MATHEMATICAL CONTROL THEORY
BANACH CENTER PUBLICATIONS, VOLUME 14
PWN – POLISH SCIENTIFIC PUBLISHERS
WARSAW 1985

НЕКОТОРЫЕ ЗАДАЧИ КОНСТРУКТИВНОЙ ТЕОРИИ ОПТИМАЛЬНОГО УПРАВЛЕНИЯ(1)

Р. ГАБАСОВ

Факультет прикладной математики, Белорусский государственный университет им. В. И. Ленина,
Минск, БССР, СССР

I. ДИНАМИЧЕСКАЯ МОДЕЛЬ

Первая часть работы посвящена построению оптимального управления [2], [3], [9] в линейной задаче

$$
\begin{aligned}
&c'x(t_1)\to\max,\\
(1)\qquad &\dot{x}(t) = \tilde{A}x(t)+\tilde{b}u(t), \quad x(0) = x_0, \quad Hx(t_1) = g,\\
&f_*(t)\leqslant u(t)\leqslant f^*(t), \quad t\in[0, t_1];
\end{aligned}
$$

где $\tilde{A}$ — $n\times n$-матрица, $\tilde{b}$ — n-вектор, H — $m\times n$-матрица, $\operatorname{rank} H = m$.

Тщательное исследование линейных задач важно, с одной стороны, потому, что каждый эффективный результат линейной теории можно использовать для нелинейной теории, а с другой — потому, что каждый эффективный алгоритм для нелинейных задач должен быть, как правило, эффективным и для линейных задач.

К настоящему времени качественная теория оптимального управления достигла высокого уровня развития. Для многих задач, среди которых задача (1) является простейшей, получены глубокие результаты по управляемости, наблюдаемости, существованию оптимальных управлений, необходимым условиям оптимальности и многим другим проблемам. Среди них особое положение занимает принцип максимума Понтрягина, ибо традиционно, ещё с исследований по элементарным

(1) Все результаты получены Ф. М. Кирилловой, О. И. Костюковой и автором.

задачам на максимум и минимум, подобные результаты предназначались для фактического решения задач и, в частности, для построения численных методов. История развития неклассических задач оптимизации показывает, что по мере усложнения задач становится все труднее и труднее использовать качественные результаты для конструктивного решения исходных задач. Так из-за чрезвычайной неэффективности процедуры проверки традиционных условий оптимальности в задачах линейного программирования очень эффективный конструктивный метод (симплекс-метод) был создан, исходя из других принципов. И в теории оптимального управления самый сильный результат (принцип максимума) позволил полностью решить лишь двумерные линейные задачи быстродействия без фазовых ограничений. До сих пор нет [10] робастного (по английски „robust“) алгоритма, построенного для нетривиального класса экстремальных задач, который устойчиво решал бы задачу

$$\dddot{x} = u, \quad x(0) = \dot{x}(0) = \ddot{x}(0) = 0, \quad x(t_1) = x_1, \quad \dot{x}(t_1) = x_2, \quad \ddot{x}(t_1) = x_3,$$

$$|u(t)| \leqslant 1, \quad t \in [0, t_1], \quad t_1 \to \min$$

для разных чисел x_1, x_2, x_3.

Достижимая современной математикой степень качественного анализа экстремальных задач явно недостаточна для построения эффективных алгоритмов. Это стало особенно заметным при появлении задач управления с фазовыми ограничениями и очевидно в теории стохастических задач и дифференциальных игр.

В связи со стохастическими задачами, адаптивным управлением и дифференциальными играми возросли требования к качеству алгоритмов решения экстремальных задач. Кроме робастности от них стали требовать высокое быстродействие с тем, чтобы алгоритмы можно было использовать в режиме реального времени. Сейчас в этой области надежды можно возлагать только на алгоритмы построения программ в реальном времени, способных решать нетривиальные классы задач стохастического управления и дифференциальных игр. Поиск управлений типа обратной связи с помощью динамического программирования [1], [4] и близких к нему идей почти невозможен из-за чрезмерных подлежащих хранению объемов информации, обеспечивающей процесс решения. Решенные на этом пути задачи носят пока характер изолированных (или модельных) примеров.

В свете сказанного становится очевидной актуальность построения эффективных алгоритмов решения задачи (1). В первой части работы будет приведен один алгоритм как реализация подхода к решению экстремальных задач, который развивается в Минске и изложен в [5], [6], [8]. Алгоритм будет описан для задач оптимизации обыкновенных систем (1), хотя можно построить его обобщения на системы с последействием и другие линейные системы.

1. Постановка задачи

Пусть отрезок времени $[0, t_1]$ квантован с периодом $h = t_1/N > 0$. Обозначим: $T = \{0, h, \dots, t_1 - h\}$, $T_1 = T \cup t_1$. Функция $u(t)$, $t \in [0, t_1]$, называется *импульсной*, если

$$u(t) \equiv u(\tau h), \quad t \in [\tau h, (\tau+1)h), \quad \tau = 0, 1, \dots, N-1.$$

Считая функции $u(t)$, $f_*(t)$, $f^*(t)$, $t \in [0, t_1]$, импульсными, запишем задачу (1) в эквивалентном виде

$$\begin{gathered} c'x(t_1) \to \max, \quad x(t+h) = x(t) + h[Ax(t) + bu(t)], \quad x(0) = x_0, \\ Hx(t_1) = g, \quad f_*(t) \leqslant u(t) \leqslant f^*(t), \quad t \in T, \end{gathered} \tag{2}$$

где $n \times n$-матрица A и n-вектор b определенным образом выражаются через $\tilde{A}$, $\tilde{b}$.

Функцию $u(t)$, $t \in T$, назовем *допустимым управлением*, если она вместе с соответствующей ей траекторией $x(t)$, $t \in T_1$, системы (2) удовлетворяет всем ограничениям задачи (2).

Допустимое управление $u^0(t)$, $t \in T$, называется *оптимальным*, если оно доставляет решение задаче (2): $c'x^0(t_1) = \max c'x(t_1)$, где $x^0(t)$, $t \in T_1$, — оптимальная траектория, порожденная управлением $u^0(t)$, $t \in T$, а максимум берется по всем допустимым управлениям.

Допустимое управление $u^\varepsilon(t)$, $t \in T$, назовем *ε-оптимальным* (*субоптимальным*), если соответствующая ему траектория $x^\varepsilon(t)$, $t \in T_1$, удовлетворяет неравенству

$$c'x^0(t_1) - c'x^\varepsilon(t_1) \leqslant \varepsilon.$$

Будем считать, что наряду с математической моделью (2) известно некоторое допустимое управление $u(t)$, $t \in T$. Задача состоит в том, чтобы по этой информации и заданному $\varepsilon \geqslant 0$ построить алгоритм, который за конечное число итераций находит ε-оптимальное управление, причем на итерациях получаются только допустимые управления и целевая функция $c'x(t_1)$ монотонно возрастает. Основное достоинство подобных (конечных прямых релаксационных) алгоритмов: останов на любой итерации позволяет иметь допустимое управление, которое не хуже начального.

2. Опорное управление

Несмотря на предельную простоту, задача (2) содержит условие

$$Hx(t_1) = g, \tag{3}$$

обязательное соблюдение которого составляет основную трудность для большинства алгоритмов. Эта трудность существенно возрастает, если требовать, чтобы равенство (3) выполнялось на каждой итерации. Любой алгоритм, в котором гарантируется свойство (3), содержит некоторый элемент, ответственный за выполнение (3). Расходы, связанные с хранением и преобразованием этого элемента составляют плату за (3) и поэтому неизбежны в любом алгоритме со свойством (3). В предлагаемом далее алгоритме для выполнения (3) используется опора и соответствующая ей опорная матрица.

Пусть $I = \{1, 2, \ldots, m\}$ — множество индексов строк матрицы H. Множество $T_{оп} \subset T$, $|T_{оп}| = m$, назовем *опорой*, если является неособой опорная $m \times m$-матрица $P = P(I, T_{оп}) = \{p(t), t \in T_{оп}\}$, столбцы $p(t)$ которой получены по правилу $p(t) = Hq(t)$ из решения $q(t)$, $t \in T$, определяющего уравнения [3] системы (2)

$$q(t-h) = q(t) + hAq(t), \quad t \in T; \quad q(t_1-h) = b. \tag{4}$$

Таким образом, опора связана с важным внутренним свойством системы управления (2): для существования опоры необходимо и достаточно, чтобы система (2) была относительно управляема [3].

Пару $\{u, T_{оп}\}$, где $u = \{u(t), t \in T\}$ — допустимое управление, $T_{оп}$ — опора, будем называть *опорным управлением*.

3. Принцип ε-максимума

Пусть $\{u, T_{оп}\}$ — начальное опорное управление. В этой паре элемент u — часть исходной информации, элемент $T_{оп}$ выбран произвольно. Эффективность предлагаемого алгоритма зависит от качества как $u(t)$, $t \in T$, так и $T_{оп}$. Качество управления определяется опытом, интуицией, талантом специалистов, занятых задачей (2). Качество $T_{оп}$, как покажут дальнейшие вычисления, связано с определенным планом задачи, двойственной к (2). Ниже будет найдена мера оптимальности $u(t)$, $t \in T$, и $T_{оп}$, из которой можно получить рекомендации по выбору начальной опоры $T_{оп}$.

