ -19.996 \ 13.777 \ -1.032 \ 16.507 \\ 11.284 \ 9.916 \ -19.273 \ -14.311 \ -19.925 \ -9.799 \ -19.950 \ -0.301 \ -17.725 \ 19.832 \\ 12.084 \ -20.000 \ -10.225 \ 18.007 \ 11.890 \ 19.008 \ -17.670 \ -19.971 \ -7.332 \ -19.675 \\ -19.948 \ 18.937 \ 10.148 \ 16.420 \ 14.907 \ 19.959 \ 3.142]^T.$$

На рис. 2 показаны траектории движения четырех мобильных роботов на плоскости для наилучшего найденного управления

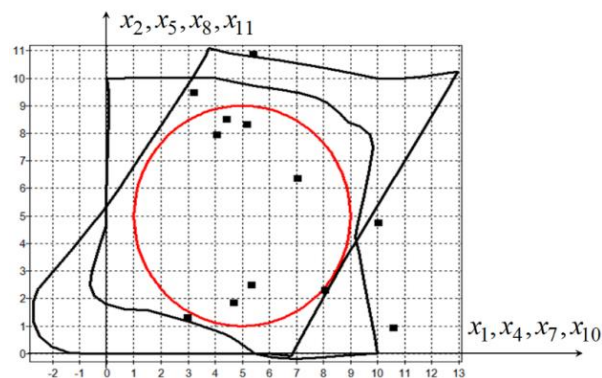


Рисунок 2. Траектории роботов на плоскости для наилучшего решения

7. Выводы

В статье рассмотрены условия, при которых функционал в задаче оптимального управления теряет свойства уни-modalности. Эти условия связаны с наличием фазовых ограничений и управлением несколькими объектами.

Приведен вычислительный эксперимент, результаты которого показали, что при наличии фазовых ограничений и при управлении несколькими объектами более эффективно использовать эволюционные алгоритмы, чем классические градиентные методы.

Литература

- [1] *Евтушенко Ю. Г.* Оптимизация и быстрое автоматическое дифференцирование. — М. : ВЦ РАН, 2013.
- [2] *Евтушенко Ю. Г., Посыпкин М. А.* Метод неравномерных покрытий для решения задач многокритериальной оптимизации с заданной точностью // *Автоматика и телемеханика*. 2014, № 6. С. 49–68.
- [3] *Leena N., Saju K. K.* Modelling and trajectory tracking of wheeled mobile robots // *Procedia Technology*. 2016. Vol. 24. P. 538–545.
- [4] *Šuster P., Jadlovska A.* Tracking trajectory of the mobile robot Khepera II using approaches of artificial intelligence // *Acta Electrotechnica et Informatica*. 2011. Vol. 11. No. 1. P. 38–43.
- [5] *Дивеев А. И., Шмалько Е. Ю.* Метод аппроксимации кривыми Безье для решения задачи оптимального управления посадкой космического аппарата // *Труды института Системного анализа РАН. Динамика неоднородных систем*. 2007. Т. 31, № 1. С. 8–13.
- [6] *Дивеев А. И., Константинов С. В.* Исследование эволюционных алгоритмов для решения задачи оптимального управления // *Труды МФТИ*. 2017. Т. 9, № 3 (35). С. 76–85.
- [7] *Карпенко А. П.* Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой. — М. : Изд-во МГТУ им. Н. Э. Баумана, 2014.
- [8] *Kennedy J., Eberhart R.* Particle Swarm Optimization // *IEEE International Conference on Neural Networks IV*. — Perth, 1995. P. 1942–1948.
- [9] *Kingma D. P., Ba J.* Adam: A Method for Stochastic Optimization // *3rd International Conference for Learning Representations*. arXiv:1412.6980v8 [cs.LG], 2015 (<https://arxiv.org/pdf/1412.6980v8.pdf>).
- [10] *Goldberg D. E.* Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, 1989.

Автор:

Асхат Ибрагимович Дивеев — доктор технических наук, профессор, главный научный сотрудник, Федеральный исследовательский центр «Информатика и управление» РАН

Conditions of not unimodality of functional in problem of optimal control with phase restrictions

A. Diveev

Federal Research Center "Computer Science and Control" of Russian Academy of Sciences
Vavilova str., 44, build. 2, Moscow, Russian Federation, 119333

e-mail: aidiveev@mail.ru

Abstract. The paper is devoted a research of properties of an unimodality of a functional in a problem of optimal control. In the paper it is proved that under certain conditions the phase restrictions included in a functional in the form of a penalty function make a functional not unimodal. The lack of an unimodality of a functional indicates small effectiveness of use in these problems of gradient methods. It is offered to use evolutionary algorithms for numerical problem solving of optimum control with phase restrictions. The proof of the theorem is provided, that when performing determinate conditions evolutionary algorithms have convergence higher than a random search. Results of a computing experiment in which algorithms of the solution of a problem of optimal control of two mobile robots were compared to phase restrictions are given. Results of an experiment showed effectiveness of application of evolutionary algorithms for numerical problem solving of optimum control with phase restrictions in comparison with gradient methods and a random search.

Key words: problem of optimal control with phase restrictions, control of group of robots, functional unimodality, convergence of evolutionary algorithm.

References

- [1] Evtushenko Yu. G. (2013) Optimizatsiya i bystroye avtomaticheskoye differentsirovaniye. Moscow, Computing Center of RAS. [In Rus]
- [2] Yevtushenko Yu. G., Posypkin M. A. (2014) *Autom. and Remote Control*, **75**(6):1025–1040.
- [3] Leena N., Saju K. K. (2016) *Procedia Technology*. 24:538–545.
- [4] Šuster P., Jadlovská A. (2011) *Acta Electrotechnica et Informatica*, **11**(1):38–43.
- [5] Diveev A. I., Shmalko E. Yu. (2007) *Proceedings of the Institute of the system analysis of the Russian Academy of Sciences. Dynamics of inhomogeneous systems*, **31**(1):8–13.
- [6] Diveev A. I., Konstantinov S. V. (2017) *Trudy MFTI*. **9**(3):76–85 [In Rus]
- [7] Karpenko A. (2014) Modern algorithms of search optimization. Algorithms inspired by nature. Moscow, Izd-vo MSTU them. N. E. Bauman. [In Rus]
- [8] Kennedy J., Eberhart R. (1995) Particle Swarm Optimization. Proceedings of the IEEE International Conference on Neural Networks IV, Perth. P. 1942–1948.
- [9] Kingma D. P., Ba J. (2015) Adam: A Method for Stochastic Optimization. 3rd International Conferencing for Learning Representations. arXiv: 1412.6980v8 [cs.LG].
- [10] Goldberg D. E. (1989) Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley.