УДК 519.626.2; 517.977.5

К РЕШЕНИЮ НЕВЫПУКЛЫХ ЗАДАЧ ОПТИМАЛЬНОГО УПРАВЛЕНИЯ С ТЕРМИНАЛЬНЫМ ЦЕЛЕВЫМ ФУНКЦИОНАЛОМ

А. С. Стрекаловский[1], **М. В. Янулевич**[1]

Рассматривается невыпуклая задача оптимального управления линейной по состоянию системой обыкновенных дифференциальных уравнений с терминальным целевым функционалом, задаваемым разностью двух выпуклых функций. Проводится тестирование новых алгоритмов локального и глобального поиска. С этой целью разработан специальный метод генерации тестовых линейно-квадратичных задач оптимального управления. Результаты тестирования демонстрируют эффективность предложенных алгоритмов.

Ключевые слова: невыпуклые задачи оптимального управления, локально и глобально оптимальные процессы, алгоритмы локального и глобального поиска.

1. Введение. Как известно, в невыпуклых задачах оптимального управления (ОУ) может существовать большое количество процессов, удовлетворяющих принципу максимума Понтрягина (ПМП), далеких от глобально оптимального даже по значению целевого функционала [1, 3, 4, 6, 11]. Как следствие, прямое применение стандартных численных методов решения задач ОУ [4, 5, 7, 12–15] оказывается в большинстве случаев неэффективным и сильно зависит от начального приближения. Поэтому разработка новых методов глобального поиска для невыпуклых задач ОУ все более актуализируется. Весьма продуктивными оказываются методы решения специальных классов невыпуклых задач, учитывающие их структуру и свойства [23] и основанные на новых принципах [14, 15, 17].

Пусть $t \in T \triangleq [t_0, t_1]$ — независимая переменная (время), а вектор-функции $t \mapsto x(t)\colon\ \mathbb{R} \to \mathbb{R}^n$ и $t \mapsto u(t)\colon \mathbb{R} \to \mathbb{R}^r$ — фазовое состояние и управление соответственно. Рассмотрим процесс управления $\big(x(\cdot), u(\cdot)\big)$ линейной по состоянию системой обыкновенных дифференциальных уравнений (ОДУ):

$$\dot{x}(t) = A(t)x(t) + b(u(t), t) \quad \overset{\circ}{\forall} t \in T, \quad x(t_0) = x_0 \in \mathbb{R}^n, \tag{1}$$

$$u(\cdot) \in \mathcal{U} = \Big\{ u \in L^r_\infty(T) \mid u(t) \in U \quad \overset{\circ}{\forall} t \in T \Big\}, \tag{2}$$

где множество ограничений на управление $U \subset \mathbb{R}^r$ является компактным [4, 6]. Здесь и далее символ $\overset{\circ}{\forall}$ означает “для почти всех”. Будем предполагать, что компоненты матрицы $A(t) \in \mathcal{M}_{n,n}(\mathbb{R})$ измеримы и ограничены почти всюду на множестве T, а вектор-функция $(u,t) \mapsto b(u,t)\colon \mathbb{R}^r \times \mathbb{R} \to \mathbb{R}^n$ удовлетворяет условиям Каратеодори [4, 6], где $\mathcal{M}_{m,n}(\mathbb{R})$ — множество $m \times n$ матриц с действительными элементами.

Как известно из [3, 4, 6], при выполнении вышеописанных предположений, каждое управление $u(\cdot) \in \mathcal{U}$ порождает единственное абсолютно непрерывное по t состояние $x(t) \triangleq x(t,u)$, $t \in T$, в системе (1), (2). Кроме того, следует отметить, что для любого $t \in T$ множество достижимости системы управления (1) и (2) $R(t) \triangleq \big\{ x(t,u) \in \mathbb{R}^n \mid u \in \mathcal{U} \big\}$ является выпуклым и компактным в пространстве $\mathbb{R}^n$ [3, 4, 6, 7].

Рассмотрим задачу $(\mathcal{P})$ минимизации терминального функционала

$$(\mathcal{P})\colon \quad J(u) \triangleq F\big(x(t_1)\big) \triangleq g\big(x(t_1)\big) - h\big(x(t_1)\big) \downarrow \min_u, \quad u \in \mathcal{U}, \tag{3}$$

над системой управления (1), (2), где $g\colon \mathbb{R}^n \to \mathbb{R}$ и $h\colon \mathbb{R}^n \to \mathbb{R}$ — выпуклые и непрерывно дифференцируемые функции на $\mathbb{R}^n$ [23, 24, 28]. Отметим, что функция $F(\cdot)$ является d.c. функцией, т.е. представима в виде разности двух выпуклых функций, а пара функций (g,h) является ее d.c. разложением [16, 23]. Функционал $J(u) \triangleq F\big(x(t_1)\big)$ будем называть d.c. терминальным функционалом.

Следует отметить специфику данной задачи. Если функция $F(\cdot)$, задающая целевой функционал задачи $(\mathcal{P})$, является выпуклой на выпуклом множестве достижимости $R(t_1)$ системы управления (1), (2) (т.е. $h \equiv 0$), то задача $(\mathcal{P})$ является выпуклой, любой локально оптимальный процесс управления будет

[1] Институт динамики систем и теории управления СО РАН, ул. Лермонтова, 134, 664033, г. Иркутск; А. С. Стрекаловский, зав. лаб., e-mail: strekal@icc.ru; М. В. Янулевич, аспирант, e-mail: max@irk.ru

глобально оптимальным [1, 3], и ПМП является необходимым и достаточным условием оптимальности в задаче $(\mathcal{P})$ [1, 3–7, 11–14].

Иначе дело обстоит, когда функция, задающая целевой функционал $F(\cdot)$, невыпукла (т.е. $h \not\equiv 0$). В этом случае в задаче $(\mathcal{P})$ могут существовать локально оптимальные и критические (удовлетворяющие ПМП) процессы, которые не являются глобально оптимальными, что означает наличие невыпуклой структуры. Тогда, как известно (например, см. [4, 14]), методы, основанные, в частности, на принципе максимума Понтрягина (ПМП), не позволяют найти глобально оптимальное управление в невыпуклой задаче $(\mathcal{P})$. Также не указывает пути улучшения стационарных управлений и принцип Беллмана [3, 4, 11], поскольку на любом отрезке $\widehat{T} = \big[\widehat{t}, t_1\big]$, $t_0 < \widehat{t} < t_1$, задача ОУ обладает прежней невыпуклой структурой.

Частный случай невыпуклой задачи $(\mathcal{P})$, а именно задача максимизации выпуклого (минимизации вогнутого) терминального функционала ($g \equiv 0$): $h\big(x(t_1, u)\big) \uparrow \max\limits_u$, $u \in \mathcal{U}$, над системой управления (1) и (2) исследовалась в [25–27].

Поэтому далее в работе будем предполагать, что функции $g(\cdot)$ и $h(\cdot)$ в представлении (3) целевого функционала задачи $(\mathcal{P})$ нетривиальны, т.е. $g \not\equiv 0$ и $h \not\equiv 0$.

Предложенные в [24] необходимые и достаточные условия глобальной оптимальности для задачи $(\mathcal{P})$ и теория глобального поиска, разработанная в [23], позволили сконструировать алгоритмы поиска локально и глобально оптимальных процессов управления, теоретическое обоснование которых было проведено в [28]. В настоящей работе производится численное тестирование этих алгоритмов для задачи $(\mathcal{P})$.

Статья организована следующим образом. Прежде всего, в разделе 2 предлагается метод генерации тестовых задач вида $(\mathcal{P})$ с известными локальными и глобальными решениями. Насколько нам известно, подобным образом тестовые задачи ОУ генерируются впервые. Раздел 4 посвящен решению выпуклых (частично линеаризованных) задач, полученных из $(\mathcal{P})$ с помощью линеаризации по базовой невыпуклости. Далее, в разделе 5 приводятся результаты численного тестирования алгоритма локального поиска, являющегося одной из важнейших составляющих глобального поиска. После этого в разделе 6 рассматривается алгоритм поиска глобально оптимальных процессов управления, обсуждаются особенности его реализации и приводятся и анализируются результаты вычислительного эксперимента по решению сгенерированных в разделе 3 невыпуклых задач ОУ.