Работа алгоритма начинается с проверки начального опорного управления $\{u, T_{оп}\}$ на субоптимальность. Для этого по опоре $T_{оп}$ построим m-вектор потенциалов

$$v' = \{c'q(t), \ t \in T_{оп}\}'P^{-1}$$

и найдем решение $\psi(t)$, $t \in T$, сопряженной системы

$$\psi(t-h) = \psi(t)+hA'\psi(t), \quad \psi(t_1-h) = c-H'v. \tag{5}$$

Обозначим через $\mathscr{H}(x, \psi, u) = \psi'(Ax+bu)$ функцию Гамильтона системы (2). Справедлив следующий

Принцип ε-максимума. Если вдоль опорного управления $\{u, T_{оп}\}$ и соответствующих ему решений $x(t)$, $t \in T_1$; $\psi(t)$, $t \in T$, систем (2), (5) выполняется условие ε-максимума

$$\mathscr{H}\big(x(t), \psi(t), u(t)\big) = \max_{f_*(t)\leqslant u\leqslant f^*(t)} \mathscr{H}\big(x(t), \psi(t), u\big) - \varepsilon(t), \quad t \in T, \tag{6}$$

$$\varepsilon(t) \geqslant 0, \quad t \in T, \quad \sum_{t\in T} h\varepsilon(t) \leqslant \varepsilon.$$

то $u(t)$, $t \in T$, — ε-оптимальное управление задачи (2).

Обратно: при любом $\varepsilon \geqslant 0$ для каждого ε-оптимального управления u^ε найдется такая опора $T_{оп}$, что вдоль опорного управления $\{u^\varepsilon, T_{оп}\}$ и соответствующих ему траекторий $x^\varepsilon(t)$, $t \in T_1$; $\psi^\varepsilon(t)$, $t \in T$, будет выполняться условие ε-максимума (6).

Таким образом, если оценка субоптимальности

$$\beta(u, T_{оп}) \equiv \sum_{t\in T} h\varepsilon(t)$$

удовлетворяет неравенству $\beta(u, T_{оп}) \leqslant \varepsilon$, то алгоритм заканчивает работу на начальном управлении, поскольку оно оказалось ε-оптимальным.

4. Алгоритм. Общая схема

Пусть при заданном $\varepsilon \geqslant 0$ начальное опорное управление $\{u, T_{оп}\}$ не удовлетворяет принципу ε-максимума. Задача алгоритма состоит в такой замене $\{u, T_{оп}\} \to \{\bar{u}, \bar{T}_{оп}\}$, называемой *итерацией алгоритма*, что $\bar{u}$ — допустимое управление, $\bar{T}_{оп}$ — опора и $c'\bar{x}(t_1) \geqslant c'x(t_1)$.

Пусть

$$\Delta(t) = -\psi'(t)b, \quad t \in T.$$

В моменты $\{t\colon \Delta(t) = 0\}$ управление $u(t)$, $t \in T$, является особым [7] ибо в эти моменты гамильтониан системы (2) не зависит от u.

Особые точки $\{t\colon \Delta(t) = 0\}$ характерны для задач оптимального управления, в которых, как правило, количество переменных $|T|$ значительно превосходит количество m ограничений: $|T| = N \gg m$. Они в алгоритме играют большую роль. Для обеспечения устойчивости алгоритма введем число $\alpha \geqslant 0$ (параметр итерации) и множество

$T_н = T \setminus T_{оп}$ неопорных моментов разобъем на два подмножества

$$T^{(0)}_{н\mu} = \{t \in T_н \colon |\Delta(t)| \leqslant \alpha\}, \qquad T^{(0)}_{нб} = \{t \in T_н \colon |\Delta(t)| > \alpha\}.$$

Допустимое управление $\bar{u} = \{\bar{u}(t), t \in T\}$ будем искать в виде $\bar{u} = u + \theta l$, где $\theta \geqslant 0$ — шаг (скаляр), $l = \{l(t), t \in T\}$ — направление ($|T|$ — вектор).

Направление l в общем случае строится с помощью процедуры корректировки (см. следующий п. 5) начального направления $l^{(1)}$:

$$l^{(1)}(t) = f_*(t) - u(t), \qquad \text{если} \quad \Delta(t) > \alpha;$$

$$l^{(1)}(t) = f^*(t) - u(t), \qquad \text{если} \quad \Delta(t) < -\alpha, \; t \in T^{(0)}_{нб};$$

$$l^{(1)}(t) = 0, \qquad \text{если} \quad t \in T^{(0)}_{н\mu};$$

$$l^{(1)}(t) = -P^{-1}(t, I)Hr^{(1)}(t_1)/h, \qquad t \in T_{оп},$$

где $r^{(1)}(t)$, $t \in T$, — решение уравнения в вариациях

$$r(t+h) = r(t) + h[Ar(t) + bw(t)], \qquad r(0) = 0, \; t \in T, \tag{7}$$

порожденное возмущением $w(t) = l^{(1)}(t)$, $t \in T_н$; $w(t) = 0$, $t \in T_{оп}$.

При достаточно малом $\alpha \geqslant 0$ вектор $l^{(1)} = \{l^{(1)}(t), t \in T\}$ является направлением возрастания из u. Построим управление $u^{(1)} = u + \theta^{(1)} l^{(1)}$, где $\theta^{(1)}$ — максимальный шаг, допускаемый ограничениями задачи (2), т.е.

$$\theta^{(1)} = \min\{1, \theta(t^{(1)})\}, \qquad \theta(t^{(1)}) = \min \theta(t), \qquad t \in T_{оп}; \tag{8}$$

где

$$\theta(t) = \begin{cases} \big(f_*(t) - u(t)\big)/l^{(1)}(t), & \text{если} \quad l^{(1)}(t) < 0; \\ \big(f^*(t) - u(t)\big)/l^{(1)}(t), & \text{если} \quad l^{(1)}(t) > 0; \\ \infty, & \text{если} \quad l^{(1)}(t) = 0. \end{cases}$$

Можно подсчитать, что

$$\beta(u^{(1)}, T_{оп}) = (1 - \theta^{(1)})\beta(u, T_{оп}) + \tag{9}$$
$$+ \theta^{(1)}\Big[\sum_{t \in T_{оп} \cup T^{(0)}_{н\mu}, \Delta(t) > 0} h\Delta(t)\big(u(t) - f_*(t)\big) + \sum_{t \in T_{оп} \cup T^{(0)}_{н\mu}, \Delta(t) < 0} h\Delta(t)\big(u(t) - f^*(t)\big)\Big].$$

Если $\beta(u^{(1)}, T_{оп}) \leqslant \varepsilon$, то корректировать направление $l^{(1)}$ не нужно, ибо $u^{(1)}$ — ε-оптимальное управление. Решение задачи (2) прекращается.

Из формул (6), (9) можно сделать следующие выводы. Если в задаче (2) допускаются неограниченные импульсы в моменты времени $\tau_1, \tau_2, \ldots, \tau_k \in T$ ($k \geqslant m$) и из этих моментов можно составить хотя бы одну опору, то для любого начального управления и при всех достаточно малых $\alpha_{(\varepsilon)} \geqslant 0$, $\alpha(0) = 0$, управление $u^{(1)}$ будет ε-оптимальным, т.е. задача (2) решается алгоритмом за одну итерацию. Величина $\theta^{(1)}$ косвенно характеризует качество опоры $T_{\text{оп}}$: чем больше $\theta^{(1)}$, тем лучше опора, ибо при достаточно малых $\alpha \geqslant 0$ число $\beta(u^{(1)}, T_{\text{оп}})$ убывает с увеличением $\theta^{(1)}$. Точная характеристика опоры $T_{\text{оп}}$ и управления $u(t)$, $t \in T$, следует из формулы

$$\beta(u, T_{\text{оп}}) = \beta_u + \beta_{\text{оп}},$$

в которой $\beta_u = c'x^0(t_1) - c'x(t_1)$ — мера неоптимальности управления, $\beta_{\text{оп}}$ — мера неоптимальности опоры. Число $\beta_{\text{оп}}$ подсчитывается как разность значений целевой функции задачи, двойственной к (2), подсчитанной на двойственном плане, соответствующем опоре $T_{\text{оп}}$, и на оптимальном двойственном плане (детали см. в части II).

Если опора $T_{\text{оп}}$ оптимальна (соответствующий двойственный план оптимален), то $\beta_{\text{оп}} = 0$, $\beta(u, T_{\text{оп}}) = \beta_u$, т.е. $\beta(u, T_{\text{оп}})$ — точная оценка субоптимальности управления. В этом случае при выполнении некоторых условий невырожденности и всех достаточно малых $\alpha \geqslant 0$ управление $u^{(1)}$ является ε-оптимальным.

Продолжим изложение алгоритма. Рассмотрим случай $\beta(u^{(1)}, T_{\text{оп}}) > \varepsilon$, когда нужно сделать корректировку направления $l^{(1)}$. Возможность корректировки связана с особыми моментами $T^{(0)}_{\text{н}\mu}$. Вариация управления в эти моменты почти не влияет (из-за малости $\Delta(t)$) на значение целевой функции, но может сильно влиять на компоненты $l^{(1)}(t)$, $t \in T_{\text{оп}}$, а значит, и на величину шага $\theta^{(1)}$. Процедура корректировки описана в п. 5. Она ведется с параметром ϱ и состоит из конечного числа операций. Исходные данные для начальной ($s = 1$) корректировки:

$$R = T_{\text{оп}},\ T^{(0)}_{\text{но}},\ T^{(0)}_{\text{н}\mu},\ t^{(0)} = \varnothing,\ \beta(u^{(1)}, R) = \beta(u^{(1)}, T_{\text{оп}});\quad z_0(t) = \Delta(t),\ t \in T_{\text{н}};$$

$$t^{(1)},\ \mu_1 = 1,\quad \text{если}\quad u^{(1)}(t^{(1)}) = f_*(t^{(1)});\quad \mu_1 = -1,\quad \text{если}\quad u^{(1)}(t^{(1)}) = f^*(t^{(1)});$$

$$t^{(1)} \in \{t^{(0)}\},\ u^{(1)},\ l^{(1)},\ \theta^{(1)}.$$

Корректировка направления $l^{(1)}$ заканчивается либо построением ε-оптимального управления (задача (2) решена), либо построением нового опорного управления $\{\bar{u}, \bar{T}_{\text{оп}}\}$, для которого $c'\bar{x}(t_1) \geqslant c'x(t_1)$, $\beta(\bar{u}, \bar{T}_{\text{оп}}) < \beta(u, T_{\text{оп}})$ (начало следующей итерации).

