ЖУРНАЛ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И МАТЕМАТИЧЕСКОЙ ФИЗИКИ

Том 19 Март 1979 Апрель N° 2, с. 367–387

УДК 519.68:517.977

БИБЛИОТЕКА ПРОГРАММ ДЛЯ РЕШЕНИЯ ЗАДАЧ ОПТИМАЛЬНОГО УПРАВЛЕНИЯ

Н.И. ГРАЧЕВ, Ю.Г. ЕВТУШЕНКО

(Москва)

(Пересмотрено 20.02.2003)

Дано описание математической основы библиотеки стандартных программ, предназначенных для численного решения общей задачи оптимального управления со смешанными ограничениями. Приведен пример решения простейшей задачи динамики полета.

В статье дается математическая основа библиотеки стандартных программ, предназначенных для решения задач оптимального управления системами со смешанными ограничениями. В реализованных методах используется редукция исходной задачи к задаче нелинейного программирования. Разработанные программы приспособлены для решения задач широкого класса: задачи на быстродействие, задачи с подвижным правым концом, задачи с недифференцируемыми функционалами, оптимизация по вектору управления и управляющему параметру и т.д. Библиотека является частью создаваемого в Вычислительном центре АН СССР единого пакета прикладных программ для решения задач безусловной минимизации функций многих переменных, нелинейного программирования и оптимального управления.

§1. Основные расчетные формулы

Из обширной литературы, посвященной численным методам решения задач оптимального управления, укажем лишь некоторые монографии [1]–[3] и обзорные статьи [4, 5]. В основе используемых методов лежит редукция исходной задачи к задаче нелинейного программирования. Такой подход развивается рядом авторов (см., например, [2, 4]). Он оказался чрезвычайно эффективным по нескольким причинам: во-первых, с его помощью стали очевидны многие ранее предложенные эвристические алгоритмы, возникла возможность их обобщения; во-вторых, этот подход позволил использовать богатый арсенал методов нелинейного программирования и безусловной минимизации; в-третьих, он создал конструктивную основу для построения методов оптимизации систем, интегрируемых по схемам высокого порядка точности.

Пусть управляемый процесс описывается системой обыкновенных дифференциальных уравнений

$$\frac{dx}{dt} = f(x, u, t), \qquad 0 \le t \le T, \tag{1.1}$$

где t — независимая переменная, x — фазовый вектор, u — вектор управлений. Не будем конкретизировать размерности векторов, так как они несущественны. На управляемый процесс (1.1) наложены ограничения типа равенства и неравенства

$$\Gamma^1(x(t), u(t), t) = 0, \qquad \Gamma^2(x(t), u(t), t) \le 0. \tag{1.2}$$

Заданы также терминальные ограничения

$$\Gamma^3(x(T), T) = 0, \qquad \Gamma^4(x(T), T) \le 0. \tag{1.3}$$

Введем минимизируемый функционал

$$I = \int\limits_0^T F(x(t), u(t), t)dt + F_T(x(T), T). \tag{1.4}$$

Задача оптимального управления формулируется следующим образом.

Задача I. На фиксированном интервале $[0, T]$ среди всевозможных измеримых управлений $u(t)$ таких, что для $u(t)$ и $x(t)$ — соответствующих им решений (1.1) — выполнены условия (1.2) и (1.3), выбрать управление так, чтобы функционал I принимал наименьшее возможное значение.

Различным обобщениям этой задачи специально посвящен § 4. Здесь задача максимально упрощена, чтобы облегчить изложение.

Разобьем отрезок $[0, T]$ на $k-1$ интервалов. Величину i-го интервала обозначим через h_i, положим также

$$t_1 = 0, \qquad t_i = \sum_{s=1}^{i-1} h_s, \qquad x_i = x(t_i), \qquad u_i = u(t_i), \qquad t_k = T,$$
$$z_i = (x_i, u_i, t_i), \qquad f(z_i) = f(x_i, u_i, t_i), \qquad i = 1, 2, \ldots, k.$$

На каждом i-м интервале проинтегрируем (1.1) по схеме, относящейся к семейству методов Рунге–Кутта:

$$x_{i+1} = x_i + h_i \sum_{s=1}^{r} g_s f(z_{is}); \tag{1.5}$$

здесь $z_{is} = (x_{is}, u_{is}, t_{is})$, $u_{i1} = u_i$, $u_{is} = u(t_{is})$,

$$t_{i1} = t_i, \qquad t_{i,j+1} = t_i + \beta_j h_i, \qquad x_{i1} = x_i, \qquad x_{i,j+1} = x_i + \beta_j h_i f(z_{ij}), \tag{1.6}$$

где g_s, β_j — некоторый набор чисел, причем все $0 \le \beta_j \le 1$. В (1.5), (1.6) и всюду далее в этом параграфе (если не оговорено иное) индексы i, s, j принимают целочисленные значения из интервалов $[1, k-1]$, $[1, r]$, $[1, r-1]$ соответственно.

В простейшем случае, когда $r = 1$, (1.5) переходит в схему интегрирования Эйлера

$$x_{i+1} = x_i + h_i f(x_i, u_i, t_i). \tag{1.7}$$

Вычислив квадратуру в (1.4) по формулам (1.5), (1.6), получим

$$I(\bar{x}, w) = F_T(z_k) + \sum_{i=1}^{k-1} h_i \sum_{s=1}^{r} g_s F(z_{is}), \tag{1.8}$$

где $\bar{x} = (x_{11}, \dots, x_{1r}, x_{21}, \dots, x_k)$, $w = (u_{11}, \dots, u_{1r}, u_{21}, \dots, u_k)$.

Задав вектор управлений w, с помощью (1.5) и (1.6) однозначно определим фазовый вектор $\bar{x}$. Учитывая эту связь, будем писать $\bar{x} = x(w)$.

Ограничения (1.2), (1.3) заменим их дискретными аналогами:

$$\bar{\Gamma}^1(\bar{x}, w) = 0, \qquad \bar{\Gamma}^2(\bar{x}, w) \le 0. \tag{1.9}$$

Здесь объединены ограничения типа равенства и неравенства вдоль траектории и терминальные ограничения:

$$\begin{aligned} \bar{\Gamma}^1(\bar{x}, w) &= [\Gamma^1(z_{11}), \Gamma^1(z_{21}), \dots, \Gamma^1(z_{i1}), \dots, \Gamma^1(z_k), \Gamma^3(z_k)], \\ \bar{\Gamma}^2(\bar{x}, w) &= [\Gamma^2(z_{11}), \Gamma^2(z_{21}), \dots, \Gamma^2(z_{i1}), \dots, \Gamma^2(z_k), \Gamma^4(z_k)]. \end{aligned}$$

Исходную **Задачу I** заменим следующей задачей нелинейного программирования.

Задача II. Найти вектор управления w такой, чтобы $\bar{x} = x(w)$ и w удовлетворяли ограничениям (1.9) и функция $I(\bar{x}, w)$ принимала наименьшее возможное значение.

Предполагаем, что **Задачи I** и **II** имеют решение. Вопрос об их близости в случае, когда вместо (1.5) используется аппроксимация по схеме Эйлера, исследовался в ряде работ (см., например, [4, 6]). Эти результаты, по-видимому, переносятся и на более общий случай схем вида (1.5).

Предположим, что функции f, $\bar{\Gamma}^1$, $\bar{\Gamma}^2$, F, F_T дифференцируемы по x и u. В дальнейшем понадобятся формулы для вычисления производных дифференцируемой функции вида

$$B(\bar{x}, w) = \sum_{i=1}^{k-1} h_i \sum_{s=1}^{r} g_s R(z_{is}) + \alpha(z_k). \tag{1.10}$$

Определим векторы

$$p_i = \frac{dB}{dx_i}, \qquad p_{ij} = \frac{dB}{dx_{ij}}, \qquad p_k = \frac{dB}{dx_k}, \qquad p_{ir} = \frac{dB}{dx_{ir}} \tag{1.11}$$

той же размерности, что и x. Учитывая зависимости (1.5), (1.6), определяющие векторы x_{i+1}, $x_{i,j+1}$ как дифференцируемые функции “предыдущих” векторов, получим рекуррентные соотношения

$$p_i = \frac{\partial B}{\partial x_i} + \sum_{s=1}^{r} \frac{\partial x_{is}}{\partial x_i} \frac{dB}{dx_{is}} + \frac{\partial x_{i+1}}{\partial x_i} \frac{dB}{dx_{i+1}} = p_{i+1} + \sum_{s=1}^{r} p_{is}; \tag{1.12}$$

$$p_{ij} = \frac{\partial B}{\partial x_{ij}} + \frac{\partial x_{i,j+1}}{\partial x_{ij}} \frac{dB}{dx_{i,j+1}} + \frac{\partial x_{i+1}}{\partial x_{ij}} \frac{dB}{dx_{i+1}},$$

$$p_{ir} = \frac{\partial B}{\partial x_{ir}} + \frac{\partial x_{i+1}}{\partial x_{ir}} \frac{dB}{dx_{i+1}} = \frac{\partial B}{\partial x_{ir}} + h_i g_r f_x(z_{ir}) p_{i+1}. \tag{1.13}$$

Введем функцию

$$H(z_{is}, p_{i+1}, p_{i,s+1}) = h_i[g_s R(z_{is}) + (f(z_{is}), g_s p_{i+1} + \beta_s p_{i,s+1})].$$

Здесь (a, b) — скалярное произведение векторов a и b. Считая, что $\beta_r = 0$, $p_{i,r+1} = 0$, соотношения (1.13) перепишем компактным образом:

$$p_{is} = H_x(z_{is}, p_{i+1}, p_{i,s+1}). \tag{1.14}$$

Здесь и всюду далее нижние индексы x и u обозначают частные производные по x и u соответственно. Аналогичным образом выводится следующая основная формула для вычисления градиента функции B:

$$\frac{dB(x(w), w)}{du_{is}} = \frac{\partial B}{\partial u_{is}} + \frac{\partial x_{i,s+1}}{\partial u_{is}} \frac{dB}{dx_{i,s+1}} + \frac{\partial x_{i+1}}{\partial u_{is}} \frac{dB}{dx_{i+1}} = H_u(z_{is}, p_{i+1}, p_{i,s+1}). \tag{1.15}$$

Приведенный вывод оказался существенно проще, чем это сделано для системы (1.7) в [2], благодаря введению вспомогательных векторов с помощью соотношений (1.11).

Задав вектор управлений w, можно, используя соотношения (1.5) и (1.6), последовательно определить фазовый вектор $\bar{x}$ и вычислить значение функции B. В случае непрерывной системы это соответствует интегрированию (1.1) "слева направо". Далее с помощью (1.12), (1.14) вычислим последовательность p_i, p_{is}. Эти расчеты соответствуют в оптимальном управлении интегрированию уравнений для импульсов "справа налево". После этого по формуле (1.15) определим производные.

Особо следует подчеркнуть следующее обстоятельство. Использование схем интегрирования вида (1.5) при $r \geq 2$ предполагает, что на каждом интервале интегрирования функция $f(x, u, t)$ имеет ограниченные производные по всем аргументам. Если внутри интервала интегрирования управление $u(t)$ изменяется значительно (на величины ~ 1), то происходит потеря точности расчетов. Поэтому в процессе поиска оптимального управления следует контролировать расположение узлов сетки по t и в случае необходимости либо изменять ее, либо ограничивать величины изменения управления на каждом шаге интегрирования. Проще всего внутри каждого интервала считать управление постоянным. Такой прием оправдан при решении многих практических задач, в которых необходимо интегрировать систему (1.1) с высокой точностью, а вычисление оптимального управления можно проводить грубее. Это связано с тем, что обычно оптимальное управление можно реализовать лишь приближенно и шаг задания управления не должен быть слишком мал.

Если на интервале интегрирования положить управление постоянным, то $u_i = u_{i1} = \ldots = u_{ir}$, вектор w можно считать состоящим из набора $[u_1, \ldots, u_k]$, для производных B по u_i имеет место формула

$$\frac{dB(\bar{x}, w)}{du_i} = \sum_{s=1}^{r} H_u(z_{is}, p_{i+1}, p_{i,s+1}).$$

В полученных формулах возможны существенные упрощения, связанные с тем, что градиенты dB/du нужны лишь для отыскания приближенных оптимальных управлений и точность их расчетов следует согласовывать с точностью решения оптимизационной **Задачи II**. Благодаря этому можно в формулах для определения градиентов и импульсов отбрасывать малые члены.

Рассмотрим некоторые частные случаи схем интегрирования (1.1).

$\boxed{\textbf{1.}}$ В случае метода Эйлера (1.7) имеем

$$\begin{aligned}
B &= \sum_{i=1}^{k-1} h_i R(z_i) + \alpha(z_k), & H(z_i, p_{i+1}) &= h_i[R(z_i) + (f(z_i), p_{i+1})], \\
p_i &= p_{i+1} + H_x(z_i, p_{i+1}), & p_k &= \alpha_x(z_k), \\
\frac{dB}{du_i} &= H_u(z_i, p_{i+1}), & \frac{dB}{du_k} &= \alpha_u(z_k).
\end{aligned} \tag{1.16}$$

Для обоснования сходимости важно уметь вычислять вторые производные функции B. Если предполагать, что функции, определяющие **Задачу I**, дважды дифференцируемы по x и u, то можно получить формулы для вторых производных функций B. Введем квадратные симметрические матрицы вторых производных

$$P_i = \frac{d^2 B(\bar{x}, w)}{dx_i dx_i}, \qquad G_{ij} = \frac{d^2 B(\bar{x}, w)}{du_i du_j}.$$

Несложно получить следующие рекуррентные соотношения:

$$\begin{aligned}
G_{ii} &= H_{uu}(z_i, p_{i+1}) + \Phi_u(z_i) P_{i+1} \Phi_u^\top(z_i), & G_k &= B_{u_k u_k}, \\
P_i &= H_{xx}(z_i, p_{i+1}) + \Phi_x(z_i) P_{i+1} \Phi_x^\top(z_i), & P_k &= B_{x_k x_k}, \\
G_{ij} &= \Phi_u(z_j) D_{j+1}^i [H_{xu}(z_i, p_{i+1}) + \Phi_x(z_i) P_{i+1} \Phi_u^\top(z_i)], \\
D_{j+1}^i &= \frac{\partial x_i}{\partial x_{j+1}} = \Phi_x(z_{j+1}) \Phi_x(z_{j+2}) \ldots \Phi_x(z_{i-1}), & \Phi(z_i) &= x_i + h_i f(z_i);
\end{aligned}$$

здесь $j < i$, D_i^i — единичная матрица, символ "$\top$" над матрицей обозначает транспонирование.

Эти формулы обобщены для случая (1.5), однако мы их не приводим из-за громоздкости. Заметим, что полученные в [7] формулы для вторых производных ошибочны, так как в них нет членов со вторыми производными функции H и только в случае линейных систем они дают верный результат.

$\boxed{\textbf{2.}}$ Для метода Эйлера с пересчетом

$$\begin{aligned}
r &= 2, \quad g_1 = 0, & g_2 &= 1, \quad \beta_1 = 1/2, \\
x_{i+1} &= x_i + h_i f(z_{i2}), & x_{i2} &= x_i + 0.5 h_i f(z_{i1}), \\
p_{i2} &= h_i[R_x(z_{i2}) + f_x(z_{i2}) p_{i+1}], & p_{i1} &= 0.5 h_i f_x(z_{i1}) p_{i2}, \\
\frac{dB}{du_{i1}} &= 0.5 h_i f_u(z_{i1}) p_{i2}, & \frac{dB}{du_{i2}} &= h_i[R_u(z_{i2}) + f_u(z_{i2}) p_{i+1}],
\end{aligned}$$

$$p_i = p_{i+1} + p_{i1} + p_{i2}.$$

По сравнению со схемой Эйлера здесь число точек, в которых ищется вектор управлений, удвоилось. Можно положить всюду $u_i = u_{i1} = u_{i2}$ и учесть, что при использовании схемы Эйлера с пересчетом были отброшены на каждом шаге интегрирования величины порядка h_i^3. С такой же погрешностью можно определять векторы p_i и производные. В результате получим формулы

$$\begin{aligned}
p_i &= p_{i+1} + h_i[f_x(z_{i2}) \bar{p}_{i+1} + R_x(z_{i2})], \\
\bar{p}_{i+1} &= p_{i+1} + 0.5 h_i [f_x(z_{i2}) p_{i+1} + R_x(z_{i2})], \\
\frac{dB}{du_i} &= h_i[f_u(z_{i2}) \bar{p}_{i+1} + R_u(z_{i2})].
\end{aligned}$$

Здесь производные по x и u следует считать с погрешностью порядка h_i^2.

$\boxed{\textbf{3.}}$ Наиболее распространенная схема метода Рунге–Кутта получится, если положить $r = 4$, $g_1 = g_4 = 1/6$, $g_2 = g_3 = 1/3$, $\beta_1 = \beta_2 = 1/2$, $\beta_3 = 1$, $\beta_4 = 0$. Не представляет трудностей в этом случае выписать все необходимые формулы. Поэтому ограничимся лишь случаем, когда на интервале интегрирования управление постоянно. Формулы для пересчета импульсов приведем с погрешностью $O(h_i^3)$, формулы для производных — с погрешностью $O(h_i^2)$:

$$\begin{aligned} p_i = p_{i+1} \quad &+ \quad h_i \sum_{s=1}^{4} g_s[R_x(z_{is}) + f_x(z_{is})p_{i+1}] + \\ &+ \quad h_i^2 \sum_{s=2}^{4} g_s\beta_{s-1} f_x(z_{i,s-1})[R_x(z_{is}) + f_x(z_{is})p_{i+1}], \\ \frac{dB}{du_i} \quad &= \quad h_i \sum_{s=1}^{4} g_s[R_u(z_{is}) + f_u(z_{is})p_{i+1}]. \end{aligned}$$

Аналогичным образом можно получить формулы для других схем интегрирования.

§2. Необходимые условия экстремума

Введем двойственные векторы λ^1, λ^2, имеющие те же размерности, что и, соответственно, $\bar\Gamma^1$, $\bar\Gamma^2$. Составим для **Задачи II** функцию Лагранжа:

$$L(w, \lambda) = I(x(w), w) + (\lambda^1, \bar\Gamma^1(x(w), w)) + (\lambda^2, \bar\Gamma^2(x(w), w)). \tag{2.1}$$

Здесь λ — объединение векторов λ^1 и λ^2.

Используя известные результаты из нелинейного программирования, можно сформулировать для **Задачи II** необходимые и достаточные условия экстремума. Пусть w_* — решение **Задачи II**, $x_* = x(w_*)$, ограничения (1.9) удовлетворяют условию регулярности; тогда согласно теореме Куна–Таккера необходимо существование векторов λ_*^1 и λ_*^2 таких, что

$$\frac{dL(w_*, \lambda_*)}{dw} = 0, \qquad \lambda_*^{2\ell}\bar\Gamma^{2\ell}(x_*, w_*) = 0, \qquad \lambda_*^{2\ell} \geq 0, \tag{2.2}$$

где $\lambda^{2\ell}$, $\bar\Gamma^{2\ell}$ — соответственно ℓ-е координаты векторов λ^2, $\bar\Gamma^2$. Приведенные условия имеют место для всех координат указанных векторов. Производная L по w в (2.2) находится по формулам (1.11) – (1.15), если в них в качестве B взять выражение (2.1). В частности, (1.15) перепишется следующим образом:

$$\begin{aligned} \frac{dL(w_*, \lambda_*)}{du_{is}} \quad &= \quad \frac{\partial}{\partial u_{is}} H(z_{*is}, p_{*i+1}, p_{*i,s+1}, \lambda_{*i}^{1}, \lambda_{*i}^{2}), \\ H(z_{is}, p_{i+1}, p_{i,s+1}, \lambda_{*i}^{1}, \lambda_{*i}^{2}) \quad &= \quad h_i[g_s F(z_{is}) + (f(z_{is}), g_s p_{i+1} + \beta_s p_{i,s+1})] + \\ &+ \quad (\lambda_{*i}^{1}, \bar\Gamma_i^1(z_{i1})) + (\lambda_{*i}^{2}, \bar\Gamma_i^2(z_{i1})). \end{aligned}$$

Здесь $\bar\Gamma_i^1$, $\bar\Gamma_i^2$ — векторы ограничений, вычисленные на i-м шаге, λ_{*i}^{1}, λ_{*i}^{2} — соответствующие им векторы множителей Лагранжа.

При определенных предположениях решение **Задачи II** можно заменить отысканием максимина

$$\max_{\lambda^1} \max_{\lambda^2 \geq 0} \min_{w} L(w, \lambda^1, \lambda^2). \tag{2.3}$$

Аргументация перехода от **Задачи II** к задаче (2.3) та же, что и в нелинейном программировании.

Обозначим через S совокупность всех индексов ℓ, для которых $\bar{\Gamma}^{2\ell}(x(w_*), w_*) = 0$.

Если существуют векторы w_*, λ^1_*, λ^2_* такие, что выполнены условия 2.2), для любого ненулевого вектора w, для которого

$$\frac{d\bar{\Gamma}^1(x(w_*), w_*)}{dw} w = 0, \qquad \left(\frac{d\bar{\Gamma}^{2\ell}(x(w_*), w_*)}{dw},\ w = 0\right), \qquad \ell \in S,$$

справедливо неравенство

$$w^\top \frac{d^2 L(w_*, \lambda_*)}{dw^2} w > 0,$$

тогда w_* является изолированным локальным решением **Задачи II**.

Это достаточное условие может быть использовано при обосновании сходимости приведенных ниже методов решения **Задачи II**. Однако не будем давать достаточных условий сходимости, так как это был бы пересказ в несколько измененных терминах известных в нелинейном программировании фактов.

Заметим, что изложенный подход, берущий за основу технику нелинейного программирования, в отличие от результатов, полученных для непрерывных систем в [8], не приводит к “разрывам” в импульсах. Приведенную схему можно перенести на непрерывные системы, получив для них утверждения, аналогичные принципу максимума. Развивая такой подход, мы придем к необходимым условиям экстремума, полученным ранее иным путем в [9], и при определенных предположениях о выпуклости — к достаточным условиям, приведенным в статье [10].

§3. Описание алгоритмов

Составленная авторами библиотека в настоящее время содержит 16 алгоритмов решения задач оптимального управления. Многие общеизвестные алгоритмы не вошли в библиотеку. В частности, нет алгоритмов, не приспособленных для учета фазовых ограничений, алгоритмов, основанных на вариациях в пространстве состояний, и т.д.

В большинстве приводимых ниже методов решение **Задачи II** ведется по общей схеме: задается вектор w, вычисляются значения I, $\bar{\Gamma}^1$, $\bar{\Gamma}^2$, на основе которых формируется некоторая функция B вида (1.10), и производится ее минимизация по w с помощью любого метода из библиотеки алгоритмов безусловной минимизации (б.м.).

Алгоритмы б.м. можно разделить на 3 группы: 1) не требующие существования производных минимизируемой функции, 2) использующие существование первой производной, 3) использующие существование первой и второй производных. Усиление требований, налагаемых на минимизируемую функцию, позволяет строить более эффективные алгоритмы. Поэтому наиболее высокую скорость сходимости можно получить с помощью алгоритмов группы (3), однако для их реализации приходится отыскивать матрицы вторых производных минимизируемой функции. В задачах оптимального управления, когда размерность вектора w составляет $200 \div 500$, такие расчеты требуют большой машинной памяти. Поэтому при решении практических задач использовались в основном методы группы (2) (методы сопряженных градиентов, наискорейшего спуска и т.д.).

Задача II есть некоторый частный (специальный) случай задачи нелинейного программирования. Важнейшей ее особенностью является сравнительная простота вычисления производных функции B по координатам вектора w. Это дает большие возможности эффективно использовать многие методы нелинейного программирования, основанные на применении первых производных.

Различные методы решения **Задачи II** отличаются друг от друга способами построения функций B и правилами их изменения в процессе итераций. Выбор схемы интегрирования системы (1.1) влияет только на правила вычисления I, $\bar{\Gamma}^1$, $\bar{\Gamma}^2$, производных B по w и не существен с точки зрения описания алгоритмов. Это свойство использовалось при создании библиотеки. Процедуры вычисления I, $\bar{\Gamma}^1$, $\bar{\Gamma}^2$, отыскания производных B были оформлены как отдельные модули. При замене схем интегрирования в алгоритмах библиотеки изменялись только указанные модули, остальные блоки оставались неизменными. Чтобы максимально упростить изложение, будем считать, что интегрирование ведется по схеме Эйлера.

Дадим краткое описание основных алгоритмов.

1. Метод штрафных функций [11]. Составим функцию штрафов

$$B(w,\tau) = I(x(w),w) + \tau[\varphi(\bar{\Gamma}^1(x(w),w)) + \psi(\bar{\Gamma}^2(x(w),w))].$$

Здесь $0 < \tau$ — коэффициент штрафа, функции $\varphi(y)$, $\psi(y)$ в случае скалярного аргумента таковы, что $\varphi(0) = \psi(0) = 0$; $\varphi(y) > 0$ при $y \neq 0$; если $y < 0$, то $\psi(y) = 0$; если $y > 0$, то $\psi(y) > 0$. Если y есть ℓ-мерный вектор с координатами $y^1, \ldots, y^\ell$, тогда положим

$$\varphi(y) = \sum_{i=1}^{\ell} \varphi(y^i), \qquad \psi(y) = \sum_{i=1}^{\ell} \psi(y^i).$$

Метод состоит в следующем: для некоторой монотонно возрастающей последовательности $\tau_1 < \tau_2 < \ldots$ строится последовательность векторов $w(\tau_1), w(\tau_2), \ldots$, определяемых из решения задачи безусловной минимизации B по w:

$$B(w(\tau_s),\tau_s) = \min_w B(w,\tau_s), \qquad s = 1,2,\ldots. \tag{3.1}$$

При определенных условиях последовательность $w(\tau_s)$ при $s \to \infty$ сходится к решению **Задачи II**.

Чтобы воспользоваться методами дифференцируемой минимизации, целесообразно в качестве φ и ψ брать достаточно гладкие функции. Ориентируясь на использование, в частности, метода сопряженных градиентов, мы применяли следующие дважды непрерывно дифференцируемые функции:

$$\varphi(y) = y^2, \qquad \psi(y) = \begin{cases} 0, & y \leq 0, \\ k_1 y^3, & 0 < y \leq R_1, \\ y^2 + k_2 y + k_3, & y > R_1, \end{cases}$$

$k_1 = 1/(3R_1)$, $k_2 = -R_1$, $k_3 = R_1^2/3$, где $0 < R_1$ — некоторое достаточно малое число (обычно $R_1 = 10^{-4}$).

Используя далее формулу (1.15) для вычисления производных функции B, решаем приближенно задачу (3.1). Процесс минимизации прекращается, как только найден вектор w такой, что

$$\|B_w(w,\tau)\| \le \varepsilon(\tau), \tag{3.2}$$

где $\varepsilon(\tau) \to 0$ при $\tau \to \infty$. При численной реализации метода последовательности $\{\tau_s\}$, $\{\varepsilon(\tau_s)\}$ строились следующим образом:

$$\begin{array}{lll} \tau_{s+1} = \nu\tau_s, & \varepsilon(\tau_{s+1}) = b/(1+\ln\tau_{s+1}), & \text{если } \tau_s[\varphi(\bar\Gamma^1)+\psi(\bar\Gamma^2)] > \tilde e, \\ \tau_{s+1} = \tau_s, & \varepsilon(\tau_{s+1}) = \mu\varepsilon(\tau_s), & \text{если } \tau_s[\varphi(\bar\Gamma^1)+\psi(\bar\Gamma^2)] \le \tilde e. \end{array}$$

Здесь числа τ, ν, μ, $\tilde e$, b таковы, что $0 < \tau_1$, $1 < \nu$, $0 < b$, $0 < \mu < 1$, $\tilde e > 0$.

Составлена модификация метода, в которой в качестве φ и ψ были взяты недифференцируемые функции, благодаря чему отпала необходимость стремить τ к бесконечности. Однако при этом пришлось для решения задачи (3.1) использовать менее эффективные методы недифференцируемой оптимизации, что существенно замедлило расчеты. Поэтому переход к недифференцируемым функциям штрафа оправдан только в том случае, если среди функций, определяющих **Задачу I**, есть недифференцируемые. Метод штрафов наиболее полезен, когда начальное приближение известно грубо. Обычно с него начинаются расчеты и определяются начальные значения двойственных переменных. Недостатком метода является низкая скорость сходимости вблизи решения.

2. Двойственный метод [12]. В качестве B возьмем функцию

$$B(w,\lambda^1,\lambda^2) = L(w,\lambda^1,\lambda^2) + \tau[\varphi(\bar\Gamma^1(x(w),w)) + \psi(\bar\Gamma^2(x(w),w))],$$

где L определена в § 2, τ — некоторый фиксированный положительный параметр. Пусть известны λ^1_s, λ^2_s; тогда s-й шаг выполняется по формуле

$$\begin{aligned} &B(w_s,\lambda^1_s,\lambda^2_s) = \min_w B(w,\lambda^1_s,\lambda^2_s), \\ &\lambda^1_{s+1} = \lambda^1_s + c\bar\Gamma^1(x(w_s),w_s), \\ &\lambda^{2\ell}_{s+1} = \lambda^{2\ell}_s + 4c\bar\Gamma^{2\ell}_s \times \begin{cases} e_1, & \text{если } \lambda^{2\ell}_s < e_1,\ \bar\Gamma^{2\ell}_s > 0, \\ \lambda^{2\ell}_s & \text{в противном случае.} \end{cases} \end{aligned} \tag{3.3}$$

Здесь $\bar\Gamma^{2\ell}_s = \bar\Gamma^{2\ell}(x(w_s),w_s)$, c — шаг спуска, $0 < e_1$ — некоторое достаточно малое число (обычно $e_1 = 10^{-4}$). Процесс б.м. (3.3) прекращается при выполнении условия, аналогичного (3.2).

3. Метод модифицированных функций Лагранжа. Имеется большое число публикаций по этому методу (см., например, [13]). Обобщенную функцию Лагранжа составим в виде

$$B(w,\lambda^1,\lambda^2) = I(x(w),w) + [\|\lambda^1 + \tau\bar\Gamma^1\|^2 + \|(\lambda^2 + \tau\bar\Gamma^2)_+\|^2]/2\tau;$$

здесь $\|\cdot\|$ — евклидова норма вектора; компоненты вектора a^i_+ выражаются через компоненты вектора a^i по формуле: $a^i_+ = a^i$, если $a^i > 0$; в противном случае $a^i_+ = 0$.

Переход к $(s+1)$-му шагу осуществляется по формулам

$$\begin{aligned} &B(w_s,\lambda^1_s,\lambda^2_s) = \min_w B(w,\lambda^1_s,\lambda^2_s), \\ &\lambda^1_{s+1} = \lambda^1_s + \tau\bar\Gamma^1(x(w_s),w_s), \qquad \lambda^2_{s+1} = (\lambda^2_s + \tau\bar\Gamma^2(x(w_s),w_s))_+. \end{aligned} \tag{3.4}$$

Параметр τ должен быть достаточно большим, чтобы обеспечить разрешимость задачи (3.4) и сходимость метода.

4. Метод простой итерации [14]. Обобщенная функция Лагранжа строится в виде

$$B(w,\lambda^1,\lambda^2)=I(x(w),w)+[\|\lambda^1+\tau\bar\Gamma^1\|^2+\psi(\tau\bar\Gamma^2)+2(\lambda^2,\gamma(\tau\bar\Gamma^2))]/2\tau;$$

здесь $\psi(y)$ — некоторая достаточно гладкая функция штрафа, например $\psi(y)= = y_+^4$. Вектор-функция $\gamma(a)$ такова, что ее ℓ-я координата γ^ℓ выражается через ℓ-ю координату a^ℓ вектора a по правилу: если $a^\ell\le 0$, то $\gamma^\ell=(1-a^\ell)^{-q}$; в противном случае

$$\gamma^\ell=1+qa^\ell+\frac{q(q+1)}{2}(a^\ell)^2+\frac{q(q+1)(q+2)}{3!}(a^\ell)^3;$$

здесь q — произвольное положительное число (обычно $q=1$).

Метод состоит в следующем:

$$B(w_s,\lambda_s^1,\lambda_s^2)=\min_w B(w,\lambda_s^1,\lambda_s^2),\quad \lambda_{s+1}^1=\lambda_s^1+\tau\bar\Gamma^1(x(w_s),w_s),\quad \lambda_{s+1}^{2\ell}=\frac{\partial B(w_s,\lambda_s^1,\lambda_s^2)}{\partial\bar\Gamma^{2\ell}}.$$

Для ряда тестовых задач этот метод обеспечил наиболее высокую скорость сходимости. Его недостатком, как и двух предыдущих методов, является необходимость знания двойственных переменных. Поэтому для их определения использовался какой-либо из методов **1**, **5** или **6**.

5. Метод Моррисона [15]. Составим функцию

$$B(w,\eta)=(I(x(w),w)-\eta)^2+\varphi(\bar\Gamma^1)+\psi(\bar\Gamma^2),$$

где φ, ψ строятся так же, как и в методе штрафов, η — некоторый параметр. Предполагается, что в начале расчетов известна некоторая нижняя оценка для решения **Задачи II**, а именно: если $I_*=I(x(w_*),w_*)$, то можно указать $\eta_0<I_*$. На $(s+1)$-м шаге определяем η_{s+1} по формулам

$$B(w_s,\eta_s)=\min_w B(w,\eta_s),\qquad \eta_{s+1}=\eta_s+[B(w_s,\eta_s)]^{1/2}. \tag{3.5}$$

6. Модифицированный метод Моррисона [14]. Отличие от метода Моррисона состоит в способе изменения параметра η:

$$\eta_{s+1}=\eta_s+\frac{B(w_s,\eta_s)}{(I(x(w_s),w_s)-\eta_s)}. \tag{3.6}$$

Метод обоснован только в случае задач выпуклого программирования. Несложно показать, что при точном решении задачи б.м. из условия $\eta_0\le I_*$ следует, что $\eta_s\le I_*$ для всех s. Значение η_{s+1}, даваемое (3.5), меньше или равно η_{s+1}, получаемого из (3.6), что обеспечивает в последнем случае более высокую скорость сходимости. Существенным недостатком обоих методов является требование, чтобы безусловная минимизация производилась с высокой точностью. Только в этом случае в процессе расчетов будет выполнено необходимое условие функционирования методов: $\eta_s\le$

$\leq I_*$. В последнем методе, из-за того что увеличение η происходит более быстро, это требование особенно важно.

Оба метода близки к методу штрафов: решив на s-м шаге задачу б.м. и найдя некоторую точку w_s, можно определить такое значение τ, при котором решение задачи б.м. для метода штрафов даст точно такой же результат. Удобство этих методов состоит, таким образом, в том, что изменение η происходит "автоматически", в то время как в методе штрафов приходится специально задавать последовательность τ_s.

7. Метод линеаризации. Для решения задач нелинейного программирования метод предложен в [16], обоснован в [17], для оптимального управления предлагался в [18].

На s-м шаге, линеаризуя минимизируемую функцию и ограничения, получаем

$$I(w_s + \delta w) = I(w_s) + (I_w(w_s), \delta w) + O(\|\delta w\|^2),$$
$$\bar{\Gamma}^1(w_s + \delta w) = \bar{\Gamma}^1(w_s) + \bar{\Gamma}^1_w(w_s)\delta w + O(\|\delta w\|^2),$$
$$\bar{\Gamma}^2(w_s + \delta w) = \bar{\Gamma}^2(w_s) + \bar{\Gamma}^2_w(w_s)\delta w + O(\|\delta w\|^2).$$

Здесь для сокращения записи опущена зависимость функций от x.

Поставим следующую задачу квадратичного программирования: найти минимум по δw функции

$$(I_w(w_s), \delta w) + a\|\delta w\|^2 \tag{3.7}$$

при условиях

$$\bar{\Gamma}^1(w_s) + \bar{\Gamma}^1_w(w_s)\delta w = 0, \qquad \bar{\Gamma}^2(w_s) + \bar{\Gamma}^2_w(w_s)\delta w \leq 0, \tag{3.8}$$

где a — некоторый положительный коэффициент. После нахождения оптимального значения δw_* положим

$$w_{s+1} = w_s + c\delta w_*.$$

Существуют разные варианты выбора шага c. Следуя [17], c находим из условия уменьшения на каждом шаге функции штрафа:

$$W = I(w_{s+1}) + N \max[0, |\bar{\Gamma}^1(w_{s+1})|, \bar{\Gamma}^2(w_{s+1})];$$

здесь N — некоторое достаточно большое число; $\bar{\Gamma}^2$ и $|\bar{\Gamma}^1|$ обозначают набор из компонент $\bar{\Gamma}^2$ и модулей компонент $\bar{\Gamma}^1$ соответственно. При решении (3.7), (3.8) совершался переход к двойственной задаче. Если положить $a = 0$, то (3.7), (3.8) переходит в задачу линейного программирования; к (3.8) следует добавить требование, чтобы модули компонент вектора δw были ограничены.

Из всех собранных в библиотеке методов данный наиболее громоздок и требует наибольшей памяти ЭВМ для расчетов. Это связано с необходимостью запоминать градиенты "активных" ограничений. Поэтому целесообразно пользоваться этим методом только тогда, когда количество существенных фазовых ограничений невелико.

8. Метод итераций Крылова–Черноусько [19]. Первоначально он был предложен для решения задач, в которых отсутствуют фазовые и терминальные ограничения. Покажем, как можно приспособить метод для решения **Задачи I**

(в этом случае его следует рассматривать как алгоритм решения задачи б.м.). Обозначим

$$\bar{H}(x,w,p) = \sum_{i=1}^{k-1} H_i(z_i,p_{i+1}), \qquad p_k = \frac{\partial B}{\partial x_k},$$
$$H_i(z_i,p_{i+1}) = h_i[R(z_i) + (f(z_i),p_{i+1})].$$

Задав некоторое управление w_s, из системы (1.7) определим $\bar{x}_s = x(w_s)$ и из (1.16) — вектор $\bar{p}_s = p(w_s)$. Согласно § 1 всякий вектор $\bar{w}$, доставляющий минимум функции $B(x(w),w)$, является стационарной точкой функции $\bar{H}$, т.е.

$$\frac{\partial H(x(\bar{w}),\bar{w},p(\bar{w}))}{\partial w} = 0. \tag{3.9}$$

Заменим задачу минимизации $B(x(w),w)$ задачей отыскания стационарных точек функции $\bar{H}$ (т.е. точек $\bar{w}$, удовлетворяющих (3.9)). Если

$$\frac{\partial H(x(w_s),w_s,p(w_s))}{\partial w} = a,$$

где $\|a\| \neq 0$, то, разрешив уравнения (3.9) относительно явно входящего w_s, получим

$$w_s = \rho(x(w_s),p(w_s),a).$$

Для решения (3.9) применим метод простой итерации, положив

$$w_{s+1} = \rho(x(w_s),p(w_s),0).$$

Если $\|w_{s+1} - w_s\| < e$, где e — точность расчетов, то считаем, что стационарная точка функции $\bar{H}$ найдена. Предполагаем, что она совпадает с точкой минимума B. Далее изменяем функцию B в соответствии с любым приведенным выше алгоритмом, снова поиск минимума B по w заменим отысканием стационарной точки $\bar{H}$, и т.д.

Приведенная схема привлекает простотой расчета: вместо минимизации B по вектору большой размерности w решается большое число задач отыскания стационарных точек функции от вектора небольшой размерности. К сожалению, такое расщепление не имеет места в весьма часто встречающемся случае оптимизации по вектору w и дополнительному вектору управляющих параметров (см. об этом § 4). Кроме того, приведенный метод часто расходится и его можно применять только в том случае, если w_s известно достаточно хорошо. Поэтому вначале ведется минимизация B по w каким-либо методом б.м. и только после этого происходит обращение к описанной процедуре. Чтобы в некоторой степени ослабить расходимость метода, бралось $w_{s+1} = w_s$ и изменение w_{s+1} велось "справа налево" до тех пор, пока норма $\|w_{s+1} - w_s\|$ оставалась достаточно малой. После этого процесс изменения w прекращался, вычислялись новые значения x, p, снова "справа налево" велось исправление w_{s+1}.

9. Редукция к краевой задаче [1]–[5]. Пусть на s-м шаге итераций известны w_s, $x(w_s)$, $p(w_s)$. Обозначим через ξ вектор, совпадающий с вектором $p_1(w_s)$. Проинтегрируем (1.7), (1.16) "слева направо". Последовательность p_i при этом определяется из неявной схемы. Управление на каждом шаге интегрирования находим из условия

(3.9). На последнем k-м шаге интегрирования получаем некоторые векторы $\bar{x}_{ks}$, $\bar{p}_{ks}$. Если

$$\bar{p}_{ks} = \frac{\partial B(\bar{x}_{ks}, u_{ks})}{\partial x_k}, \tag{3.10}$$

то вспомогательная задача нахождения стационарной точки функции $\bar{H}$ по w решена. Предполагаем, что эти точки совпадают с точкой минимума B по w. В соответствии с любым из алгоритмов **1**–**7** изменяем B и переходим к новому $(s+1)$-му шагу. В противном случае, если (3.10) не имеет места, отыскиваем вектор ξ так, чтобы выполнить это условие. Обычно для отыскания ξ используется метод Ньютона. При реализации метода возникает ряд численных трудностей, связанных с расходимостью (более подробно об этом см. [5]).

В [20] показано, что при определенных предположениях в **Задаче II** имеет место необходимое условие экстремума, аналогичное принципу максимума Понтрягина:

$$H_i(x_i^*, u_i^*, p_{i+1}^*, \lambda_*^1, \lambda_*^2) = \min_{u_i} H_i(x_i^*, u_i, p_{i+1}^*, \lambda_*^1, \lambda_*^2); \tag{3.11}$$

здесь символ $*$ означает, что соответствующий вектор вычислен для оптимального решения **Задачи II**. Функция H_i определена в § 2.

Условие (3.11) позволяет заменить отыскание стационарных точек функции H_i нахождением минимума H_i по u_i. Таким образом, отыскание минимума функции B расщепляется на решение последовательности задач минимизации H_i по вектору низкой размерности u_i. Такой прием, однако, приводит к успеху лишь вблизи решения, поэтому он используется лишь в конце расчетов.

В заключение параграфа заметим, что многочисленные другие методы нелинейного программирования могут быть перенесены аналогичным образом на решение **Задачи I**. Здесь приведены лишь те методы, которые вошли в библиотеку и были апробированы авторами на решении ряда практических задач.

§4. Некоторые обобщения

Вошедшие в библиотеку алгоритмы предназначены для решения задач более общего вида, чем это было описано в § 1. Укажем основные обобщения.

$\boxed{\mathbf{1.}}$ Вместо (1.1) рассматриваются системы, содержащие вектор управляющих параметров ξ:

$$\dot{x} = f(x, u, t, \xi) = f(z, \xi). \tag{4.1}$$

Функции F, F_T, $\bar{\Gamma}^1$, $\bar{\Gamma}^2$ могут также зависеть от ξ. В **Задаче I** следует подобрать управление $u(t)$ и вектор ξ так, чтобы удовлетворить (1.2), (1.3) и минимизировать I. Градиент функции B по ξ вычисляется следующим образом:

$$\frac{dB}{d\xi} = \frac{\partial \alpha(z_k, \xi)}{\partial \xi} + \sum_{i=1}^{k-1} \sum_{s=1}^{r} H_\xi(z_{is}, \xi, p_{i+1}, p_{i,s+1}). \tag{4.2}$$

Объединим вектор управлений и вектор параметров единым символом, положив $\bar{w} = (w, \xi)$. Далее можно использовать все приведенные выше формулы и методы, подставив в них вместо w вектор $\bar{w}$. Процесс оптимизации будет происходить одновременно по w и ξ.

$\boxed{\mathbf{2.}}$ Для системы (1.1) часть координат вектора x может быть не определена при $t = 0$. Отнесем их к координатам вектора ξ. Для вычисления производных воспользуемся формулой (4.2), добавив к ней слагаемое $(\partial x_1/\partial\xi)p_1$. Компоненты вектора x, заданные при $t = T$, следует отнести к терминальным ограничениям (1.3).

Опыт численных расчетов показывает, что часто задачи оказываются чрезвычайно чувствительными к точности решения краевой задачи для системы (1.1). Поэтому целесообразно начинать расчеты с оптимизации по ξ при фиксированном $u(t)$, после этого переходить к одновременной оптимизации по ξ и $u(t)$, в конце проводить дополнительную оптимизацию по ξ при фиксированном $u(t)$. Библиотека методов оптимизации составлялась так, чтобы такие расчеты проходили по одним и тем же алгоритмам без каких-либо преобразований.

$\boxed{\mathbf{3.}}$ Задача об оптимальном быстродействии легко сводится к рассматриваемому классу. Пусть для (1.1) отыскивается управление такое, что $u(t)$ и соответствующая траектория всюду удовлетворяют (1.2) и терминальное множество

$$\Gamma^3(x(t), u(t), t) = 0, \qquad \Gamma^4(x(t), u(t), t) \leq 0 \tag{4.3}$$

достигается при наименьшем возможном значении t. В качестве ξ возьмем $(k-$ $-1)$-мерный вектор, у которого каждая i-я координата совпадает с величиной шага интегрирования h_i на i-м интервале. Тогда минимизируемая функция будет равна сумме

$$I = \sum_{i=1}^{k-1} \xi^i.$$

Фиксируем количество интервалов дискретизации k и отыскиваем ξ, w так, чтобы выполнялись условия (1.9), (4.3) и достигался минимум I. Производную B по ξ следует вычислять по формуле

$$\frac{dB}{d\xi^i} = \sum_{s=1}^{r} \frac{\partial H(z_{is}, \xi^i, p_{i+1}, p_{i,s+1})}{\partial h_i}.$$

Теперь задача имеет стандартный вид и можно использовать любой приведенный выше метод.

Если шаги интегрирования всюду постоянны, $h = h_1 = \ldots = h_{k-1}$, достаточно ввести скаляр ξ, положить $I = \xi(k-1)$, производную B считать по формуле

$$\frac{dB}{d\xi} = \sum_{i=1}^{k-1} \frac{dB}{d\xi^i}.$$

$\boxed{\mathbf{4.}}$ Если вектор-функция $f(x, u, t)$ дифференцируема по x, u всюду, кроме конечного числа точек $\{t_j\}$, где имеет место разрыв I рода, то величины шагов h_i следует подбирать так, чтобы все точки $\{t_j\}$ были в узлах основной сетки по t. Приведенные в § 1 формулы остаются без изменений. Аналогично, если в некоторых точках $\{t_i\}$ из основной сетки по t фазовая траектория претерпевает заданный разрыв, не зависящий от x и u, то формулы § 1 для вычисления производных также не изменяются. Если разрыв зависит от x, u, т.е.

$$x(t_{i+}) = \psi(x(t_{i-}), u(t_i)),$$

где

$$x(t_{i+}) = \lim_{t \to t_{i+0}} x(t), \qquad x(t_{i-}) = \lim_{t \to t_{i-0}} x(t),$$

то в этих точках следует производить пересчет импульсов. При определении p_{i-1} вместо p_i следует брать вектор $(\partial\psi(x(t_{i-}), u(t_i))/\partial x)p_i$ и в формуле для производной dB/du_i добавлять слагаемое $(\partial\psi(x(t_{i-}), u(t_i))/\partial u)p_i$.

$\boxed{\mathbf{5.}}$ Если система (1.1) линейная, то мы приходим к некоторому весьма простому частному случаю **Задачи I**. Все алгоритмы остаются применимыми, их можно упростить, учтя, что производные правых частей (1.1) постоянны.

$\boxed{\mathbf{6.}}$ Во многих задачах существуют ограничения “параллелепипедного” типа:

$$a \leq w \leq b. \tag{4.4}$$

Ограничения такого вида нецелесообразно относить к ограничениям (1.9). Проще учесть их при решении задач безусловной минимизации. Действительно, можно показать, что все приведенные выше формулы остаются в силе, если вместо обычной задачи безусловной минимизации B по w решать задачу о минимизации B по w на множестве (4.4). Поэтому библиотека программ б.м. была составлена в двух вариантах: с учетом ограничений вида (4.4) и без их учета. В методе линеаризации вектор δw ищется при ограничениях (3.7) и при условии $a \leq w_s + \delta w \leq b$.

$\boxed{\mathbf{7.}}$ В ряде задач накладывается дополнительное ограничение на скорость изменения управления:

$$|du(t)/dt| \leq c. \tag{4.5}$$

Если t_i и t_{i+1} — соседние точки сетки интегрирования по t, то в дискретном варианте приведенное ограничение перепишется в виде

$$-ch_i \leq u(t_{i+1}) - u(t_i) \leq ch_i.$$

Полученные два ограничения можно отнести к ограничениям Γ^2. При использовании формул § 1 для производных следует учесть дополнительные слагаемые в связи с тем, что ограничения на i-м шаге зависят не только от u_i, но и от u_{i+1}.

Заметим, что ограничения (4.5) можно ввести искусственно для регуляризации задачи. Аналогично можно учесть условия запаздывания в системе (1.1).

$\boxed{\mathbf{8.}}$ При отыскании оптимальных прочностных конструкций требуется, чтобы управление $u(t)$ на некоторых заданных интервалах $A_j = [t_j, t_j + \tau_j]$ было постоянным. Для учета этого фактора следует задать начальное управление, удовлетворяющее указанному требованию, а в дальнейших расчетах в качестве dB/du_{js}, где $t_{js} \in A_j$, брать сумму производных dB/du_{js}, где суммирование ведется по всем индексам таким, что $t_{js} \in A_j$.

$\boxed{\mathbf{9.}}$ Согласно [5] применение современных методов решения **Задачи I** резко осложняется, если функционал имеет вид

$$I_1 = \max_{0 \leq t \leq T} v(x(t), u(t), t), \qquad I_2 = \int_0^T |v(x(t), u(t), t)| dt.$$

Несмотря на то, что v — дифференцируемая функция своих аргументов, функционалы I_1 и I_2 дифференцируемы лишь по направлению. В первом случае введем дополнительный управляющий параметр ξ, по которому будем проводить минимизацию, и новое ограничение, положив

$$I_1 = \xi, \qquad v(x(t), u(t), t) \leq \xi.$$

Во втором случае используем обычный прием: введем новое управление $\tilde{u}(t)$ и два ограничения, положив

$$I_2 = \int_0^T \tilde{u}(t)dt, \qquad -v(x(t), u(t), t) \leq \tilde{u}(t) \geq v(x(t), u(t), t).$$

Минимизация ведется по $u(t)$ и $\tilde{u}(t)$. Задачи в обоих случаях сведены к виду I. Никаких недифференцируемых функций не возникает, и все методы могут быть применены без каких-либо дополнительных модификаций. Аналогичным образом можно редуцировать простейшие минимаксные задачи [21].

§5. Пример числовых расчетов

Из большого числа разнообразных практических задач, решенных с помощью описанной библиотеки, приведем лишь одну, посвященную оптимизации движения летательного аппарата (л.а.). Уравнения движения описываются следующей системой:

$$\begin{aligned}
&\dot{x} = V\cos\theta\cos\psi, \qquad \dot{y} = V\sin\theta, \qquad \dot{z} = -V\cos\theta\sin\psi,\\
&G\dot{V} = g(k_1 P\cos\alpha - C_x qS - G\sin\theta), \qquad \dot{G} = -C_s,\\
&V\dot{\theta} = g(k_2 N\cos\gamma - \cos\theta), \qquad \dot{\psi}V\cos\theta = -gk_2 N\sin\gamma.
\end{aligned} \tag{5.1}$$

Здесь x, y, z — декартовы координаты л.а., V — модуль вектора скорости, θ — угол наклона траектории, ψ — угол курса, G — вес, α — угол атаки, γ — угол крена, k_1 — величина тяги двигателя, отнесенная к максимальному значению тяги P, k_2 — величина перегрузки, отнесенная к максимальному значению N, k_3 — относительная величина тормозящей силы, S — характерная площадь л.а., C_x — коэффициент лобового сопротивления, $g = 9.81\,\text{м/сек}^2$, q — скоростной напор, C_s — секундный расход топлива.

Использовались следующие зависимости:

$$\begin{aligned}
&q = \rho(y)V^2/2, \quad \rho(y) = 3.3\cdot 10^{-10}y^2 - 1.155\cdot 10^{-5}y + 0.125,\\
&P = [10 + V^2/a^2(y)](2500 - y)/12.5, \qquad a(y) = 340.3 - 4.08\cdot 10^{-3}y,\\
&\alpha = \frac{k_2 NG}{k_1 P + 4.6qS}, \quad C_s = [0.7 + 2(k_1 - 0.3)^2]k_1 P/3600, \quad S = 55\,\text{м}^2,\\
&N = \min\left(\frac{qS}{G}, \frac{150000}{G}, 8\right), C_x = 0.02 + 3.174\alpha^2 + 0.03k_3.
\end{aligned}$$

На управления наложены ограничения:

$$\begin{aligned}
&0.05 \leq k_1 \leq 1, \qquad 0.01 \leq k_2 \leq 1, \qquad 0 \leq k_3 \leq 1,\\
&|\dot{k}_1| \leq 0.2, \qquad |\dot{k}_2| \leq 0.25, \qquad |\dot{k}_3| \leq 1, \qquad |\dot{\gamma}| \leq 1.57\,\text{рад/сек}.
\end{aligned} \tag{5.2}$$

Начальные значения фазовых координат и управлений следующие:

$$\begin{array}{ccccc} x = z = 0, & y = 5000\,\text{м}, & V = 300\,\text{м/с}, & \psi = \theta = 0, \\ G = 20000\,\text{кГ}, & k_1 = 1, & k_2 = 1/N, & k_3 = 0, & \gamma = 0. \end{array} \tag{5.3}$$

Для системы (5.1) ищутся управления $k_1(t)$, $k_2(t)$, $k_3(t)$, $\gamma(t)$ таким образом, чтобы они удовлетворяли условиям (5.2) и переводили л.а. за кратчайшее время из (5.3) в область, задаваемую условиями

$$y = 7000\,\text{м}, \qquad \theta = 0, \qquad \psi = -\pi, \qquad k_2 = 1/N, \qquad \gamma = 0. \tag{5.4}$$

В качестве начального управления было взято следующее:

$$k_1(t) \equiv 1, \qquad k_2(t) \equiv 0.3, \qquad k_3(t) \equiv 0, \qquad \gamma(t) = 1.5\sin(\pi t/T_1), \qquad T_1 = 21\,\text{сек}.$$

Задача решалась пятью методами. Чтобы более четко проследить процесс сходимости, для решения задачи б.м. всегда использовался один и тот же метод сопряженных градиентов, расчеты заканчивались либо при выполнении (3.2), либо когда количество шагов превосходило величину D. Обозначим через T текущее в процессе расчетов значение функционала (величину времени перехода из (5.3) в (5.4)). На фигуре показано, как изменялась величина T в зависимости от времени расчетов t_1 на ЭВМ БЭСМ-6. Точки — значения T в момент окончания каждой s-й итерации, в промежутках проводилась интерполяция. Номера кривых совпадают с номерами соответствующих алгоритмов, приведенных в § 4. Методы 3, 4 требовали знания двойственных переменных. Поэтому они использовались лишь после того, как был сделан один шаг по методу штрафов.