2. Генерация тестовых задач. Как известно из [4–7, 11–15], проблема построения достаточно широкого поля тестовых задач является весьма актуальной, особенно важной в невыпуклой оптимизации. Говоря о задачах ОУ и алгоритмах их решения, одним из требований, предъявляемым к тестовым задачам, является то, что в них желательно знать глобально оптимальные процессы или хотя бы нижние оценки значений задачи, а также стартовые управления, с которых известные методы не попадают на глобально оптимальные процессы.

Повторим, что из-за наличия в невыпуклых задачах ОУ большого числа локально оптимальных или критических (удовлетворяющих ПМП) процессов управления известные методы ОУ останавливаются чаще всего на локально оптимальных процессах. Следует отметить, что, как правило, число таких критических процессов растет экспоненциально с ростом размерности задачи.

Поскольку для задач ОУ $(\mathcal{P})$ не удалось отыскать библиотеку тестовых задач заданной структуры (см. (3)), здесь предлагается метод генерации тестовых задач ОУ с известными локально и глобально оптимальными, а также критическими (удовлетворяющими ПМП) процессами. Предлагаемый метод основан на идее генерации тестовых задач из [21, 22], суть которого заключается в следующем. Сначала строятся задачи небольшой размерности с известными решениями и свойствами — так называемые задачи-ядра. После чего из них конструируется задача произвольной размерности с известными решениями и свойствами. Опишем методику построения тестовых задач ОУ более подробно.

Пусть заданы m задач-ядер $(\mathcal{P}_k)$, $k = 1, 2, \ldots, m$, заключающихся в минимизации функционала

$$J_k(u_k) = \big\langle d_k, x_k(t_1)\big\rangle + \frac{1}{2}\,\big\langle x_k(t_1), D_k x_k(t_1)\big\rangle \downarrow \min_{u_k}$$

над системой управления процессами $\big(x_k(\cdot), u_k(\cdot)\big)$, $x_k(t) \in \mathbb{R}^{n_k}$, $u_k(t) \in \mathbb{R}^{r_k}$, $t \in T$,

$$\dot{x}_k(t) = A_k(t)x_k(t) + B_k(t)u_k(t) + c_k(t), \quad x_k(t_0) = x_k^0,$$

$$u_k(\cdot) \in \mathcal{U}_k = \Big\{ u_k(\cdot) \in L_\infty^{r_k}(T) \mid u_k(t) \in U_k \quad \overset{\circ}{\forall} t \in T \Big\},$$

где вектор $d_k \in \mathbb{R}^{n_k}$ и матрица $D_k \in \mathcal{M}_{n_k, n_k}(\mathbb{R})$. Кроме того, компоненты матричных функций $t \mapsto A_k(t)$, $t \mapsto B_k(t)$ размерности $n_k \times n_k$ и $n_k \times r_k$, соответственно, а также вектор-функция $c_k \colon \mathbb{R} \to \mathbb{R}^{n_k}$ являются непрерывными. Пусть также $U_k \subset \mathbb{R}^{r_k}$ является выпуклым компактным множеством. Отметим сразу же, что ниже при построении тестов $n_k = 2$, $r_k \leqslant 2$.

Рассмотрим задачу ОУ $(\mathcal{P}_0)$:

$$
\begin{aligned}
&J(u) = \big\langle d, x(t_1)\big\rangle + \frac{1}{2}\big\langle x(t_1), Dx(t_1)\big\rangle \downarrow \min_u, \quad u \in \mathcal{U},\\
&\dot{x}(t) = A(t)x(t) + B(t)u(t) + c(t), \quad x(t_0) = x_0,\\
&u(\cdot) \in \mathcal{U} = \Big\{ u \in L^r_\infty(T) \mid u(t) \in U \quad \overset{\circ}{\forall} t \in T \Big\},
\end{aligned}
$$

в которой допустимый управляемый процесс $\big(x(\cdot), u(\cdot)\big)$ в задаче $(\mathcal{P}_0)$ таков, что $x(t) \in \mathbb{R}^n$, $n = \sum\limits_{k=1}^m n_k$, и $u(t) \in \mathbb{R}^r$, $r = \sum\limits_{k=1}^m r_k$, $t \in T$. Далее, векторы $d \in \mathbb{R}^n$ и $x_0 \in \mathbb{R}^n$, квазидиагональная матрица D [19] и вектор-функция $c(\cdot)$ строятся следующим образом: $c(t)^\top \triangleq \big[c_1(t)^\top, c_2(t)^\top, \dots, c_m(t)^\top\big] \in \mathbb{R}^n$, $t \in T$, $x_0^\top = \Big(x_1^{0\,\top}, x_2^{0\,\top}, \dots, x_m^{0\,\top}\Big)$, $d^\top \triangleq \big(d_1^\top, d_2^\top, \dots, d_m^\top\big)$, $D \triangleq \operatorname{diag}\big\{D_1, D_2, \dots, D_m\big\}$. Кроме того, блочно-диагональные матричные функции $A(\cdot)$ и $B(\cdot)$ определены правилом: $A(t) = \operatorname{diag}\big\{A_1(t), A_2(t), \dots, A_m(t)\big\}$, $B(t) = \operatorname{diag}\big\{B_1(t), B_2(t), \dots, B_m(t)\big\}$, а множество U конструируется с помощью декартова произведения $U \triangleq U_1 \times U_2 \times \ldots \times U_m$.

Таким образом, задача $(\mathcal{P}_0)$ представляет собой объединение (или составляется из) m задач-ядер (P_k), $k = 1, 2, \dots, m$.

Предложение 1. 1) *Пусть процесс* $\big(x_k(\cdot), u_k(\cdot)\big)$ *удовлетворяет ПМП в задаче* $(\mathcal{P}_k)$ *для значений* $k = 1, 2, \dots, m$. *Тогда управляемый процесс* $\big(x(\cdot), u(\cdot)\big)$, *где* $t \in T$, $x(t)^\top = \big[x_1(t)^\top, x_2(t)^\top, \dots, x_m(t)^\top\big]$, $u(t)^\top = \big[u_1(t)^\top, u_2(t)^\top, \dots, u_m(t)^\top\big]$ *удовлетворяет ПМП в задаче* (P_0).

2) *Если процесс* $\big(z_k(\cdot), w_k(\cdot)\big)$ *является глобально (локально) оптимальным в задаче* $(\mathcal{P}_k)$ *для значений* $k = 1, 2, \dots, m$, *то управляемый процесс* $\big(z(\cdot), w(\cdot)\big)$, *где* $t \in T$, $z(t)^\top = \big[z_1(t)^\top, z_2(t)^\top, \dots, z_m(t)^\top\big]$, $w(t)^\top = \big[w_1(t)^\top, w_2(t)^\top, \dots, w_m(t)^\top\big]$, *оказывается глобально (локально) оптимальным в задаче* (P_0).

Доказательство. а) Прежде всего отметим, что для каждого управления $u(t)$, $t \in T$, такого, что $u(t)^\top = \big[u_1(t)^\top, u_2(t)^\top, \dots, u_m(t)^\top\big]$, $u_k(t) \in \mathbb{R}^{r_k}$, $k = 1, 2, \dots, m$, целевой функционал $J(u)$ задачи $(\mathcal{P})$ представим в виде суммы функционалов соответствующих задач-ядер

$$
J(u) = \big\langle d, x(t_1)\big\rangle + \frac{1}{2}\big\langle x(t_1), Dx(t_1)\big\rangle = \sum_{k=1}^m \Big\{ \big\langle d_k, x_k(t_1)\big\rangle + \frac{1}{2}\big\langle x_k(t_1), D_k x_k(t_1)\big\rangle \Big\} \triangleq \sum_{k=1}^m J_k(u_k). \tag{4}
$$

б) Пусть процесс $\big(x_k(\cdot), u_k(\cdot)\big)$ удовлетворяет ПМП в задаче $(\mathcal{P}_k)$, $k = 1, 2, \dots, m$, т.е.

$$
\big\langle B_k(t)^\top \psi_k(t), u_k(t)\big\rangle \geqslant \Big\langle B_k(t)^\top \psi_k(t), v_k\Big\rangle \quad \forall v_k \in U_k, \quad \overset{\circ}{\forall} t \in T, \tag{5}
$$

где сопряженная функция $\psi_k(\cdot)$ будет являться решением следующей задачи Коши $\dot{\psi}_k(t) = -A_k(t)^\top \psi_k(t)$, $\psi_k(t_1) = -D_k x_k(t_1) - d_k$.

Отметим, что вектор-функция $\psi(t) = \big[\psi_1(t)^\top, \psi_2(t)^\top, \dots, \psi_m(t)^\top\big]^\top$ удовлетворяет сопряженной системе ОДУ $\dot{\psi}(t) = -A(t)^\top \psi(t)$, $\psi_k(t_1) = -Dx(t_1) - d_k$. Из (5) вытекает, что $\forall v = \Big[v_1^\top, v_2^\top, \dots, v_m^\top\Big]^\top \in U$

$$
\big\langle B(t)^\top \psi(t), u(t)\big\rangle = \sum_{k=1}^m \Big[\big\langle B_k(t)^\top \psi_k(t), u_k(t)\big\rangle\Big] \geqslant \sum_{k=1}^m \Big[\big\langle B_k(t)^\top \psi_k(t), v_k\big\rangle\Big] = \Big\langle B(t)^\top \psi(t), v\Big\rangle, \overset{\circ}{\forall} t \in T.
$$

Таким образом, процесс $\big(x(\cdot), u(\cdot)\big)$ удовлетворяет ПМП в задаче $(\mathcal{P}_0)$.

в) Если процесс $\big(z_k(\cdot), w_k(\cdot)\big)$ является глобально (локально) оптимальным в задаче $(\mathcal{P}_k)$ для значений $k = 1, 2, \dots, m$, то в силу (4) нетрудно видеть, что процесс $\big(z(\cdot), w(\cdot)\big)$ является глобально (локально) оптимальным в задаче $(\mathcal{P}_0)$. Что и требовалось доказать.

Согласно предложению 1, любая комбинация глобально (локально) оптимальных или критических (ПМП) процессов задач-ядер $(\mathcal{P}_i)$ дает глобально (локально) оптимальный или критический процесс в задаче $(\mathcal{P}_0)$ соответственно. Поэтому из предложения 1 вытекает

Следствие 1. *Пусть в задаче* $(\mathcal{P}_i)$, $i = 1, 2, \ldots, m$, *существует* K_i *и* L_i *глобально и локально оптимальных процессов соответственно, а также* M_i *процессов, удовлетворяющих ПМП. Тогда в задаче* $(\mathcal{P}_0)$ *существует* $\prod\limits_{i=1}^{m} K_i$ *и* $\prod\limits_{i=1}^{m} L_i$ *глобально и локально оптимальных процессов соответственно и* $\prod\limits_{i=1}^{m} M_i$ *процессов, удовлетворяющих ПМП.*

Следует отметить, что задача $(\mathcal{P}_0)$, построенная из задач-ядер $(\mathcal{P}_k)$, $k = 1, 2, \ldots, m$, по вышеописанному правилу, обладает следующей особенностью. Поскольку матрица D и матричные функции $A(\cdot)$ и $B(\cdot)$ имеют блочно-диагональную структуру [19], они имеют большое количество нулевых элементов и, как следствие, являются разреженными (например, см. [19]). Это может облегчать работу тестируемых алгоритмов. Для того, чтобы этого не произошло, необходимо трансформировать матрицы в менее разреженные, что и делается ниже.

Пусть задана невырожденная ортогональная симметричная матрица $H \in \mathcal{M}_{n,n}(\mathbb{R})$. Это означает, что $H = H^{-1} = H^{\top}$, $\det H \neq 0$. Рассмотрим следующую задачу $(\mathcal{P}_H)$:

$$J_H(u) = \big\langle d_H, x(t_1)\big\rangle + \frac{1}{2}\big\langle x(t_1), D_H x(t_1)\big\rangle \downarrow \min_u,$$

$$\dot{x}(t) = A_H(t)x(t) + B_H(t)u(t) + c_H(t), \quad x(t_0) = x_0^H, \tag{6}$$

$$u(\cdot) \in \mathcal{U} = \Big\{ u(\cdot) \in L_\infty^r(T) \mid u \in U \quad \overset{\circ}{\forall} t \in T \Big\}, \tag{7}$$

где векторы d_H, x_0^H, матрица D_H, вектор-функция $c_H(t)$, $t \in T$, а также матричные функции $A_H(t)$, $B_H(t)$, $t \in T$, строятся по следующим правилам: $d_H = Hd$, $x_0^H = Hx_0$, $D_H = HDH$, $c_H(t) = Hc(t)$, $A_H(t) = HA(t)H$, $B_H(t) = HB(t)$.

Предложение 2. 1) *Если управляемый процесс* $\big(\widehat{z}(\cdot), \widehat{w}(\cdot)\big)$ *является глобально (локально) оптимальным в задаче* $(\mathcal{P}_0)$, *то процесс* $\big(z(\cdot), w(\cdot)\big)$, *где* $z(t) = H\widehat{z}(t)$, $w(t) = \widehat{w}(t)$, $t \in T$, *является глобально (локально) оптимальным в задаче* $(\mathcal{P}_H)$. *Наоборот, если* $\big(z(\cdot), w(\cdot)\big)$ *является глобально (локально) оптимальным процессом в задаче* $(\mathcal{P}_H)$, *то процесс* $\big(z_0(\cdot), w_0(\cdot)\big)$, *где* $\widehat{z}(t) = Hz(t)$, $\widehat{w}(t) = w(t)$, $t \in T$, *является глобально (локально) оптимальным в задаче* $(\mathcal{P}_0)$.

2) *Если процесс* $\big(\widehat{x}(\cdot), \widehat{u}(\cdot)\big)$ *удовлетворяет ПМП в задаче* $(\mathcal{P}_0)$, *то процесс* $\big(x(\cdot), u(\cdot)\big)$, *где* $t \in T$, $x(t) = H\widehat{x}(t)$, $u(t) = \widehat{u}(t)$, *удовлетворяет ПМП в задаче* $(\mathcal{P}_H)$. *И наоборот, если процесс* $\big(x(\cdot), u(\cdot)\big)$ *удовлетворяет ПМП в задаче* $(\mathcal{P}_H)$, *то процесс* $\big(\widehat{x}(\cdot), \widehat{u}(\cdot)\big)$, *где* $\widehat{x}(t) = Hx(t)$, $\widehat{u}(t) = u(t)$, $t \in T$, *удовлетворяет ПМП в задаче* $(\mathcal{P}_0)$.

3) *Значения задач* $(\mathcal{P}_0)$ *и* $(\mathcal{P}_H)$ *совпадают:* $\inf\limits_u \big\{ J(u) \mid u \in \mathcal{U} \big\} = \inf\limits_u \big\{ J_H(u) \mid u \in \mathcal{U} \big\}$.

Доказательство. Сначала покажем, что если процесс $\big(\widehat{x}(\cdot), \widehat{u}(\cdot)\big)$ является допустимым в задаче $(\mathcal{P}_0)$, то процесс $\big(x(\cdot), u(\cdot)\big)$, где $x(t) = H\widehat{x}(t)$, $t \in T$, является допустимым в задаче $(\mathcal{P}_H)$. Действительно, поскольку $\widehat{x}(t) = \widehat{x}(t_0) + \int\limits_{t_0}^{t} \big[A(\tau)\widehat{x}(\tau) + B(\tau)\widehat{u}(\tau) + c(\tau)\big]\, d\tau$, то, подставляя в равенство $\widehat{x}(t) = Hx(t)$, $t \in T$, получаем $Hx(t) = Hx(t_0) + \int\limits_{t_0}^{t} \big[A(\tau)Hx(\tau) + B(\tau)u_0(\tau) + c(\tau)\big]\, d\tau$. Умножая теперь обе части равенства на матрицу H слева, приходим к равенству $x(t) = x(t_0) + \int\limits_{t_0}^{t} \big[A_H(\tau)x(\tau) + B_H(\tau)u_0(\tau) + c_H(\tau)\big]\, d\tau$, из которого вытекает, что процесс $\big(x(\cdot), u(\cdot)\big)$ является допустимым в задаче $(\mathcal{P}_H)$.