Теорема. *Для любых u, $T_{\text{оп}}$, $\varepsilon \geqslant 0$, $\varrho > 0$ и всех достаточно малых $\alpha = \alpha(\varepsilon, u, T_{\text{оп}})$, $\alpha(0, u, T_{\text{оп}}) = 0$, алгоритм за конечное число итераций строит ε-оптимальное управление задачи (2).*

5. Алгоритм. Процедура корректировки

Пусть для натурального числа s известны: множества R, $T_{н} = T\setminus R$, $T_{нδ}^{(k)}$, $k = \overline{0, s-1}$; $T_{нμ}^{(s-1)} = T_{н}\setminus \bigcup_{k=0}^{s-1} T_{нδ}^{(k)}$; числа[2] $Z_k(t)$, $t \in T_{нδ}^{(k)}$, $k = \overline{0, s-1}$; $t^{(0)}$, $t^{(k)}, \mu_k$, $k = \overline{1, s}$; $\theta^{(s)}$; направление $l^{(s)}$, управление $u^{(s)}$ и оценка субоптимальности $\beta(u^{(s)}, R)$.

Возможны два случая: (1) $t^{(s)} \notin \{t^{(0)}, \ldots, t^{(s-1)}\}$, (2) $t^{(s)} \in \{t^{(0)}, \ldots, t^{(s-1)}\}$. Рассмотрим случай (1). Положим

$$Z_s(t) = \mu_s P^{-1}(t^{(s)}, I)Hq(t), \qquad t \in T_{н}. \tag{10}$$

Здесь $P^{-1}(t^{(s)}, I)$ — $t^{(s)}$-ая строка матрицы $[P(I, R)]^{-1}$. Возможны подслучаи: (1а) $T_{нμ}^{(s-1)} = \emptyset$; (1б) $T_{нμ}^{(s-1)} \neq \emptyset$.

В подслучае (1а) заменяем множество R: $R \to \tilde{R}$. Для этого подсчитаем числа

$$\sigma(t) = \begin{cases} -Z_k(t)/Z_s(t), & \text{если } Z_k(t)Z_s(t) < 0, \\ \infty & \text{в остальных случаях,} \end{cases}$$

$t \in T_{нδ}^{(k)}$, $k = \overline{0, s-1}$. Для каждого индекса k числа $\sigma(t) < \infty$, $t \in T_{нδ}^{(k)}$, упорядочим по возрастанию: $\sigma(t_{1,k}) \leqslant \sigma(t_{2,k}) \leqslant \ldots \leqslant \sigma(t_{p_k,k})$, $k = s-1, s-2, \ldots, 0$. Положив $a(0, s-1) = u^{(s)}(t^{(s)}) + (1-\theta^{(s)})l^{(s)}(t^{(s)}) - f_*(t^{(s)})$, если $\mu_s = 1$; $a(0, s-1) = f^*(t^{(s)}) - u^{(s)}(t^{(s)}) - (1-\theta^{(s)})l^{(s)}(t^{(s)})$, если $\mu_s = -1$, из рекуррентных соотношений

$$\begin{gathered} a(i, k) = a(i-1, k) + |Z_s(t_{i,k})|\big(f^*(t_{i,k}) - f_*(t_{i,k})\big), \\ i = \overline{1, p_k}; \quad k = s-1, s-2, \ldots, 0; \\ a(0, k) = a(p_{k+1}, k+1), \quad k = s-2, \ldots, 0, \end{gathered} \tag{11}$$

найдем числа $a(i, k)$, $i = \overline{1, p_k}$, $k = \overline{0, s-1}$.

По построению, для чисел $a(i, k)$, $i = \overline{1, p_k}$, $k = \overline{0, s-1}$, выполняются неравенства: $a(0, s-1) < 0$, $a(p_0, 0) \geqslant 0$;

$$\begin{aligned} a(0, s-1) &\leqslant a(1, s-1) \leqslant a(2, s-1) \leqslant \ldots \leqslant a(p_{s-1}, s-1) \leqslant \\ &\leqslant a(1, s-2) \leqslant a(2, s-2) \leqslant \ldots \leqslant a(p_{s-2}, s-2) \leqslant \ldots \\ &\ldots \leqslant a(1, 0) \leqslant a(2, 0) \leqslant \ldots \leqslant a(p_0, 0). \end{aligned}$$

[2] При реализации метода на ЭВМ числа $Z_k(t)$, $t \in T_{нδ}^{(k)}$, $k = \overline{0, s-1}$, можно не хранить в памяти машин, восстанавливая их каждый раз по формулам типа (10).

Следовательно, существуют такие индексы ν, s_*, что $a(\nu, s_*) \geqslant 0$, $a(\nu-1, s_*) < 0$ (при $\nu = 1$ полагаем $a(0, s_*) = a(p_{s_*+1}, s_*+1)$. Новое множество $\tilde{R}$ строим в виде $\tilde{R} = (R \setminus t^{(s)}) \cup t_{\nu, s_*}$.

Если $s_* = 0$, то корректировку завершаем построением нового опорного управления $\{\bar{u}, \bar{T}_{оп}\}$, $\bar{u} = u^{(s)}$, $\bar{T}_{оп} = \tilde{R}$, для которого по построению, выполняется неравенство $\beta(\bar{u}, \bar{T}_{оп}) < \beta(u, T_{оп})$. Новому опорному управлению приписываем функцию $\bar{\Delta}(t)$, $t \in T$:

$$
\begin{aligned}
(12) \qquad & \bar{\Delta}(t) = \Delta(t) + \sigma(t_{\nu, s_*}) Z_s(t), \qquad t \in T_н; \\
& \bar{\Delta}(t) = \Delta(t), \qquad t \in R \setminus t^{(s)}; \qquad \bar{\Delta}(t^{(s)}) = \Delta(t^{(s)}) + \mu_s \sigma(t_{\nu, s_*}).
\end{aligned}
$$

При $s_* > 0$ все проведенные ранее шаги корректировки с номерами $k = \overline{s_*+1, s}$ и их результаты исключаем из рассмотрения. Корректировку продолжаем со следующими исходными данными:

$$
\begin{aligned}
& s_*, \quad \tilde{R}, \quad \tilde{T}_н = T \setminus \tilde{R}, \quad \tilde{T}^{(k)}_{нд} = T^{(k)}_{нд}, \quad \tilde{Z}_k(\tilde{T}^{(k)}_{нд}) = Z_k(T^{(k)}_{нд}), \\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad k = \overline{0, s_*-1}; \\
(13) \quad & \tilde{T}^{(s_*-1)}_{н\mu} = \tilde{T}_н \setminus \bigcup_{k=0}^{s_*-1} \tilde{T}^{(k)}_{нд}, \quad \tilde{t}^{(0)} = t^{(0)}, \quad \tilde{t}^{(k)} = t^{(k)}, \quad \tilde{\mu}_k = \mu_k, \\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad k = \overline{1, s_*}; \\
& \tilde{u}^{(s_*)} = u^{(s)}, \quad \tilde{l}^{(s_*)} = l^{(s)}, \quad \theta^{(s_*)} = \theta^{(s)}, \quad \beta(u^{(s_*)}, \tilde{R}) = \beta(u^{(s)}, R).
\end{aligned}
$$

Рассмотрим случай (1б): $T^{(s-1)}_{н\mu} \neq \varnothing$. Попытаемся осуществить еще один шаг корректировки направления $l^{(s)}$, не меняя опору $T_{оп}$. Множество $T^{(s-1)}_{н\mu}$ разобъем на подмножества

$$
(14) \qquad T^{(s)}_{н\mu} = \{t \in T^{(s-1)}_{н\mu} \colon Z_s(t) = 0\}, \qquad T^{(s)}_{нд} = \{t \in T^{(s-1)}_{н\mu} \colon Z_s(t) \neq 0\},
$$

где $Z_s(t)$, $t \in T^{(s-1)}_{н\mu}$, определено согласно (10). Положим

$$
\begin{aligned}
& l^{(s+1)}(t) = (1 - \theta^{(s)}) l^{(s)}(t), \qquad t \in T_н \setminus T^{(s)}_{нд}; \\
(15) \qquad & l^{(s+1)}(t) = f_*(t) - u^{(s)}(t) \qquad \text{при} \qquad Z_s(t) > 0; \\
& l^{(s+1)}(t) = f^*(t) - u^{(s)}(t) \qquad \text{при} \qquad Z_s(t) < 0, \ t \in T^{(s)}_{нд}; \\
& l^{(s+1)}(t) = -P^{-1}(t, I) H r^{(s+1)}(t_1)/h, \qquad t \in R.
\end{aligned}
$$

В зависимости от фиксированного числа $\varrho > 0$ (параметра алгоритма) возможны подслучаи:

(1б$_1$) $\mu_s l^{(s+1)}(t^{(s)}) < 0$;

(1б$_2$) $\mu_s l^{(s+1)}(t^{(s)}) \geqslant 0$, $\xi = \big(f^*(t^{(s)}) - f_*(t^{(s)})\big) \beta(u^{(s)}, R) / |l^{(s+1)}(t^{(s)})| \geqslant$
$\geqslant \min\{\beta(u^{(s)}, R), \varrho\}$;

(1б$_3$) $\mu_s l^{(s+1)}(t^{(s)}) \geqslant 0$, $\xi < \min\{\beta(u^{(s)}, R), \varrho\}$.

В случае (16_1) находим индексы ν, s_* и множество $\tilde{R}$, следуя правилам, описанным для случая (1a), положив в формулах

$$\alpha(0, s-1) = \begin{cases} u^{(s)}(t^{(s)}) + l^{(s+1)}(t^{(s)}) - f_*(t^{(s)}), & \text{если} \quad \mu_s = 1, \\ f^*(t^{(s)}) - u^{(s)}(t^{(s)}) - l^{(s+1)}(t^{(s)}), & \text{если} \quad \mu_s = -1. \end{cases}$$

Если при этом $s_* = 0$, то корректировка завершается построением нового опорного управления $\{\bar{u}, \bar{T}_{\text{оп}}\}$, $\bar{u} = u^{(s)}$, $\bar{T}_{\text{оп}} = \tilde{R}$, для которого $\beta(\bar{u}, \bar{T}_{\text{оп}}) < \beta(u, T_{\text{оп}})$, и новой функции $\bar{\Delta}(t)$, $t \in T$ (12). При $s_* > 0$ продолжаем корректировку с исходных данных (13).