Подчеркнем, что на процесс расчетов в значительной степени влияет удачный выбор параметров методов. Авторы не проводили специальной оптимизации по ним и не пытались получить "рекордные" результаты. Обычно каждым методом делалось 2-3 просчета, а наилучший результат приведен на фигуре. Были взяты следующие параметры.

Для метода 1: $R_1 = 10^{-4}$, $\tau_1 = b = 1$, $\nu = 1.5$, $\mu = 0.9$, $D = 100$, $\tilde{e} = 10^{-4}$.

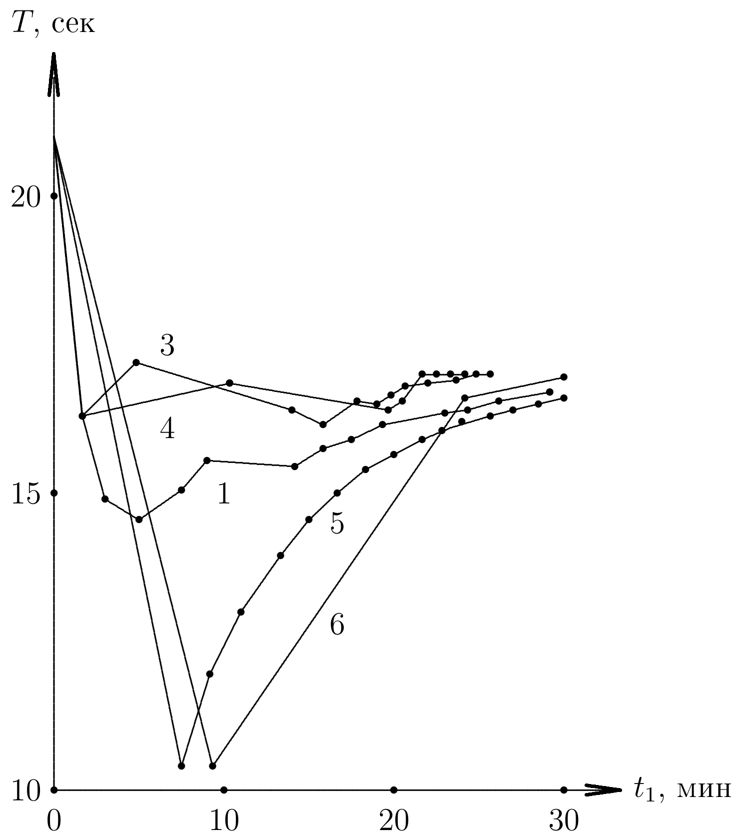
Для метода 3: $\tau = 10$, $\varepsilon(s) = 0.5/(1 + 0.5s)$, $D = 100$.

Для метода 4: $\tau = 10$, $\varepsilon(s) = 0.5/(1 + 0.05s)$, $D = 100$.

Для метода 5: $\eta_0 = 10$, $\varepsilon(s) = 0.1/(1 + s)$, $D = 100$.

Для метода 6: $\eta_0 = 10$, $\varepsilon(s) = 0.01/(1 + s)$, $D = 200$.

Во всех случаях делалась равномерная сетка по независимой переменной с $k = 51$ интегрирование велось по схеме Эйлера, исходная задача сводилась к задаче нелинейного программирования, в которой отыскивался минимум по 205 переменным при наличии 702 ограничений типа неравенства и 5 ограничений типа равенства. Ограничения "параллелепипедного" типа учитывались в методе безусловной минимизации, остальные ограничения были отнесены к ограничениям вида (1.9). Их количество равно 405. Такую же размерность имел вектор двойственных переменных λ. Расчеты по методам прекращались, если два последующих значения T отличались меньше чем на 10^{-4}. Из фигуры видно, что наибольшей скоростью сходимости обладают методы 4, 3. Приемлемая точность была достигнута за 25 мин. Метод штрафов потребовал более часа, чтобы получить близкие результаты. При использовании метода 6 пришлось увеличить точность решения задачи б.м. по сравнению с методом 5, так как в противном случае после второго шага получалась величина T, превосходящая оптимальное значение. Несмотря на значительные усилия, авторам не удалось с помощью метода линеаризации получить результаты, способные конкурировать с другими методами, и поэтому они не были представлены на рисунке.



Фиг.1

Описанная библиотека программ использовалась для решения многих других задач. В [22], в частности, дано решение задачи оптимального проектирования силовых конструкций, решена игровая задача Айзекса о долихобрахистохроне.

ЦИТИРОВАННАЯ ЛИТЕРАТУРА

1. *Н.Н. Моисеев.* Численные методы в теории оптимальных систем. М., "Наука", 1971.

2. *Э. Полак.* Численные методы оптимизации. Единый подход. М., "Мир", 1974.

3. *Ф.П. Васильев.* Лекции по методам решения экстремальных задач. М., Изд-во МГУ, 1974.

4. *Ю.М. Ермольев, В.П. Гуленко.* Конечно-разностный метод в задачах оптимального управления. Кибернетика, 1967, No.3, 1–20.

5. *Р.П. Федоренко.* Приближенное решение современных вариационных задач. В сб. “Математическое программирование и смежные вопросы. Вычислительные методы”. М., ЦЭМИ АН СССР, 1976, 28–97.

6. *Б.М. Будак, Е.М. Беркович, Е.Н. Соловьева.* О сходимости разностных аппроксимаций для задач оптимального управления. Ж. вычисл. матем. и матем. физ., 1969, 9, No.3, 522–547.

7. *I.V. Outrata.* The second-order methods in discrete optimal control problems. Kybernetika, 1976, 12, No.1, 38–45.

8. *А.Я. Дубовицкий, А.А. Милютин.* Задачи на экстремум при наличии ограничений. Ж. вычисл. матем. и матем. физ., 1965, 5, No.3, 395–453.

9. *Э.Р. Смольяков.* Принцип максимума для задач с фазовыми ограничениями. В сб. “Иссл. операций”. Вып. 2. М., ВЦ АН СССР, 1971, 136–155.

10. *O.L. Mangasarian.* Sufficient conditions for the optimal control of nonlinear systems. SIAM J. Control, 1966, 4, No.1, 139–158.

11. *А. Фиакко, Г. Мак-Кормик.* Нелинейное программирование. Методы последовательной безусловной минимизации. М., “Мир”, 1972.

12. *Ю.Г. Евтушенко.* Численные методы решения задач нелинейного программирования. Ж. вычисл. матем. и матем. физ., 1976, 16, No.2, 307–324.

13. *R.T. Rockafellar.* Augmented Lagrange multiplier functions and duality in nonconvex programming. SIAM J. Control, 1974, 12, No.2, 268–285.

14. *А.И. Голиков, Ю.Г. Евтушенко.* Об одном классе методов решения задач нелинейного программирования. Докл. АН СССР, 1978, 239, No.3, 519–522.

15. *D. Morrison.* Optimization by least squares. SIAM J. Numer. Analys., 1968, 5, No.1, 83–88.

16. *R.E. Griffith, R.A. Stewart.* A nonlinear programming technique for the optimization of continuous processing system. Manag. Sci., 1961, 7, No.4, 379–392.

17. *Б.Н. Пшеничный, Ю.М. Данилин.* Численные методы в экстремальных задачах. М., “Наука”, 1975.

18. *Р.П. Федоренко.* Приближенное решение некоторых задач оптимального управления. Ж. вычисл. матем. и матем. физ., 1964, 4, No.6, 1045–1064.

19. *И.А. Крылов, Ф.Л. Черноусько.* Алгоритм метода последовательных приближений для задач оптимального управления. Ж. вычисл. матем. и матем. физ., 1972, No.1, 14–34.

20. *Е.Г. Давыдов.* О распределении ресурсов на графах. В сб. “Системы распределения ресурсов на графах”. М., ВЦ АН СССР, 1970, 31–48.

21. *T.L. Magnanti.* Non-linear programming and the maximum principle for discrete time optimal control problems. Rev. franc. Automat. inform. Rech. oper., 1975, No.3, 75–91.

22. *Н.И. Грачев, Ю.Г. Евтушенко.* Пакет программ для решения задач оптимального управления. М., ВЦ АН СССР, 1978.