Далее нетрудно проверить равенство:

$$J_H(u) = \big\langle d_H, x(t_1)\big\rangle + \frac{1}{2}\big\langle x(t_1), D_H x(t_1)\big\rangle = \big\langle d_H, H\widehat{x}(t_1)\big\rangle + \frac{1}{2}\big\langle H\widehat{x}(t_1), D_H H\widehat{x}(t_1)\big\rangle = J(\widehat{u}).$$

Отсюда следует, что процесс управления $\big(\widehat{z}(\cdot), \widehat{w}(\cdot)\big)$ является глобально оптимальным в задаче $(\mathcal{P}_0)$, т.е. $J(\widehat{w}) \leqslant J(u)\ \forall\, u \in \mathcal{U}$, тогда и только тогда, когда процесс $\big(z(\cdot), w(\cdot)\big)$, где $z(t) = H\widehat{z}(t)$, $w(t) = \widehat{w}(t)$, $t \in T$, доставляет минимум целевому функционалу $J_H(\cdot)$: $J(w) = J(\widehat{w}) \leqslant J(u)\ \forall\, u \in \mathcal{U}$. Что и требовалось доказать.

Следствие 2. 1) *Количество процессов, удовлетворяющих ПМП в задаче* $(\mathcal{P}_H)$, *равно количеству процессов, удовлетворяющих ПМП в задаче* $(\mathcal{P}_0)$.

2) *Количество глобально (локально) оптимальных процессов в задачах* $(\mathcal{P}_H)$ *и* $(\mathcal{P}_0)$ *соответственно равны.*

Замечание. Как известно из [20], симметричную ортогональную матрицу H можно построить с помощью некоторого вектора $y \in \mathbb{R}^n$посредством формулы $H := E - \dfrac{2}{\langle y, y\rangle}\, yy^\top$. Отметим, что $yy^\top = \{y_{ij}\}$, где $y_{ij} = y_i y_j$. Данную матрицу принято называть матрицей отражения [20], поскольку с геометрической точки зрения преобразование H означает симметричное отражение относительно вектора y. Кроме того, поскольку $D_H = H^{-1}DH$, то собственные значения матриц D_H и D совпадают.

3. Тестовые примеры. Ниже будет представлен процесс генерации задач вида

$$J(u) = \frac{1}{2}\,\big\langle x(t_1,u), Dx(t_1,u)\big\rangle \downarrow \min, \quad u \in \mathcal{U}, \tag{8}$$

над линейной системой управления (1), (2). Пусть множество ограничений на управление представляет собой декартово произведение $U \triangleq \prod\limits_{i=1}^{r} [u_-, u_+]$, где $u_- \in \mathbb{R}$ и $u_+ \in \mathbb{R}$ — параметры, задающие параллелепипедные ограничения на компоненты управления.

В качестве задач-ядер в работе предлагается использовать следующие задачи вида (8) со знаконеопределенной симметричной матрицей D_k, $D_k = D_k^\top$, $k = 1, 2, 3, 4$. Введем обозначения. Пусть S_k — задача управления системой, D_k — матрица целевого функционала, $k = 1, 2, 3, 4$.

Задача-ядро № 1:

$$S_1 = \begin{cases} \dot x_1(t) = u_1(t), & x_1(0) = 0, \\ \dot x_2(t) = u_2(t), & x_2(0) = 0, \\ -1 \leqslant u_i(t) \leqslant 2, & T = [0,1], \end{cases} \qquad D_1 = \begin{pmatrix} -3 & 1 \\ 1 & 1 \end{pmatrix}.$$

При этом три управления удовлетворяют ПМП и только одно управление $\overline{u}(t) \equiv (2, -1)$ доставляет глобальный минимум целевому функционалу: $J(\overline{u}) = -7.5$.

Задача-ядро № 2:

$$S_2 = \begin{cases} \dot x_1(t) = u_1(t), & x_1(0) = 0, \\ \dot x_2(t) = u_2(t), & x_2(0) = 0, \\ -1 \leqslant u_i(t) \leqslant 2, & T = [0,1], \end{cases} \qquad D_2 = \begin{pmatrix} 2 & 3 \\ 3 & 1 \end{pmatrix}.$$

В данной задаче существуют три процесса, удовлетворяющих ПМП, и среди них лишь управление $\overline{u}(t) \equiv (-1, 2)$ является глобально оптимальным. Значение задачи при этом равно $J(\overline{u}) = -3$.

Задача-ядро № 3:

$$S_3 = \begin{cases} \dot x_1(t) = x_2(t), & x_1(0) = 0, \\ \dot x_2(t) = u(t), & x_2(0) = 2, \\ -1 \leqslant u(t) \leqslant 2, & T = [0,1], \end{cases} \qquad D_3 = \begin{pmatrix} 5 & -2 \\ -2 & -1 \end{pmatrix}.$$

В данной задаче два управляемых процесса удовлетворяют ПМП, и только один из них является глобально оптимальным.

Задача-ядро № 4:

$$S_4 = \begin{cases} \dot x_1(t) = x_2(t) + u_1(t), & x_1(0) = 0.25, \\ \dot x_2(t) = u_2(t), & x_2(0) = 0, \\ -1 \leqslant u(t) \leqslant 2, & T = [0,1], \end{cases} \qquad D_4 = \begin{pmatrix} 7 & -3 \\ -3 & 1 \end{pmatrix}.$$

В задаче-ядре 4-го типа количество процессов управления, удовлетворяющих ПМП, равно 2 и среди них только один является глобально оптимальным.

4. Решение линеаризованных задач. Рассмотрим следующую выпуклую задачу ОУ:

$$I(u) \triangleq \varphi\big(x(t_1)\big) = \frac{1}{2}\,\big\langle x(t_1), Cx(t_1)\big\rangle - \big\langle d, x(t_1)\big\rangle \downarrow \min_u, \tag{9}$$

над системой управления (6), (7), где симметричная матрица $C \in \mathcal{M}_{n,n}(\mathbb{R})$ является положительно определенной ($C = C^\top > 0$).

Для решения этой задачи воспользуемся следующими итерационными методами: методом приращений (МП, [14]), методом условного градиента (МУГ, [2, 4, 9, 15]) и методом игольчатой вариации (МИВ, [15]). Кратко напомним схемы используемых методов при решении задачи (9), (6), (7).

4.1. Схема метода приращений. Пусть задан некоторый параметр метода $\alpha > 0$. Найдем матричную функцию $\Psi(t)$, $t \in T$, интегрируя систему $\dot{\Psi} = -A(t)^\top \Psi - \Psi A(t)$, $\Psi(t_1) = -C$.

Пусть задано некоторое начальное управление $u_0(\cdot) \in \mathcal{U}$. И пусть P_U — оператор проектирования на множество U.

Шаг 0а. *Инициализация метода.* $m := 0$, $u^m(t) := u_0(t)$, $t \in T$.

Шаг 0б. Найти состояние $x^m(t) = x(t, u^m)$, $t \in T$.

Шаг 1. Пусть вектор-функция $w^m(\cdot,\cdot) : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^r$ определена следующим образом: $w^m(p,t) \triangleq P_U\big(u^m(t) + \alpha B(t)^\top p\big)$. Найти решение $p^m(t)$, $t \in T$, системы ОДУ

$$\dot{p} = -A(t)^\top p - \Psi(t)B(t)\big(w^m(p,t) - u^m(t)\big), \quad p(t_1) = d - Cx^m(t_1). \tag{10}$$

Шаг 2. Найти промежуточное управление $\bar{u}^m(t) := w^m\big(p^m(t), t\big)$, $t \in T$, соответствующее найденной функции $p^m(\cdot)$.

Шаг 3. Пусть вектор-функция $v^m(\cdot,\cdot) : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^r$ определена по следующей формуле: $v^m(x,t) \triangleq P_U\big(\bar{u}^m(t) + \alpha B(t)^\top q^m(x,t)\big)$, где вектор-функция $q^m(\cdot,\cdot) : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ такова, что $q^m(x,t) \triangleq p^m(t) + \Psi(t)\big(x - x^m(t)\big)$. Найти решение $x^{m+1}(t)$, $t \in T$, системы ОДУ

$$\dot{x} = A(t)x + B(t)v^m(x,t) + c(t), \quad x(t_0) = x_0. \tag{11}$$

Шаг 4. Вычислить управление, соответствующее найденной траектории $x^m(t)$:

$$u^{m+1}(t) := v^m\big(x^{m+1}(t), t\big), \quad t \in T.$$

Шаг 5. $m = m + 1$ и перейти на Шаг 1.

Замечание. Отметим, что оператор проектирования P_U удовлетворяет условию Липшица [4]:

$$\big\|P_U(u) - P_U(v)\big\| \leqslant \|u - v\| \quad \forall u, v \in \mathbb{R}^r.$$

Следовательно, оператор P_U является непрерывным. Это означает, что задачи Коши (10) и (11) имеют единственное решение.

В качестве критерия останова метода можно использовать [28] $I(u^m) - I(\bar{u}^m) \leqslant \varepsilon$, где ε — заданный параметр точности решения [14].

Далее напомним описание двух других методов, которые, в отличие от метода приращений, не используют матрицу вторых производных $\nabla^2\varphi(x(t_1)) = C$ целевого функционала.

4.2. Схема метода условного градиента (например, см. [4, 13–15]).

Пусть ε — заданный параметр точности.

Шаг 0. *Инициализация метода.* $m := 0$, $u^m(t) := u_0(t)$, $t \in T$. Интегрируя систему управления (6), (7), найти состояние $x^m(t) = x(t, u^m)$, $t \in T$.

Шаг 1. Решить сопряженную задачу Коши:

$$\begin{cases} \dot{\psi}^m = -A(t)^\top \psi^m, \quad t \in T, \\ \psi^m(t_1) = d - Cx^m(t_1) \triangleq -\nabla\varphi\big(x^m(t_1)\big). \end{cases}$$

Шаг 2. Построить вспомогательное управление: $\overline{u}^{\,m}(t) = \operatorname*{argmax}_{v \in U} \big\langle B^\top(t)\psi^m(t), v\big\rangle$, $t \in T$. При этом

$$\overline{u}_i^m(t) = \begin{cases} u_+, & \big[B^\top(t)\psi^m(t)\big] > 0, \\ u_-, & \big[B^\top(t)\psi^m(t)\big] < 0 \quad \forall i = 1, 2, \ldots, r, \\ u_i^m(t), & \big[B^\top(t)\psi^m(t)\big] = 0. \end{cases}$$

Шаг 3. Сформировать $\eta_m(t) := \big\langle B^\top(t)\psi^m(t), \overline{u}^{\,m}(t) - u^m(t)\big\rangle$ и проинтегрировать ее: $\eta_m := \int\limits_T \eta_m(t)\,dt$.

Шаг 4. *Критерий останова.* Если $\eta_m \leqslant \varepsilon$, то STOP.

Шаг 5. Найти состояние $\overline{x}^m(t) = x(t, \overline{u}^m)$, $t \in T$, решая задачу Коши

$$\dot{\overline{x}}^m(t) = A(t)\overline{x}^m(t) + B(t)\overline{u}^m(t) + c(t), \quad \overline{x}^m(t_0) = x_0.$$

Шаг 6. Сформировать управление посредством выпуклой комбинации

$$u^m(t, \alpha) = u^m(t) + \alpha(\overline{u}^m(t) - u^m(t)), \quad \alpha \in]0, 1], \quad t \in T,$$

и найти α_m из условия скорейшего спуска как решение задачи $I\big(u(\cdot, \alpha)\big) \downarrow \min$, $\alpha \in [0, 1]$. Другими словами, вычисляем $\alpha_m = \min\big\{\overline{\alpha}_m, 1\big\}$, где $\overline{\alpha}_m = \dfrac{\big\langle x^m(t_1) - \overline{x}^m(t_1), D_1 x^m(t_1)\big\rangle}{\big\langle \overline{x}^m(t_1) - x^m(t_1), \overline{x}^m(t_1) - x^m(t_1)\big\rangle} > 0$.

Шаг 7. Положить $u^{m+1}(t) := u^m(t, \alpha_m)$, $t \in T$, в силу линейности системы управления получить $x^{m+1}(t_1) := (1 - \alpha_m)x^m(t_1) + \alpha_m \overline{x}^m(t_1)$, $m := m + 1$ и вернуться на шаг 1.

Замечание. Отметим, что при выполнении критерия останова $\eta_m \leqslant \varepsilon$ управление $u^m \in \mathcal{U}$ является приближенно оптимальным, поскольку нетрудно видеть, что $\forall\, u \in \mathcal{U}$ выполняется цепочка

$$-\varepsilon \leqslant -\int\limits_T \eta_m(t)\, dt \triangleq \int\limits_T \big\langle B^\top(t)\psi^m(t), u^m(t) - \overline{u}^m(t)\big\rangle\, dt \leqslant \int\limits_T \big\langle B^\top(t)\psi^m(t), u^m(t) - u(t)\big\rangle\, dt =$$

$$= \int\limits_T \Big\langle \psi^m(t), b\big(u^m(t), t\big) - b(u(t), t)\Big\rangle\, dt = \Big\langle \nabla\varphi\big(x^m(t_1)\big), x(t_1, u) - x^m(t_1)\Big\rangle.$$

Отсюда в силу выпуклости функции $\varphi(\cdot)$ получаем неравенство $-\varepsilon \leqslant \varphi\big(x(t_1, u)\big) - \varphi\big(x^m(t_1)\big)$ $\forall\, u \in \mathcal{U}$, или $\varphi\big(x^m(t_1)\big) \leqslant \varphi\big(x(t_1, u)\big) + \varepsilon$ $\forall\, u \in \mathcal{U}$.

4.3. Схема метода игольчатой вариации [2 4, 8–10, 14, 15].

Шаг 0–Шаг 4 метода игольчатой вариации и метода условного градиента совпадают. Поэтому описание метода начнем непосредственно с шага 5.

Шаг 5. Найти величину $M_m = \max\limits_{t \in T} \eta_m(t) > 0$. Отметим, что $M_m \geqslant \dfrac{\eta_m}{\operatorname{mes} T}$.

Шаг 6. Выбрать параметр $\lambda_m \in \,]0, 1[$ и определить множество $T_m \subset T$, $\operatorname{mes} T_m > 0$, из условия $\eta_m(t) \geqslant \lambda_m M_m$ $\forall\, t \in T_m$.

Шаг 7. Построить управление $\widehat{u}^m(\cdot)$ по формуле $\widehat{u}^m(t) = \overline{u}^m(t)$ при $t \in T_m$ и $\widehat{u}^m(t) = u^m(t)$ при $t \notin T_m$.

Шаг 8. Найти состояние $\widehat{x}^m(t) = x\big(t, \widehat{u}^m\big)$, $t \in T$.

Шаг 9. Положить $u^{m+1}(t) := \widehat{u}^m(t)$, $t \in T$, $x^{m+1}(t_1) := \widehat{x}^m(t_1)$, $m := m + 1$ и вернуться на шаг 1.

Заметим, что параметр λ_m на шаге 6 метода игольчатой вариации находится как решение задачи

$$I(u_\lambda) \downarrow \min, \quad \lambda \in [\lambda_{\min}, \lambda_{\max}]. \tag{12}$$

Следует отметить недостатки вышеописанных методов. Несмотря на то что МП учитывает матрицу вторых производных функции, задающей целевой функционал, для его применения необходимо на каждой итерации интегрировать нелинейную систему ОДУ, в то время как решается задача ОУ с линейной системой (см. (1), (2)). В качестве недостатков МУГ и МИВ можно отметить низкую скорость работы метода вблизи решения задачи. Также следует отметить, что, используя МИВ, приходится многократно решать задачи одномерной минимизации.

Для проведения дальнейшего численного эксперимента были написаны программы на языке программирования C++, реализующие методы приращений, условного градиента и игольчатой вариации. При этом для решения задачи безусловной минимизации (12) использовался алгоритм золотого сечения [4, 5]. Для численного решения задач Коши для прямой и сопряженной систем обыкновенных дифференциальных уравнений была применена разностная схема Рунге–Кутта второго порядка точности с коэффициентами $\alpha_2 = 1$, $\beta_{21} = 1$, $p_1 = p_2 = \dfrac{1}{2}$, которые характеризуют расчетную формулу [20]. Для рассматриваемых тестовых задач было выбрано 100 точек дискретизации (узлов сетки) отрезка $[t_0, t_1]$, что обеспечивает точность интегрирования систем порядка $O\left(\left(\dfrac{t_1 - t_0}{100}\right)^2\right)$.

5. Локальный поиск. Напомним, что метод локального поиска для задачи $(\mathcal{P})$ заключается в последовательном решении выпуклых (частично линеаризованных) задач.

Пусть задано некоторое начальное управление $u^0 \in \mathcal{U}$. Далее, если известно допустимое управление $u^s \in \mathcal{U}$, то следующая итерация $u^{s+1} \in \mathcal{U}$ строится как приближенное решение линеаризованной задачи

$$(\mathcal{P}L_s): \qquad I_s(u) \triangleq g\big(x(t_1,u)\big) - \Big\langle \nabla h\big(x^s(t_1)\big), x(t_1,u)\Big\rangle \downarrow \min_u,$$

над системой управления (6) и (7). Как и выше, через $x^s(\cdot) = x(\cdot, u^s)$ обозначено состояние, соответствующее управлению $u^s(\cdot)$.

Пусть задана числовая последовательность $\{\delta_s\}$ такая, что $\delta_s > 0$, $s = 0, 1, 2, \ldots$, $\sum\limits_{s=0}^{\infty} \delta_s < +\infty$.

Согласно правилу, описанному выше, управление $u^{s+1} \in \mathcal{U}$ удовлетворяет условию

$$g\big(x^{s+1}(t_1)\big) - \Big\langle \nabla h\big(x^s(t_1)\big), x^{s+1}(t_1)\Big\rangle - \delta_s \leqslant \inf_u \Big\{ g\big(x(t_1,u)\big) - \Big\langle \nabla h\big(x^s(t_1)\big), x(t_1,u)\Big\rangle \mid u \in \mathcal{U} \Big\},$$

или, что тоже самое, $I_s\big(u^{s+1}\big) - \delta_s \leqslant \inf\limits_u \big\{ I_s(u) \mid u \in \mathcal{U} \big\}$.

В качестве критериев останова можно использовать следующее условие: $I\big(u^s\big) - I\big(u^{s+1}\big) \leqslant \tau$. Более подробно о критериях останова данного метода локального поиска написано в [28]. Тестирование алгоритма локального поиска проводилось при параметре точности $\tau = 10^{-2}$.

Численное тестирование алгоритма локального поиска в задаче $(\mathcal{P})$ было проведено на задачах размерности от 2×2 до 20×20, построенных посредством процедуры генерации тестовых задач из раздела 3. При этом для построения матрицы преобразования H вектор y выбирался случайным образом с равномерным распределением в интервале $]-10, 10[$. Для тестирования алгоритма локального поиска было сгенерировано более 50 задач с различным выбором количества задач-ядер каждого класса. В табл. 1 частично приведены результаты данного тестирования.

Алгоритм локального поиска был реализован на языке программирования C++. Все расчеты проводились на компьютере Intel Core 2 Duo 2.0 ГГц с оперативной памятью объемом 2 Гб.

В качестве стартовых было выбрано несколько управлений с целью выбрать наихудшее для глобального поиска, стартуя откуда, локальный поиск доставляет управление, самое далекое по целевому функционалу от глобально оптимального.

В табл. 1 использовались следующие обозначения: № — номер тестовой задачи; m_1, m_2, m_3, m_4 — количество задач-ядер, используемых при генерации тестовых задач, 1-го–4-го типов соответственно; n и r — размерность задач серии по состоянию и управлению соответственно; J_0 — значение целевого функционала в начале работы метода локального поиска; J_{st} — значение целевого функционала в момент останова метода; Time — время работы метода локального поиска (мин : сек).

Отметим, что при тестировании алгоритма локального поиска в задачах наименьшей размерности (2×2), стартуя с выбранного управления, был найден глобально оптимальный процесс управления. Это объясняется тем, что согласно вышеописанному методу генерации тестовых задач (см. следствие 1), в построенных тестовых примерах с ростом размерности наблюдается экспоненциальный рост количества управляемых процессов, удовлетворяющих ПМП, но не являющихся при этом глобально оптимальными. Напомним, что в тестовых примерах, сгенерированных с помощью m_i задач-ядер i-го типа, $i = \overline{1,4}$, по построению существует $3^{m_1+m_2} \times 2^{m_3+m_4}$ процессов, удовлетворяющих ПМП, и, подчеркнем, только один глобально оптимальный (см. следствия предложений 1 и 2). Например, в задачах размерности 20×20, построенных из задач-ядер 1-го и 2-го типа, количество критических процессов, не являющихся глобально оптимальными, составляет $3^{10} - 1 = 59048$.

Во время проведения эксперимента сравнивались три варианта алгоритма локального поиска, в которых частично линеаризованные (выпуклые) задачи решаются методом приращений (МП), методом условного градиента (МУГ) и методом игольчатой вариации (МИВ) соответственно. Из результатов численного тестирования видно, что наиболее эффективным по времени работы оказался алгоритм, использующий метод приращений из [14]. Поскольку для глобального поиска именно время решения линеаризованных задач является наиболее важным показателем, то данный алгоритм будем использовать при поиске глобально оптимальных процессов управления.

6. Глобальный поиск. Как показано выше, локальный поиск не обеспечивает, вообще говоря, достижения решения даже в задачах небольшой размерности. Поэтому для решения невыпуклых задач в [28] был предложен алгоритм глобального поиска, состоящий фактически из двух основных этапов:

1) процедуры локального поиска,

2) процедуры выхода из процесса, удовлетворяющего ПМП.

Таблица 1

Результаты тестирования метода локального поиска

№	m_1	m_2	m_3	m_4	n	r	J_0	МП		МУГ		МИВ	
								J_{st}	Time	J_{st}	Time	J_{st}	Time
1	1	1	0	0	4	4	-1.54	-3.50	0:02.28	-3.50	0:15.59	-3.50	0:11.27
2	0	1	1	0	4	3	2.29	-3.18	0:02.56	-3.18	0:18.26	-3.19	0:15.58
3	0	0	1	1	4	3	-2.46	-5.19	0:02.01	-5.18	0:14.60	-5.19	0:17.11
4	1	0	0	1	4	4	2.78	-1.93	0:01.59	-1.93	0:17.12	-1.93	0:12.48
5	2	1	0	0	6	6	4.23	-5.50	0:04.26	-5.50	0:25.47	-5.50	0:26.24
6	0	2	0	1	6	6	3.47	-2.78	0:06.67	-2.78	0:28.18	-2.79	0:21.42
7	1	0	2	0	6	4	4.15	-8.93	0:05.33	-8.92	0:33.29	-8.92	0:31.18
8	0	0	1	2	6	5	-2.39	-9.62	0:05.70	-9.62	0:39.16	-9.62	0:15.89
9	3	1	1	0	10	9	-3.18	-12.28	0:10.41	-12.29	0:51.62	-12.27	0:55.32
10	2	1	2	0	10	8	-2.75	-9.36	0:11.36	-9.36	0:47.70	-9.36	0:33.18
11	1	1	1	2	10	9	3.16	-5.40	0:09.18	-5.40	0:48.19	-5.40	0:55.56
12	0	2	2	1	10	8	-5.11	-9.73	0:12.17	-9.74	0:55.11	-9.73	0:32.82
13	3	2	2	0	14	12	-6.57	-12.60	0:15.40	-12.61	1:01.12	-12.61	0:59.31
14	3	1	1	2	14	13	-5.95	-11.18	0:17.56	-11.18	1:03.26	-11.18	1:12.39
15	2	2	2	1	14	12	-3.12	-8.72	0:18.41	-8.73	0:59.75	-8.72	1:15.14
16	1	1	4	1	14	10	-4.59	-9.82	0:17.93	-9.82	1:12.32	-9.82	1:01.19
17	3	2	2	3	20	18	-2.57	-10.30	0:25.43	-10.29	1:47.77	-10.30	2:03.23
18	3	2	3	2	20	17	-9.18	-14.91	0:20.41	-14.91	2:18.45	-14.91	1:17.28
19	2	5	2	1	20	18	-6.36	-15.84	0:17.92	-15.85	1:59.18	-15.84	1:38.19
20	3	2	4	1	20	16	-5.18	-14.21	0:24.56	-14.22	2:35.62	-14.22	2:15.17
21	3	3	0	4	20	20	3.12	-7.34	0:26.32	-7.34	2:12.41	-7.33	2:41.36

Этот алгоритм основан на теории глобального поиска, разработанной в [23]. Напомним основные этапы алгоритма глобального поиска.