В подслучае (16_2) вектор $l^{(s+1)}$ корректирует направление $l^{(s)}$ и позволяет перейти к новому управлению

$$u^{(s+1)} = u^{(s)} + \theta^{(s+1)} l^{(s+1)}, \tag{16}$$

где

$$\theta^{(s+1)} = \min\{1, \theta(t^{(s+1)})\}; \qquad \theta(t^{(s+1)}) = \min \theta(t), \qquad t \in R;$$

$$\theta(t) = \big(f_*(t) - u^{(s)}(t)\big)/l^{(s+1)}(t) \quad \text{при} \quad l^{(s+1)}(t) < 0;$$

$$\theta(t) = \big(f^*(t) - u^{(s)}(t)\big)/l^{(s+1)}(t) \quad \text{при} \quad l^{(s+1)}(t) > 0;$$

$$\theta(t) = \infty \quad \text{при} \quad l^{(s+1)}(t) = 0.$$

Оценка субоптимальности нового управления равна

$$\beta(u^{(s+1)}, R) = (1 - \theta^{(s+1)})\beta(u^{(s)}, R) + \\ + \theta^{(s+1)}\Big[\sum_{t \in T_{\text{оп}} \cup T^{(s)}_{\text{н}\mu}, \Delta(t) > 0} h\Delta(t)\big(u^{(s)}(t) - f_*(t)\big) + \\ + \sum_{t \in T_{\text{оп}} \cup T^{(s)}_{\text{н}\mu}, \Delta(t) < 0} h\Delta(t)\big(u^{(s)}(t) - f^*(t)\big) + \xi^{(s+1)}\Big].$$

Здесь $\xi^{(s+1)} = \xi^{(s)} + \sum_{t \in T^{(s)}_{\text{н}\delta}, \Delta(t) Z_s(t) < 0} h\big|\Delta(t)\big(f^*(t) - f_*(t)\big)\big|$, $\xi^{(1)} = 0$. При $\beta(u^{(s+1)}, R) \leqslant \varepsilon$ корректировка прекращается на ε-оптимальном управлении $u^\varepsilon = u^{(s+1)}$.

Пусть $\beta(u^{(s+1)}, R) > \varepsilon$. Если $\theta^{(s+1)} < 1$, то полагаем $\mu_{s+1} = 1$ при $u^{(s+1)}(t^{(s+1)}) = f_*(t^{(s+1)})$; $\mu_s = -1$ при $u^{(s+1)}(t^{(s+1)}) = f^*(t^{(s+1)})$ и повторяем описанную процедуру корректировки, заменив s на $s+1$. Если $\theta^{(s+1)} = 1$, то уменьшаем параметр α: $\alpha \to \bar{\alpha}$, $\bar{\alpha} < \alpha$, и начинаем итерацию заново.

Рассмотрим подслучай (16_3). Множество R заменим на множество $\tilde{R} = (R \setminus t^{(s)}) \cup t^*$, где t^* — любой элемент из $T^{(s)}_{\text{н}\delta} \neq \varnothing$, множества $T_{\text{н}}$, $T^{(s)}_{\text{н}\delta}$, $T^{(s)}_{\text{н}\mu}$ — на множества $\tilde{T}_{\text{н}} = T \setminus \tilde{R}$, $\tilde{T}^{(s)}_{\text{н}\delta} = t^{(s)}$, $\tilde{T}^{(s)}_{\text{н}\mu} = T^{(s-1)}_{\text{н}\mu} \setminus t^*$, вектор $Z_s(T^{(s)}_{\text{н}\delta})$ — на вектор $\tilde{Z}_s(\tilde{T}^{(s)}_{\text{н}\delta}) = -\mu_s$. Остальные данные не меняем. Переходим к построению направления $l^{(s+1)}$ по формулам (15).

При этом обязательно реализуется подслучай ($1б_2$). Дальнейшие операции осуществляем согласно подслучаю ($1б_2$).

Замечание. (1) Для конкретного выбора элемента t^* из $T^{(s)}_{\text{в}\delta}$ можно предложить правила, увеличивающие эффективность корректировки.

Перейдем к рассмотрению случая (2): $t^{(s)} \in \{t^{(0)}, \ldots, t^{(s-1)}.\}$ Полагаем $\bar{u} = u^{(s)}$, $\bar{T}_{\text{оп}} = R$ и корректировку заканчиваем опорным управлением $\{\bar{u}, \bar{T}_{\text{оп}}\}$. Из предыдущих построений следует, что $J(\bar{u}) - J(u) \geqslant \varrho$, откуда $\beta(\bar{u}, \bar{T}_{\text{оп}}) \leqslant \beta(u, T_{\text{оп}}) - \varrho$.

Лемма. *При любых исходных данных для каждой корректировки требуется конечное число операций на ЭВМ.*

Замечания. (2) Вместо (10) интересно рассмотреть правило

$$T^{(s)}_{\text{в}\mu} = \{t \in T^{(s-1)}_{\text{в}\mu}\colon |Z_s(t)| \leqslant a_s\}, \qquad T^{(s)}_{\text{в}\delta} = \{t \in T^{(s-1)}_{\text{в}\mu}\colon |Z_s(t)| > a_s\},$$

где $a_s \geqslant 0$ — некоторый параметр s-ой корректировки.

(3) С практической точки зрения при корректировке целесообразно использовать параметр $\gamma \geqslant 0$, считая, что $\theta(t) = 0$, если либо $f_*(t) \leqslant u(t) \leqslant f^*(t) + \gamma$, $l(t) < 0$, либо $f^*(t) - \gamma \leqslant u(t) \leqslant f^*(t)$, $l(t) > 0$. Это позволит избежать малых шагов $\theta^{(s)}$ и „малых" пересчетов управлений (16). Управление будет пересчитываться (заменяться на новое) только в том случае, если направление l является γ-подходящим [6].

6. Модификация опоры

Пусть в задаче (2) функции $f_*(t)$, $f^*(t)$, $u(t)$, $t \in T$, получаются квантованием по времени кусочно-постоянных функций, которые определены на отрезке $[0, t_1]$ и имеют точки разрыва из T.

Каждому опорному моменту $t \in T_{\text{оп}}$ припишем множество $T(t)$ соседних моментов, считая, что $t \in T(t)$, $t \in T_{\text{оп}}$; $T(t) \cap T(\bar{t}) = \varnothing$, если $t \neq \bar{t}$; $f_*(\tau) \equiv f_*(t)$, $f^*(\tau) \equiv f^*(t)$, $u(\tau) \equiv u(t)$, если $\tau \in T(t)$, $t \in T_{\text{оп}}$. Множество $T_{\text{ооп}} = \bigcup\limits_{t \in T_{\text{оп}}} T(t)$ назовем *опорным множеством* (*обобщенной опорой*).

Опишем изменения в алгоритме п. 4, вызываемые мобификацией опоры. Прежде всего изменяется определение множества $T_{\text{н}}$: $T_{\text{н}} = T \setminus T_{\text{ооп}}$. Соответственно меняются множества $T^{(0)}_{\text{в}\delta} \subset T_{\text{н}}$, $T^{(0)}_{\text{в}\mu} \subset T_{\text{н}}$. Изменяется правило подсчета компонент $l^{(1)}(t)$, $t \in T_{\text{ооп}}$:

$$l^{(1)}(\tau) = l^{(1)}(t), \ \tau \in T(t); \ l^{(1)}(t) = -P^{-1}(t, I) H r^{(1)}(t_1)/h\,|T(t)|, \ t \in T_{\text{оп}}.$$

При достаточно малых $|T(t)|$, $t \in T_{\text{оп}}$, и α^0 вдоль направления $l^{(1)}$ целевая функция задачи (2) возрастает. Однако в общем случае направление $l^{(1)}$ не будет допустимым. На нем критерий допустимости — равенство $Hr(t_1)=0$, в котором $r(t)$, $t\in T$ — решение системы (7), соответствующее возмущению $w=l^{(1)}$ — выполняется с погрешностью $h\eta_{(1)}=hH\sum\limits_{t\in T_{\text{оп}}} l^{(1)}(t)\bar g(t)$, где

$$\bar g(t)=\sum_{\tau\in T(t)}\big(q(\tau)-q(t)\big)=\sum_{\tau\in T(t)} hg(\max\{\tau,t\}),$$

$g(\max\{\tau,t\})$ — решение определяющего уравнения (см. п. 2) с начальным условием

$$g(t_1-h)=\operatorname{sign}(\tau-t)\cdot\left[Ak+hA^2\frac{k(k-1)}{2!}+\ldots+h^{k-1}A^k\right]b,$$
$$k=|\tau-t|/h.$$

Если такая точность не устраивает, то строим вектор $l^{(1)1}$, который отличается от $l^{(1)}$ только компонентами, определенными на $T_{\text{оо п}}$:

$$l^{(1)1}(\tau)=l^{(1)1}(t),\quad \tau\in T(t);\quad l^{(1)1}(t)=l^{(1)}(t)+h\varphi^{(1)}(t),\quad t\in T_{\text{оп}};$$
$$l^{(1)1}(t)=l^{(1)}(t),\quad t\in T_{\text{н}}=T\diagdown T_{\text{оoп}};\quad \varphi^{(1)}(t)=-P^{-1}(t,I)\eta_{(1)}/h|T(t)|,$$
$$t\in T_{\text{оп}}.$$

На решении $r(t)$, $t\in T$, системы (7), соответствующей возмущению $w=l^{(1)1}$, равенство $Hr(t_1)=0$ выполняется с погрешностью

$$h^2\eta_{(2)}=h^2H\sum_{t\in T_{\text{оп}}}\varphi^{(1)}(t)\bar g(t).$$

Продолжая этот процесс, можно показать, что на решении $r(t)$, $t\in T$, системы (7), соответствующем возмущению $w=l^{(1)k}$, где

$$l^{(1)k}(\tau)=l^{(1)k}(t),\quad \tau\in T(t);\quad l^{(1)k}(t)=l^{(1)k-1}(t)+h^k\varphi^{(k)}(t),\quad t\in T_{\text{оп}};$$
$$l^{(1)k}(t)=l^{(1)}(t),\quad t\in T_{\text{н}};\quad \varphi^{(k)}(t)=-P^{-1}(t,I)\eta_{(k)}/h|T(t)|,\quad t\in T_{\text{оп}},$$

равенство $Hr(t_1)=0$ выполняется с погрешностью

$$h^{k+1}\eta_{(k+1)}=h^{k+1}\sum_{t\in T_{\text{оп}}}\varphi^{(k)}(t)\bar g(t).$$

Описанная процедура применяется в алгоритме п. 4 к каждому вектору $l^{(s)}$. Замена опоры осуществляется по правилам п. 4. При замене опоры для момента t_{ν,s_*} (или t^*), вводимого в опору, строится множество $T(t_{\nu,s_*})$ соседних моментов так, чтобы

$$t_{\nu,s_*}\in T(t_{\nu,s_*});\quad T(t_{\nu,s_*})\cap T(t)=\varnothing,\quad t\in T_{\text{оп}}\diagdown t^{(s)};$$
$$f_*(\tau)\equiv f_*(t_{\nu,s_*}),\quad f^*(\tau)\equiv f^*(t_{\nu,s_*}),\quad u^{(s)}(\tau)\equiv u^{(s)}(t_{\nu,s_*}),\quad \tau\in T(t_{\nu,s_*}).$$

В случае (16$_3$) изменения возникают и в определении множества $\tilde T^{(s)}_{\text{вб}}$ и вектора $Z_s(\tilde T^{(s)}_{\text{вб}})$: $\tilde T^{(s)}_{\text{вб}}=T(t^{(s)})$; $Z_s(t)=-\mu_s$, $t\in\tilde T^{(s)}_{\text{вб}}$.

7. Обсуждение

С точки зрения общих задач линейного программирования задача (2) при больших N, типичных при сколь-нибудь точном исследовании непрерывных динамических систем, имеет две особенности: (1) большое число основных ограничений специальной структуры; (2) большое число переменных.

Предложенный выше алгоритм начинает работу с произвольного допустимого управления, которое в практических задачах часто известно. В других случаях построение начального допустимого управления представляет самостоятельную задачу, для решения которой можно, следуя линейному программированию, ввести первую фазу [6] алгоритма. На итерациях алгоритма п. 4 допустимые управления преобразуются опять в допустимые. При этом монотонно увеличивается значение целевой функции и осуществляется постоянный контроль за уменьшающейся оценкой субоптимальности. Алгоритм является конечным. Перечисленные свойства должны приниматься во внимание в первую очередь при сравнении алгоритма с другими алгоритмами решения задачи (2).

К задаче (2) можно применить симплекс-метод. Если не учитывать специальную (динамическую) структуру основных ограничений, $x(t+1) = Ax(t)+bu(t)$, то симплекс-метод будет весьма неэффективным. Легко построить модификацию симплекс-метода, которая в силу отмеченной специфики использует только рабочие базисные $m\times m$-матрицы и позволяет „в принципе“ решить задачу (2) для произвольных N. Однако эффективность такой модификации будет также небольшой из-за неизбежного возникновения на симплексных итерациях малых шагов [10].

Алгоритм данной работы направлен прежде всего на борьбу с появлением малых шагов. Для этого, как и в [6], из множества неопорных моментов выделяется подмножество $T^{(0)}_{\text{в}\mu}$, вариации управления на котором мало сказываются на значениях целевой функции и поэтому используются для увеличения шага. В отличие от [6] на множестве $T^{(0)}_{\text{в}\mu}$ не решается вспомогательная задача, а непосредственно корректируется допустимое направление так, чтобы „отвести“ направление от границы, препятствующей изменению опорного сигнала. В п. 6 предлагается дополнительный способ увеличения допустимого шага. Здесь учитывается ещё одна особенность задачи (2), в силу которой векторы условий с индексами из $T(t)$ мало отличаются от вектора условий, соответствующего опорному моменту t. В результате нагрузка на $u(t)$, $t\in T_{\text{оп}}$, по обеспечению допустимости управления распределяется на множество $\{u(t), t\in T_{\text{ооп}}\}$, что уменьшает величину опорных компонент подходящего направления, от которой зависит длина шага.

Замена опоры в предлагаемом алгоритме производится по прин ципам работ [6], [8]. Однако специальная форма корректировки допустимых направлений позволяет доказать конечность алгоритма без каких-либо предположений о невырожденности.

В данной части работы приведены формулировка основных утверждений и описание алгоритма. Их обоснование опирается на [6] и может быть легко восстановлено.

II. СТАТИЧЕСКАЯ МОДЕЛЬ

Во второй части работы излагаются основные принципы подхода [6] к решению задач оптимизации, одна реализация которого для линейной задачи оптимального управления была приведена в первой части. Там был описан алгоритм решения динамической (многоэтапной) задачи. Необходимым условием его эффективности согласно принципу, принятому в первой части работы, является эффективность в приложении к простейшей, статической (одноэтапной), задаче оптимизации. Поэтому в данной части рассматривается общая $m \times n$-задача линейного программирования

$$c'x \to \max, \quad b_* \leqslant Ax \leqslant b^*, \quad d_* \leqslant x \leqslant d^*. \tag{1}$$

Являясь частным случаем задачи оптимального управления, задача (1), в свою очередь, содержит в себе первую задачу.

Каждый эффективный алгоритм решения задачи (1) с матрицей A произвольной структуры можно перенести на задачи, в которых матрица A имеет специальную структуру. Чтобы при этом переходе не потерять (ослабить) эффективность исходного алгоритма, нужно с предельным вниманием учесть все особенности специальной задачи, отраженные в параметрах c, b_*, b^*, A, d_*, d^*. Фундаментальных принципов оптимизации для общих задач известно немного. Многообразие существующих алгоритмов объясняется различными формами реализации фундаментальных принципов на разнообразных специальных задачах.

1. Модель

Хотя с формальной точки зрения модель (1) обобщает известные модели

$$c'x \to \max, \quad Ax \leqslant b, \quad x \geqslant 0; \tag{2}$$

$$c'x \to \max, \quad Ax = b, \quad 0 \leqslant x \leqslant d, \tag{3}$$

но это не главная причина, в силу которой далее рассматривается только задача (1).

Модель (1) возникает естественно и в тех случаях, когда традиционная формулировка задачи имеет вид (2) или (3). Дело в том, что при обычной постановке задач упускается из виду одна компонента априорной информации, учет которой, с одной стороны, повышает эффективность алгоритмов оптимизации, с другой – приводит от задач (2), (3) к задаче (1).

Ситуацию проще всего объяснить на типичной задаче линейного программирования, связанной с планированием производства. Пусть на предприятии имеется m видов сырья (с индексами $I = \{1, 2, \ldots, m\}$), которые используются в n технологических процессах (с индексами $J = \{1, 2, \ldots, n\}$). Известны: запасы b_i каждого i-го вида сырья; расходы a_{ij} каждого i-го вида сырья при единичной интенсивности j-го технологического процесса; прибыль c_j от использования единичной интенсивности j-го технологического процесса. Требуется найти оптимальные интенсивности технологических процессов (оптимальный план производства), при которых прибыль предприятия максимальна.

Если через x_j, $j \in J$, обозначить интенсивность j-го технологического процесса, то легко получить традиционную модель (2) производственной задачи, в которой $b = \{b_i, i \in I\}$, $c = \{c_j, j \in J\}$, $A = \{a_{ij}, i \in I, j \in J\}$. Задачи линейного программирования в форме (2) или (3) при небольших m, n удобно решать со студентами на практических занятиях. Но это – формальные упражнения, преследующие некоторые учебные цели. В действительности же предприятия, как правило работают, в них имеются специалисты, наделенные известным опытом, интуицией, талантом. План производства, существующий на предприятии, отражает как-то эти качества коллектива. Если у специалистов собрать и обработать информацию о их предложениях по оптимальному плану производства, то получим сведения такого сорта. По каждому j-му технологическому процессу будут указаны наиболее вероятная интенсивность x_j, нижний $d_{*j} \geqslant 0$ и верхний $d_j^* < \infty$ пределы интенсивностей, за которые оптимальные интенсивности выйти не могут. Будут также указаны наиболее вероятные $A(i, J)x$ расходы i-го сырья и нижние $b_{*i} \geqslant 0$ и верхние $b_i^* \leqslant b_i$ пределы, за которые расход i-го сырья заведомо не выйдет в оптимальном плане производства.

Таким образом, в реальных ситуациях имеется производственная задача не с моделью (2), а с математической моделью (1) и, главное, кроме этой модели имеется n-вектор x, удовлетворяющий её ограничениям и достаточно близкий к решению задачи (1). Математическая часть решения производственной задачи свелась к организации такой обработки вектора x, чтобы для заданного $\varepsilon \geqslant 0$ с минимальными затратами получить ε-оптимальный план производства.

В силу сказанного, в дальнейшем предполагается, что известны математическая модель (1) и начальный план x, который, по определению, представляет n-вектор, удовлетворяющий всем ограничениям задачи([3]) (1).

Оптимальным называется план x^0, доставляющий решение задаче (1). План x^ε будем называть *ε-оптимальным* (*субоптимальным*), если $c'x^0 - c'x^\varepsilon \leqslant \varepsilon$.

2. Опорный план

В работе рассматривается только прямой точный алгоритм, на итерациях которого, по определению, получаются векторы x, удовлетворяющие всем ограничениям задачи (1). Для обеспечения этого важного во многих прикладных задачах свойства используется опора задачи.

Под *опорой задачи* (1) будем понимать любую совокупность $K_{\text{оп}} = \{I_{\text{оп}}, J_{\text{оп}}\}$, $I_{\text{оп}} \subset I$, $J_{\text{оп}} \subset J$, $|I_{\text{оп}}| = |J_{\text{оп}}|$, для которой является неособой опорная матрица $A_{\text{оп}} = A(I_{\text{оп}}, J_{\text{оп}})$.

Пару $\{x, K_{\text{оп}}\}$ назовем *опорным планом*, если x — план, $K_{\text{оп}}$ — опора.

Опору $K_{\text{оп}}$ включим в совокупность исходной информации. Её качество, а значит и рекомендации по выбору, следуют из дальнейших вычислений.

Пусть $\{x, K_{\text{оп}}\}$ — начальный опорный план задачи (1). Обозначим через $\bar{x} = x + \Delta x$ любой другой план и найдем приращение $c'\bar{x} - c'x$ целевой функции задачи (1). Для этого по опоре $K_{\text{оп}}$ вычислим вектор потенциалов

$$u' = u'(I_{\text{оп}}) = c'_{\text{оп}} A_{\text{оп}}^{-1}, \qquad c_{\text{оп}} = c(I_{\text{оп}}),$$

и вектор оценок

$$\Delta' = \Delta'(J) = u'A(I_{\text{оп}}, J) - c'.$$

Обозначим через

$$w_* = b_* - Ax, \qquad w^* = b^* - Ax$$

нижний и верхний векторы невязок. Пусть

$$f = A\Delta x.$$

Тогда можно получить следующую формулу приращения:

$$c'\Delta x = -\Delta'(J_{\text{н}})\Delta x(J_{\text{н}}) + u'f(I_{\text{оп}}), \tag{4}$$

([3]) Если даже на начальном плане x некоторые ограничения не выполняются, то в большинстве случаев неразумно отказываться от такой информации. Существуют [6] процедуры обработки такой информации.

где $J_{\text{н}} = J \setminus J_{\text{оп}}$, векторы $\Delta x(J_{\text{н}})$, $f(I_{\text{оп}})$ независимы и удовлетворяют неравенствам

$$
(5)\qquad \begin{gathered} d_*(J_{\text{н}}) - x(J_{\text{н}}) \leqslant \Delta x(J_{\text{н}}) \leqslant d^*(J_{\text{н}}) - x(J_{\text{н}}), \\ w_*(I_{\text{оп}}) \leqslant f(I_{\text{оп}}) \leqslant w^*(I_{\text{оп}}). \end{gathered}
$$

При этом

$$
(6)\qquad \begin{gathered} d_*(J_{\text{оп}}) - x(J_{\text{оп}}) \leqslant \Delta x(J_{\text{оп}}) = A_{\text{оп}}^{-1} f(I_{\text{оп}}) - A_{\text{оп}}^{-1} A(I_{\text{оп}}, J_{\text{н}}) \Delta x(J_{\text{н}}) \leqslant \\ \leqslant d^*(J_{\text{оп}}) - x(J_{\text{оп}}), \\ w^*(I_{\text{н}}) \leqslant f(I_{\text{н}}) = A(I_{\text{н}}, J) \Delta x(J) \leqslant w^*(I_{\text{н}}), \qquad I_{\text{н}} = I \setminus I_{\text{оп}}. \end{gathered}
$$

Дальнейший анализ опорного плана $\{x, K_{\text{оп}}\}$ и алгоритм основаны полностью на формуле приращения (4).

3. Критерий субоптимальности

Для начального опорного плана $\{x, K_{\text{оп}}\}$ и любого плана $\bar{x}$ можно найти векторы $\Delta x(J_{\text{н}})$, $f(I_{\text{оп}})$, удовлетворяющие неравенствам (5), (6) так, чтобы $\bar{x} = x + \{\Delta x(J_{\text{оп}}), \Delta x(J_{\text{н}})\}$, где вектор $\Delta x(J_{\text{оп}})$ вычисляется по формуле (6). Отсюда следует, что максимум $\beta = \beta(x, K_{\text{оп}})$ функции (4) при учете ограничений (5), но без учета ограничений (6) на $\Delta x(J_{\text{оп}})$, $f(I_{\text{н}})$, является оценкой субоптимальности опорного плана $\{x, K_{\text{оп}}\}$:

$$
(7)\qquad c'x^0 - c'x \leqslant \beta.
$$

Легко подсчитать, что

$$
(8)\qquad \begin{aligned} \beta = \sum_{\Delta_j > 0, j \in J_{\text{н}}} \Delta_j (x_j - d_{*j}) + \sum_{\Delta_j < 0, j \in J_{\text{н}}} \Delta_j (x_j - d_j^*) + \\ + \sum_{u_i < 0, i \in I_{\text{оп}}} u_i w_{*i} + \sum_{u_i > 0, i \in I_{\text{оп}}} u_i w_i^*. \end{aligned}
$$

Оценка субоптимальности β зависит от плана x и опоры $K_{\text{оп}}$. Чтобы найти точную зависимость, введем задачу, двойственную к (1):

$$
(9)\qquad \begin{gathered} \varphi(\lambda) = b^{*\prime} s - b_*' t - d_*' v + d^{*\prime} w \to \min, \\ A'y - v + w = c, \quad s - t - y = 0, \quad s \geqslant 0,\ t \geqslant 0,\ v \geqslant 0,\ w \geqslant 0, \end{gathered}
$$

где

$$
\lambda = \{y, s, t, v, w\}.
$$

Вектор λ с компонентами

$$
\begin{array}{llll}
 & y(I_{\text{оп}}) = u, & y(I_{\text{н}}) = 0; & \\
s_i = y_i, & t_i = 0, & \text{если} & y_i \geqslant 0; \\
s_i = 0, & t_i = -y_i, & \text{если} & y_i < 0,\ i \in I; \\
v_j = \Delta_j, & w_j = 0, & \text{если} & \Delta_j \geqslant 0; \\
v_j = 0, & w_j = -\Delta_j, & \text{если} & \Delta_j < 0,\ j \in J,
\end{array}
$$

однозначно построенный по опоре $K_{\text{оп}}$, удовлетворяет всем ограничениям задачи (9), т.е. является двойственным планом задачи (1).

Обозначим через λ^0 оптимальный двойственный план. Из (1), (8), (9) следует, что

$$
\beta(x, K_{\text{оп}}) = \beta(x) + \beta(K_{\text{оп}}), \tag{10}
$$

где $\beta(x) = c'x^0 - c'x$ — мера неоптимальности начального плана x, $\beta(K_{\text{оп}}) = \varphi(\lambda) - \varphi(\lambda^0)$ — мера неоптимальности начальной опоры $K_{\text{оп}}$.

Если опора $K_{\text{оп}}$ оптимальна ($\lambda = \lambda^0$), то $\beta(K_{\text{оп}}) = 0$ и $\beta(x, K_{\text{оп}}) = \beta(x) = c'x^0 - c'x$, т.е. оценка (7) является точной.

Критерий субоптимальности. Если оценка субоптимальности опорного плана $\{x, K_{\text{оп}}\}$ удовлетворяет неравенству $\beta(x, K_{\text{оп}}) \leqslant \varepsilon$, то x — ε-оптимальный план задачи (1). При любом $\varepsilon \geqslant 0$ для каждого ε-оптимального плана x^ε существует такая опора $K^\varepsilon_{\text{оп}}$, что

$$
\beta(x^\varepsilon, K^\varepsilon_{\text{оп}}) \leqslant \varepsilon.
$$

Принцип ε-максимума в Части I получен расшифровкой этого критерия.

4. Итерация

Чтобы яснее представить сущность подхода к построению предлагаемого алгоритма, рассмотрим сначала задачу (3), которая лежит в основе симплекс-метода. Для задачи (3) формула приращения (4) упрощается:

$$
c'\Delta x = -\Delta'(J_{\text{н}})\Delta x(J_{\text{н}}) = -\sum_{j \in J_{\text{н}}} \Delta_j \Delta x_j. \tag{11}
$$

Из (11) видно, что при $\Delta_j < 0$ увеличение j-ой компоненты x_j плана x ведет к увеличению значений целевой функции $c'x$. Если множество $\{j \in J_{\text{н}}\colon\ \Delta_j < 0\}$ состоит из нескольких элементов, то возникает вопрос, по какому закону увеличивать соответствующие компоненты плана? В симплекс-методе изменяют только одну компоненту (в разных модификациях (реализациях) различную, но всегда только одну). Это означает, что в симплекс-методе в основу положен принцип покоординатного подъема в пространстве неопорных переменных $x_j, j \in J_н$ (опорные переменные x_j, $j \in J_{оп}$, „расходуются" на удержание ограничения $Ax = b$). Если даже учесть, что в случае базисного плана x ($x_j = 0 \vee d_j, j \in J_н$) движение происходит по вершинам множества планов, среди которых обязательно имеется вершина, соответствующая оптимальному плану (считаем, что задача имеет решение), то и в этом случае принцип покоординатного подъема не имеет сколь-нибудь очевидных преимуществ (кроме простоты) перед остальными возможными принципами увеличения значений целевой функции. Напротив, требование обязательного движения по вершинам можно считать недостатком метода, ибо оно накладывает дополнительные ограничения на движение, которых нет в исходной постановке задачи. Неубедительным является и стандартное предложение изменять неопорные переменные по проекции градиента на множество допустимых направлений в точке x. Дело в том, что градиент задает направление наискорейшего подъема только в эвклидовом пространстве. В другой метрике будут другие направления наискорейшего подъема. Более того, для каждого направления увеличения целевой функции можно подобрать метрику пространства, в которой это направление будет направлением наискорейшего подъема. В постановке задачи (3) не говорится о какой-нибудь метрике.

Естественная, внутренняя, метрика задачи (3) определяется ограничениями задачи. Полное описание этой метрики по трудности не уступает решению всей задачи. Однако, без учета ограничений (6) на опорные переменные x_j, $j \in J_{оп}$, метрика задается предельно простыми неравенствами (5) на неопорные переменные x_j. В этой метрике легко найти направление наискорейшего возрастания в задаче (5). Достоинство этого направления состоит в том, что если при движении вдоль него не нарушаются ограничения на опорные переменные, то оно приведет в оптимальный план задачи (5). В этом смысле указанный принцип выбора направления является предельно оптимистическим. Однако, при том объеме информации, которая заложена в принятую метрику, это единственно разумный, на наш взгляд, принцип. В [6] рассмотрены случаи, когда при введении метрики используется дополнительная инфомация об ограничениях (6) задачи. Тогда получаются другие направления наискорейшего возрастания, но принцип максимального приращения целевой функции при их выборе остается неизменным. Вопрос, в сущности, сводится к тому, чтобы введенная метрика не очень сильно усложняла задачу построения направления. Вводя последовательно ограничения на опорные переменные, можно за m шагов получить при неизменной опоре $K_{оп}$ направление, ведущее в оптимальный план [6]. Выбор конкретного числа шагов зависит от ситуации.

Применение изложенного принципа к задаче (1) приводит в силу (4), (5), но без учета (6), к следующему направлению движения из точки x:

$$l_j = \begin{cases} d_{*j} - x_j, & \text{если} \quad \Delta_j > 0, \\ d_j^* - x_j, & \text{если} \quad \Delta_j < 0, \\ 0, & \text{если} \quad \Delta_j = 0, \quad j \in J_{\text{н}}; \end{cases}$$

$$l(J_{\text{оп}}) = A_{\text{оп}}^{-1} w(I_{\text{оп}}) - A_{\text{оп}}^{-1} A(I_{\text{оп}}, J_{\text{н}}) l(J_{\text{н}});$$

где компоненты вектора $w(I_{\text{оп}})$ равны $w_i = w_{*i}$, если $u_i < 0$; $w_i = w_i^*$, если $u_i > 0$; $w_i = 0$, если $u_i = 0$, $i \in I_{\text{оп}}$.

Вдоль этого направления целевая функция $c'x$ возрастает. Если опора $K_{\text{оп}}$ оптимальна, а соответствующий ей двойственный план λ невырожден [6], то движение вдоль l приведет на оптимальный план, т.е. вектор

$$\bar{x} = x + l$$

окажется решением задачи (1).

В общем случае не известно, оптимальна ли опора, и поэтому при движении вдоль l нужно следить за выполнением ограничений на $\Delta x(J_{\text{оп}})$ и $f(I_{\text{н}})$, которые не были учтены при выборе направления l. Максимально допустимый шаг θ вдоль l равен

$$\theta = \min\{1, \theta_{i0}, \theta_{j0}\},$$

где θ_{i_0} — максимально допустимый шаг по $f(I_{\text{н}})$:

$$\theta_{i_0} = \min \theta_i, \quad i \in I_{\text{н}}; \quad \theta_i = w_{*i}/a_i' l, \quad \text{если} \quad a_i' l < 0;$$

$$\theta_i = w_i^*/a_i' l, \quad \text{если} \quad a_i' l > 0; \quad \theta_i = \infty, \quad \text{если} \quad a_i' l = 0;$$

$$a_i = A(i, J);$$

θ_{j_0} — максимально допустимый шаг по $\Delta x(J_{\text{оп}})$:

$$\theta_{j_0} = \min \theta_j, \quad j \in J_{\text{оп}}; \quad \theta_j = (d_{*j} - x_j)/l_j, \quad \text{если} \quad l_j < 0;$$

$$\theta_j = (d_j^* - x_j)/l_j, \quad \text{если} \quad l_j > 0; \quad \theta_j = \infty, \quad \text{если} \quad l_j = 0;$$

единица — максимально допустимый шаг по $f(I_{\text{оп}})$, $\Delta x(J_{\text{н}})$.

На новом плане

$$\bar{x} = x + \theta l$$

выполняется неравенство

$$c'x^0 - c'\bar{x} \leqslant (1 - \theta)\beta(x, K_{\text{оп}}),$$

т.е. при $(1-\theta)\beta(x, K_{\text{оп}}) \leqslant \varepsilon$ процесс решения задачи (1) заканчивается на ε-оптимальном плане $\bar{x}$. Величина θ характеризует косвенно качество опоры $K_{\text{оп}}$: чем больше θ, тем лучше опора $K_{\text{оп}}$.

Пусть на начальном опорном плане выполняется неравенство

$$(1-\theta)\beta(x, K_{\text{оп}}) > \varepsilon.$$

В этом случае перейдем([4]) к замене опоры $K_{\text{оп}}$. Изложенный выше принцип замены плана x на языке оценки субоптимальности $\beta(x, K_{\text{оп}})$ означает такой выбор направления l, при котором оценка субоптимальности максимально уменьшается. Выбор новой опоры $\bar{K}_{\text{оп}}$ также связан с $\beta(x, K_{\text{оп}})$: вместо $K_{\text{оп}}$ выберем $\bar{K}_{\text{оп}}$ так, чтобы $\beta(\bar{x}, \bar{K}_{\text{оп}}) < \beta(\bar{x}, K_{\text{оп}})$, т.е. общий принцип, положенный в основу итерации $\{x, K_{\text{оп}}\} \to \{\bar{x}, \bar{K}_{\text{оп}}\}$, а значит и алгоритма, можно назвать *принципом уменьшения оценки субоптимальности*. Поскольку справедливо разложение (10), то для этого достаточно двойственный план λ заменить на новый базисный $\bar{\lambda}$, при котором $\varphi(\bar{\lambda}) < \varphi(\lambda)$. Вообще говоря, можно, решая двойственную задачу (9), получить λ^0 и на следующей итерации построить оптимальный план x^0 задачи (1). Однако, процесс решения двойственной задачи занимает определенное время, которое может оказаться больше допустимого и поэтому действительного улучшения плана x не произойдет. При достаточно хорошем начальном плане x субоптимальный план скорее получится не при использовании оптимальных опор, а опор, хоть и неоптимальных, но быстро построенных по предыдущим опорам. Исходя из этого, достаточно в задаче (9) сделать одну или несколько итераций и по ним найти $\bar{K}_{\text{оп}}$. Количество итераций зависит от величины θ и запаса времени, доступного к данному моменту. Если θ мало и ресурсы времени позволяют, то разумно улучшать λ до тех пор, пока происходит заметное убывание двойственной целевой функции. При больших θ и дефиците времени ограничиваются одной итерацией двойственного метода. Итерации двойственного метода можно разделить на итерации с коротким шагом [6] и на итерации с длинным шагом. В первом случае движение в пространстве y происходит до появления первых нулевых компонент коплана [6], после чего заменяется опора $K_{\text{оп}}$. Во втором случае движение продолжается без изменения опоры, если скорость изменения двойственной целевой функции отрицательна. Дополнительные операции в двойственном методе с длинным шагом просты и, как правило, вполне окупаются хорошим качеством новой опоры. В [6] описана ещё одна процедура (блочной) замены опоры, которая

([4]) В части I была введена дополнительная операция корректировки l. Она разработана и для задачи (1), но ради упрощения изложения основных принципов здесь не приводится.

оказалась эффективной для некоторых классов задач. В Части I приведены операции по осуществлению одной итерации двойственного метода с длинным шагом для задачи оптимального управления. Приведем описание итерации двойственного метода с коротким шагом для задачи (1).

Предположим, что $\theta = \theta_{i_0}$, т.е. движению вдоль l препятствует i_0-ое ограничение группы $b_* \leqslant Ax \leqslant b^*$. Скорость изменения двойственной целевой функции в точке λ при изменении одной компоненты u_{i_0} вектора потенциалов равна $|\alpha|$, где $\alpha = a'_{i_0} l$. Максимально допустимый шаг равен

$$\sigma = \min\{\sigma_{i*}, \sigma_{j*}\}, \tag{12}$$

где

$$\sigma_{i*} = \min \sigma_i, \quad i \in I_{\text{оп}}; \qquad \sigma_{j*} = \min \sigma_j, \quad j \in J_{\text{н}};$$

$$\sigma_i = \begin{cases} u_i/z_i, & \text{если} \quad u_i z_i < 0; \\ 0, & \text{если} \quad u_i = 0,\ z_i > 0,\ a'_i \bar{x} \neq b^*_i \quad \text{или} \\ & \qquad\ \ u_i = 0,\ z_i < 0,\ a'_i \bar{x} = b_{*i},\ i \in I_{\text{оп}}; \\ \infty, & \text{в остальных случаях}; \end{cases}$$

$$\sigma_j = \begin{cases} -\Delta_j/z_j, & \text{если} \quad \Delta_j z_j < 0; \\ 0, & \text{если} \quad \Delta_j = 0,\ x_j \neq d_{*j},\ z_j > 0 \quad \text{или} \\ & \qquad\ \ \Delta_j = 0,\ x_j \neq d^*_j,\ z_j < 0,\ j \in J_{\text{н}}; \\ \infty, & \text{в остальных случаях}; \end{cases}$$

$$z'(J_{\text{н}} \cup I_{\text{оп}}) = k e'_{i_0} \{A(I_{\text{н}}, J_{\text{н}}) - A(I_{\text{н}}, J_{\text{оп}}) A_{\text{оп}}^{-1} A(I_{\text{оп}}, J_{\text{н}}) - \\ - A(I_{\text{н}}, J_{\text{оп}}) A_{\text{оп}}^{-1}\};$$

$$k = 1, \quad \text{если} \quad a'_{i0} \bar{x} = b^*_{i0}; \quad k = -1, \quad \text{если} \quad a'_{i_0} \bar{x} = b_{*i_0}.$$

Для случая $\theta = \theta_{j_0}$, когда движению вдоль l препятствует j_0-ое ограничение группы $d_* \leqslant x \leqslant d^*$, скорость уменьшения двойственной функции в точке λ равна $|\alpha|$, $\alpha = l_{j_0}$, если изменять одну компоненту Δ_{j_0} вектора оценок Δ. Максимально допустимый шаг вычисляется по формуле (12), если внести следующие изменения:

$$z'(J_{\text{н}} \cup I_{\text{оп}}) = k e'_{j_0} \{A_{\text{оп}}^{-1} A(I_{\text{оп}}, J_{\text{н}}),\ A_{\text{оп}}^{-1}\};$$

$$k = 1 \quad \text{при} \quad \bar{x}_{j_0} = d_{*j_0}; \quad k = -1 \quad \text{при} \quad \bar{x}_{j_0} = d^*_{j_0}.$$

Эти вычисления позволяют вычислить оценку субоптимальности плана $\bar{x}$ (без построения опоры $\bar{K}_{\text{оп}}$):

$$\bar{\beta} = \bar{\beta}(\bar{x}, \bar{K}_{\text{оп}}) = (1 - \theta)(\beta - |\alpha| \sigma).$$

Если $\beta(\bar{x}, \bar{K}_{оп}) \leqslant \varepsilon$, то процесс решения задачи заканчивается на ε-оптимальном плане $\bar{x}$ без построения новой опоры.