6.1. Алгоритм глобального поиска. Пусть заданы начальное управление $u_0 \in \mathcal{U}$ и числовые последовательности $\{\tau_k\}$ и $\{\delta_k\}$: $\tau_k,\ \delta_k > 0,\ k = 0, 1, 2 \ldots,\ \tau_k \downarrow 0,\ \delta_k \downarrow 0\ (k \to \infty)$.

Шаг 0. *Инициализация.* Положить $k := 0$, $\overline{u}^k := u_0$.

Шаг 1. *Локальный поиск.* Исходя из управления $\overline{u}^k$, методом локального поиска построить τ_k-критический процесс управления $\big(z^k(\cdot), w^k(\cdot)\big)$: $J\big(\overline{u}^k\big) \geqslant J(w^k) = \zeta_k$.

Шаг 2. Выбрать число $\beta \in [\beta_-, \beta_+]$. В частности, процесс одномерного поиска по β можно начинать при $\beta_0 = g\big(z^k(t_1)\big)$.

Шаг 3. Построить некоторую аппроксимацию поверхности уровня

$$\mathcal{A}_k(\beta) = \big\{\, y^1, y^2, \ldots, y^{N_k} \mid h(y^i) = \beta - \zeta_k, \quad i = 1, 2, \ldots, N_k, \quad N_k = N_k(\beta) \,\big\}.$$

Шаг 4. В соответствии с [23, 28] сформировать множество индексов

$$I_k = I_k(\beta) = \big\{\, i \in \{\, 1, 2, \ldots, N_k \,\} \mid g(y^i) \leqslant \beta \,\big\}.$$

Шаг 5. Для каждого $i \in I_k$ найти управление $u^i \in \mathcal{U}$, такое что

$$g\big(x(t_1, u^i)\big) - \big\langle \nabla h(y^i), x(t_1, u^i) \big\rangle - \delta_k \leqslant \inf_u \Big\{ g\big(x(t_1, u)\big) - \big\langle \nabla h(y^i), x(t_1, u) \big\rangle \mid u \in \mathcal{U} \Big\}.$$

Шаг 6. Для каждого $i \in I_k$ решить задачу уровня с точностью δ_k, то есть найти точку $p^i \in \mathbb{R}^n$: $h(p^i) = \beta - \zeta_k$ такую, что

$$\big\langle \nabla h(p^i), x(t_1, u^i) - p^i \big\rangle + \delta_k \geqslant \sup_p \Big\{ \big\langle \nabla h(p), x(t_1, u^i) - p \big\rangle \mid h(p) = \beta - \zeta_k \Big\}.$$

Шаг 7. Положить $\eta_k(\beta) := \eta_k^0(\beta) + \beta$, где

$$\eta_k^0(\beta) := \big\langle \nabla h(p^j), x(t_1, u^j) - p^j \big\rangle - g\big(x(t_1, u^j)\big) \triangleq \max_{1 \leqslant i \leqslant N} \Big\{ \big\langle \nabla h(p^i), x(t_1, u^i) - p^i \big\rangle - g\big(x(t_1, u^i)\big) \Big\}.$$

Шаг 8. Если $\eta_k(\beta) > 0$, то положить $\overline{u}^{\,k+1} := u^j$, $k := k + 1$ и возвратиться на шаг 1.

Шаг 9. Если $\eta_k(\beta) \leqslant 0$, то выбрать новое $\beta := \beta + \Delta\beta$, $\beta \in [\beta_-, \beta_+]$ и возвратиться на шаг 3.

Шаг 10. Если $\eta_k(\beta) \leqslant 0 \; \forall \beta \in [\beta_-, \beta_+]$ (т.е. одномерный поиск по β завершен), то положить $\overline{u}^{k+1} := u^k$, $k := k + 1$. Если $\tau \leqslant \varepsilon$, то остановиться. В противном случае перейти на на шаг 1.

При этом следует отметить, что, как только условие оптимальности нарушается, можно производить переход на шаг 1 алгоритма глобального поиска [23, 28]. Данное упрощение позволяет, как показали численные эксперименты, получить значительный выигрыш в общем времени решения задачи. Поэтому далее будет проводиться численное тестирование с учетом этого замечания.

Нетрудно видеть, что представленный выше алгоритм глобального поиска не является алгоритмом в общепринятом смысле, поскольку не все его этапы конкретизированы. Прежде всего для вычисления промежутка одномерного поиска $[\beta_-, \beta_+]$ нужно решить две задачи: одну на минимум функционала $g(x(t_1))$, другую на максимум этого же функционала [23, 24, 28]. Первая задача может быть решена любым из методов, решающих выпуклые задачи ОУ. Для решения второй можно использовать алгоритм глобального поиска, разработанный в [25]. Однако следует отметить, что вычислительный процесс не требует точного знания этих границ: достаточно иметь лишь оценки (возможно, достаточно грубые).

Перейдем теперь к вопросу о построении аппроксимации (набора точек) поверхности уровня функции $h(\cdot)$ (см. шаг 3). Следует напомнить, что она должна быть достаточно репрезентативной для проверки, является ли текущий процесс управления глобальным решением. В противном случае аппроксимация должна позволять строить процесс лучший, чем исследуемый [23, 24, 28].

В данной работе в качестве аппроксимации поверхности уровня использовались следующие наборы:

$$\mathcal{R}_1 = \big\{\, y^{2j-1} = z(t_1) - \mu_{2j-1} e^{2j-1}, y^{2j} = z(t_1) + \mu_{2j} e^{2j} \mid j = 1, \ldots, n \,\big\},$$
$$\mathcal{R}_2 = \big\{\, y^{2j-1} = z(t_1) - \mu_{2j-1} [D]_j, y^{2j} = z(t_1) + \mu_{2j} [D]_j, \mid j = 1, \ldots, n \,\big\},$$

где $[D]_i$ — i-й вектор-столбец матрицы D, а e^i, $i = 1, 2, \ldots, n$, — единичные орты в пространстве $\mathbb{R}^n$. Числовые параметры $\mu_i > 0$, $i = 1, \ldots, 2n$, выбираются из условия $h(y^i) = \beta - \zeta_k$, $i = 1, 2, \ldots, 2n$.

Алгоритм глобального поиска будем останавливать при условии, что оценка $\eta_k(\beta) \leqslant 0$ для всех $\beta \in [\beta_-, \beta_+]$, а также критический процесс $\big(z(\cdot), w(\cdot)\big)$ получен на шаге 1 методом локального поиска с точностью $\tau \leqslant \varepsilon/2$, где заданный параметр точности $\varepsilon = 10^{-2}$.

При тестировании алгоритма глобального поиска был осуществлен поиск решений в сгенерированных ранее задачах размерности до 20×20 (см. раздел 5). Как было отмечено, во всех тестах, исключая задачи наименьшей размерности, локальный поиск не обеспечивает нахождения глобального решения с заданных начальных управлений.

Программа, реализующая алгоритм глобального поиска, была написана на языке программирования C++. В качестве подпрограмм, реализующих метод для решения линеаризованных (выпуклых) задач, использовалась подпрограмма, решающая задачу ОУ методом приращений, который хорошо зарекомендовал себя на предыдущем этапе вычислительного эксперимента при тестировании метода локального поиска.

Результаты численного эксперимента приведены в табл. 2, где сравниваются результаты работы алгоритма глобального поиска, использующего на шаге 3 аппроксимацию $\mathcal{R}_1$ и $\mathcal{R}_2$. Расчеты проводились на компьютере с теми же характеристиками, что и для метода локального поиска. При этом далеко не все результаты численных расчетов включены в табл. 2.

В таблице использовались следующие дополнительные к табл. 1 обозначения: Crit — количество процессов, удовлетворяющих ПМП, в задаче; St — количество процессов, удовлетворяющих ПМП, пройденных алгоритмом глобального поиска; Time — время работы программы; J_* — значение целевой функции задачи на процессе, найденном алгоритмом глобального поиска.

Анализируя результаты численного эксперимента, следует отметить, что с помощью алгоритма глобального поиска удалось получить глобальное решение во всех тестовых задачах, известное заранее в силу примененной методики построения тестовых примеров (см. предложения 1 и 2).

При сравнении эффективности вариантов алгоритма глобального поиска, использующих аппроксимации поверхности уровня $\mathcal{R}_1$ и $\mathcal{R}_2$, было выявлено, что, используя аппроксимацию $\mathcal{R}_1$, не во всех случаях был найден глобально оптимальный процесс, однако найти его позволила аппроксимация $\mathcal{R}_2$, и наоборот (выделено жирным шрифтом). Это означает, что для данного поля тестовых примеров объединение

Таблица 2

Результаты тестирования алгоритма глобального поиска

№	n	r	Crit	J_0	Аппроксимация $\mathcal{R}_1$			Аппроксимация $\mathcal{R}_2$		
					J_*	St	Time	J_*	St	Time
1	4	4	9	-1.63	-10.50	4	0:56.18	-10.50	5	0:59.67
2	4	3	6	2.29	-11.39	5	0:55.14	-11.39	4	0:49.28
3	4	3	4	-2.46	-9.54	3	0:46.92	-9.54	3	0:41.23
4	4	4	6	2.78	-8.72	5	1:01.03	-8.72	4	0:57.43
5	6	6	27	7.23	**-16.50**	10	1:26.24	-18.00	12	1:35.47
6	6	6	18	3.47	-19.43	11	1:39.27	-19.43	9	1:31.16
7	6	4	12	4.15	-12.10	10	1:15.86	-12.10	8	1:12.11
8	6	5	8	-2.39	-14.75	4	1:54.14	-14.75	5	1:37.19
9	10	9	162	-3.18	-21.89	15	2:19.32	-21.89	17	2:29.62
10	10	8	108	-2.75	-22.69	14	2:23.21	-22.69	16	2:32.15
11	10	9	72	3.16	-25.36	15	2:54.56	-25.36	15	2:47.29
12	10	8	72	-5.11	-28.03	16	2:43.10	-28.03	14	2:25.72
13	14	12	972	-6.57	-33.29	21	4:59.32	-33.29	20	4:51.10
14	14	13	648	-5.95	-32.36	20	5:18.12	-32.36	19	5:14.18
15	14	12	648	-3.12	-35.02	21	6:12.45	-35.02	23	6:57.92
16	14	10	288	-4.59	-33.02	17	5:37.11	-33.02	16	5:31.16
17	20	18	7776	-2.57	**-43.98**	25	10:32.15	-48.98	27	12:27.11
18	20	17	7776	-9.18	-48.07	28	11:10.45	-48.07	29	12:03.44
19	20	18	17496	-6.36	-56.33	37	18:53.19	**-52.33**	36	18:05.84
20	20	16	7776	-5.18	-47.14	25	11:24.47	-47.14	25	11:49.17
21	20	20	59048	-3.95	-63.19	49	17:19.54	-63.19	47	16:58.12

$\mathcal{R}_1 \cup \mathcal{R}_2$ оказывается разрешающим набором [28], поскольку во всех сгенерированных задачах было найдено глобальное решение.

7. Заключение. В работе предложен метод генерации линейно-квадратичных задач ОУ произвольной размерности с известными глобально и локально оптимальными процессами управления. Использование этого метода позволяет построить невыпуклые задачи ОУ, обладающие большим количеством процессов (≈ 60000 при размерности задачи 20×20), удовлетворяющих ПМП, но не являющихся при этом глобально оптимальными и единственным глобально оптимальным процессом.

На построенных невыпуклых линейно-квадратичных задачах ОУ протестированы разработанные алгоритмы локального и глобального поисков. Численное тестирование доказало эффективность предложенной методики на десятках сгенерированных задач до размерности 20×20. Во всех построенных задачах алгоритм глобального поиска нашел глобально оптимальный процесс.

СПИСОК ЛИТЕРАТУРЫ

1. *Понтрягин Л.С., Болтянский В.Г., Гамкрелидзе Р.В., Мищенко Е.Ф.* Математическая теория оптимальных процессов. М.: Наука, 1983.
2. *Черноусько Ф.Л., Баничук Н.В.* Вариационные задачи механики и управления. М.: Наука, 1973.
3. *Алексеев В.М., Тихомиров В.М., Фомин С.В.* Оптимальное управление. М.: Физматлит, 2005.
4. *Васильев Ф.П.* Методы оптимизации. М.: Факториал Пресс, 2002.
5. *Евтушенко Ю.Г., Жадан В.Г.* Численные методы оптимизации. М.: Наука, 2005.
6. *Ли Э., Маркус Л.* Основы теории оптимального управления. М.: Наука, 1972.
7. *Габасов Р.Ф., Кириллова Ф.М.* Оптимизация линейных систем. Минск: Изд-во Белорус. ун-та, 1973.
8. *Черноусько Ф.Л., Болотник Н.Н., Градецкий В.Г.* Манипуляционные роботы: динамика, управление, оптимизация. М.: Наука, 1989.
9. *Крылов И.А., Черноусько Ф.Л.* Алгоритм метода последовательных приближений для задач оптимального управления // Журн. вычисл. матем. и матем. физ. 1972. **12**, № 1. 14–34.
10. *Любушин А.А., Черноусько Ф.Л.* Метод последовательных приближений для расчета оптимального управления // Изв. АН СССР. Техн. кибернетика. 1983. № 2. 147–159.
11. *Кротов В.Ф.* Основы теории оптимального управления. М.: Высшая школа, 1990.

12. *Гурман В.И.* Принцип расширения в задачах управления. М.: Наука, 1997.
13. *Федоренко Р.П.* Приближенное решение задач оптимального управления. М.: Наука, 1978.
14. *Срочко В.А.* Итерационные методы решения задач оптимального управления. М.: Физматлит, 2000.
15. *Тятюшкин А.И.* Многометодная технология оптимизации управляемых систем. Новосибирск: Наука, 2006.
16. *Hiriart-Urruty J.-B.* Generalized differentiability, duality and optimization for problems dealing with difference of convex functions // Lecture Notes in Economics and Mathematical Systems. 1985. **256**. 37–69.
17. *Матвийчук А.Р., Ушаков В.Н.* О построении разрешающих управлений в задачах управления с фазовыми ограничениями // Известия Росс. акад. наук. Теория и системы управления. 2006. № 1. 5–20.
18. *Брайсон А., Хо Ю-Ши.* Прикладная теория оптимального управления. М.: Мир, 1972.
19. *Гантмахер Ф.Р.* Теория матриц. М.: Физматлит, 2004.
20. *Бахвалов Н.С., Жидков Н.П., Кобельков Г.М.* Численные методы. М.: БИНОМ. Лаборатория знаний, 2007.
21. *Vicente L.N., Calamai P.H., Judice J.J.* Generation of disjointly constrained bilinear programming test problems // Comput. Optimizat. Applic. 1992. **1**, № 3. 299–306.
22. *Calamai P.H., Vicente L.N.* Generating quadratic bilevel programming test problems // ACM Transactions on Mathematical Software. 1994. **20**. 103–119.
23. *Стрекаловский А.С.* Элементы невыпуклой оптимизации. Новосибирск: Наука, 2003.
24. *Стрекаловский А.С.* Задачи оптимального управления с терминальными функционалами, представимыми в виде разности двух выпуклых функций // Журн. вычисл. матем. и матем. физики. 2007. **47**, № 11. 1865–1879.
25. *Стрекаловский А.С., Шаранхаева Е.В.* Глобальный поиск в невыпуклой задаче оптимального управления // Журн. вычисл. матем. и матем. физики. 2005. **45**, № 10. 1785–1800.
26. *Strekalovsky A.S., Vasiliev I.L.* On global search for non-convex optimal control problems // Developments in Global Optimization. Nonconvex Optimizat. and Its Applic. Dordrecht: Kluwer Acad. Publ., 1997. 121–133.
27. *Strekalovsky A.S.* On global maximum of a convex terminal functional in optimal control problems // J. of Global Optimization. 1995. № 7. 75–91.
28. *Стрекаловский А.С, Янулевич М.В.* Глобальный поиск в задаче оптимального управления с целевым терминальным функционалом, представленным разностью двух выпуклых функций // Журн. вычисл. матем. и матем. физики. 2008. **48**, № 7. 1187–1201.

Поступила в редакцию
25.06.2010