При $\bar{\beta} > \varepsilon$ опору $K_{оп}$ заменяем на новую $\bar{K}_{оп} = \{\bar{I}_{оп}, \bar{J}_{оп}\}$ по следующим правилам:

1. Если $\theta = \theta_{i_0}$, $\sigma = \sigma_{i*}$, то $\bar{I}_{оп} = (I_{оп} \setminus i_*) \cup i_0$, $\bar{J}_{оп} = J_{оп}$.
2. Если $\theta = \theta_{i_0}$, $\sigma = \sigma_{j*}$, то $\bar{I}_{оп} = I_{оп} \cup i_0$, $\bar{J}_{оп} = J_{оп} \cup j_*$.
3. Если $\theta = \theta_{j_0}$, $\sigma = \sigma_{i*}$, то $\bar{I}_{оп} = I_{оп} \setminus i_*$, $\bar{J}_{оп} = J_{оп} \setminus j_0$.
4. Если $\theta = \theta_{j_0}$, $\sigma = \sigma_{j*}$, то $\bar{I}_{оп} = I_{оп}$, $\bar{J}_{оп} = (J_{оп} \setminus j_0) \cup j_*$.

Таким образом, при замене опоры опорная матрица $A_{оп}$ в матрице A может сдвигаться по вертикали (случай 1), по горизонтали (случай 4), увеличиваться (случай 2), уменьшаться (случай 3).

Построением опорного плана $\{\bar{u}, \bar{K}_{оп}\}$ завершается итерация.

Итерацию назовем *регулярной*, если $\theta + \sigma > 0$. Достаточное условие регулярности каждой итерации состоит в невырожденности по Данцигу двойственной задачи.

Теорема. *Для любых $\varepsilon \geqslant 0$ и начальных x, $K_{оп}$ описанный алгоритм за конечное число итераций строит ε-оптимальный план, если в процессе его работы встречается конечное число нерегулярных итераций.*

Замечание. Если на каждой итерации использовать процедуру корректировки направления, описанную в Части I, то алгоритм станет конечным для любой задачи (1).

5. Эксперимент

При установлении эффективности алгоритмов особое значение приобретают вопросы классификации. Описанный выше алгоритм является прямым, т.е. он работает с информацией о планах прямой задачи (1). Если исходная информация касается планов двойственной задачи (9), и алгоритм преобразует эту информацию, то он называется *двойственным.* Часто прямые алгоритмы предпочтительнее двойственных, если при решении конкретной задачи запас времени ограничен, и процесс решения может прерваться в любой момент. Для таких ситуаций прямой алгоритм выдает информацию о прямой задаче, которая не хуже начальной; полезная информация двойственного алгоритма для прямой задачи получается, как правило, только после окончания работы алгоритма.

Изложенный выше алгоритм является точным, т.е. на каждой его итерации ограничения задачи (1) выполнялись точно. Если в процессе итераций ограничения могут не выполняться, то алгоритм называется *приближенным.* Понятно, что ослабление требований к анготритму может упростить его реализацию и улучшить некоторые характеристики. Однако, для задач, в которых важно иметь план в любой момент, точные методы незаменимы, хотя за постоянное соблюдение ими ограничений (1) нужны дополнительные расходы времени, памяти и т.п. Наконец, изложенный алгоритм является конечным, т.е. с точки зрения приближенных методов его скорость сходимости равна бесконечности.

Исходя из приведенных рассуждений, описанный выше алгоритм сравнивается с прямым симплекс-методом, который принадлежит тому же классу и является одним из наиболее эффективных алгоритмов линейного программирования. Думается целесообразным все новые алгоритмы сравнивать не между собой, а только с симплекс-методом и его модификациями. Это существенно упростит работу по оценке эффективности алгоритмов.

При численных экспериментах возможны два подхода: (1) испытывать алгоритм на специальных тестовых примерах, (2) проводить массовый эксперимент в определенных классах задач. Первый подход нам представляется несколько искусственным, ибо в нем специально создаются ситуации, не встречающиеся в задачах, для которых разработан алгоритм. Невозможно построить алгоритм, одинаково хорошо работающий на всех задачах. В наших экспериментах выбран второй подход. Наш выбор определился тем, что симплекс-метод хорошо зарекомендовал себя в массовых задачах, хотя его теоретическая эффективность очень невелика. Дело объясняется тем, что существуют задачи, для которых симплекс-метод плох, но такие задачи очень редко реально встречаются.

Между симплекс-методом и предлагаемым алгоритмом имеется принципиальная разница: первый не использует априорную информацию о планах, во втором — это существенный элемент. Поэтому, задавая достаточно хорошие начальные планы, можно во много раз повысить эффективность второго метода перед первым. Однако, это обстоятельство в наших экспериментах не использовалось. Начальные приближения были фактически случайными, равно как и параметры задач. Основная цель экспериментов состояла в сравнении механизмов преобразования планов, положенных в основу двух алгоритмов.

Эксперименты проводились тремя сотрудниками независимо друг от друга для трех типов задач. А. И. Тятюшкин (Сибирский энергетический институт, Иркутск) рассматривал задачи типа (2), А. А. Сенько (Институт математики АН БССР) — задачи (1), В. С. Глушенков (Белгосуниверситет) — задачи (3). Результаты экспериментов, в ходе которых решены тысячи задач с $m = 5 \div 900$, $n = 7 \div 900$ во всех случаях имеют одинаковый характер. Приведем только некоторые результаты А. И. Тятюшкина [6].

В основном эксперименте решались 20×30-задачи

$$c'x\to\max,\qquad Ax\leqslant b,\qquad d_*\leqslant x\leqslant d^*.$$

Параметры задачи генерировались стандартным датчиком случайных чисел с равномерным распределением на отрезках:

$$-100\leqslant c_j\leqslant 100,\qquad -100\leqslant a_{ij}\leqslant 100,\qquad 0\leqslant b_i\leqslant 100,$$
$$-100\leqslant d_{*j}\leqslant 0,\qquad 0\leqslant d_j^*\leqslant 100.$$

Процесс решения начинался всякий раз с плана $x=0$. При симплексной нормировке начальная опора составлялась из индексов свободных переменных $x_{icb}(Ax+x_{cb}=b)$. В предлагаемом алгоритме начальная опора была пуста. Количество итераций в симплекс-методе: 59, 53, 80, 62, 60, ... Количество итераций в описанном алгоритме: 24, 33, 38, 27, 26, ... В среднем по 43 приведенным в [6] задачам отношение количества итераций равно 2,8. При малом числе итераций ошибки округления мало влияют на точность результата, что важно при работе на ЭВМ. В ходе эксперимента было замечено, что количество элементов опоры в предлагаемом методе редко превышает 10 (в симплекс-методе базисная матрица всегда имела размеры 20×20). Поэтому был поставлен первый вспомогательный эксперимент на время счета. По десяти решенным 20×30-задачам оно оказалось в симплекс-методе в 5 раз больше, чем в описанном алгоритме.

Второй вспомогательный эксперимент проводился с целью изучения влияния на эффективность алгоритма большого числа переменных, что важно с точки зрения задач оптимального управления. В 20×60-задачах отношение числа итераций оказалось равным 4,5, в 20×100-задачах — 5. В следующих трех вспомогательных экспериментах исследовалось влияние на эффективность алгоритма процента ненулевых элементов в матрице A, размеров m, n задачи, величины d^*-d_*.

Эти эксперименты показывают, что в рассмотренных классах задач предлагаемый алгоритм не уступает симплекс-методу. В настоящее время развернулись работы по экспериментальной проверке алгоритма на задачах оптимального управления (Часть I). В перспективе — включение алгоритма в процедуры решения в реальном режиме времени задач стохастического управления и дифференциальных игр. Будут продолжены работы и по включению алгоритма в пакеты программ решения нелинейных задач экономики и оптимального управления.

Литература

[1] Р. Беллман, *Динамическое программирование*, ИЛ, Москва 1960.

[2] Р. Габасов, Ф. М. Кириллова, *Принцип максимума в теории оптимального управления*, Наука и техника, Минск 1974.

[3] —, —, *Качественная теория оптимальных процессов*, Наука, Москва 1971.

[4] —, —, *Основы димамического программирования*, Изд-во БГУ, Минск 1975.

[5] —, —, *Построение субоптимальных планов транспортной задачи*, Изв. АН СССР, техн. кибернетика 6 (1975).

[6] —, —, *Методы линейного программирования*, части 1–3, Изд-во БГУ, Минск 1977, 1978, 1980.

[7] —, —, *Особые оптимальные управления*, Наука, Москва 1973.

[8] Р. Габасов, Ф. М. Кириллова, О. И. Костюкова, *Метод решения общей задачи линейного программирования*, Докл. АН БССР 13, 3 (1979).

[9] Л. С. Понтрягин, В. Г. Болтянский, Р. В. Гамкрелидзе, Е. Ф. Мищенко, *Математическая теория оптимальных процессов*, Наука, Москва 1976.

[10] Р. П. Федоренко, *Приближенное решение задач оптимального управления*, Наука, Москва 1978.