САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

КАФЕДРА МОДЕЛИРОВАНИЯ ЭКОНОМИЧЕСКИХ СИСТЕМ

Бойко Алина Владимировна

Выпускная квалификационная работа бакалавра

Реализация алгоритмов адаптивного метода оптимального управления

Направление 010400

Прикладная математика, фундаментальная информатика и программирование

Научный руководитель

доктор физ.-мат. наук,

профессор Смирнов Н. В.

Санкт-Петербург

2017

Содержание

Введение. Обзор литературы

В современном мире многие процессы можно описать математической моделью. Часть параметров модели поддается регулированию, потому ставится вопрос об управлении этими параметрами для достижения лучших результатов. Решение подобных задач заключается в нахождении оптимального управляющего воздействия, удовлетворяющего заданным условиям и обеспечивающего максимум (минимум) критерию качества системы. С появлением всевозможных машин теория оптимального управления получила широкое распространение. Развитие теории оптимального управления связано с такими учеными, как Р. Е Калман [1], Л. С Понтрягин [2], В. И Зубов [3] и многими другими. Применением динамического программирования для решения задач теории оптимального управления занимались Р. Беллман [4], а также Л. А. Петросян и В. В. Захаров [5].

В дальнейшем появилась потребность в нахождении оптимального управления в режиме реального времени, в связи с этим Р. Габасов и его ученики разработали адаптивный метод [6–8]. Данный метод применим к множеству задач, например: задача об оптимальном управлении вращательным движением вала электродвигателя [9], задача успокоения колебательной двухмассовой ситемы [10]. Именно этот метод подробно описан в данной работе и применен к конкретным задачам.

Постановка задачи

Требуется решить задачу оптимального программного управления. А именно, найти такое допустимое управление, доставляющее максимум (минимум) заданному функционалу. Для решения исходную задачу нужно свести к интервальной задаче линейного программирования (ИЗЛП), а затем применить адаптивный метод. В главе 1 подробно изложены все этапы данного подхода, а в главе 2 описаны его приложения к нескольким экономическим задачам.

Глава 1. Адаптивный метод

§1.1. Задача оптимального управления в классе линейных систем

Рассмотрим линейную задачу с начальными условиями:

$$\dot{x} = A(t)x + b(t)u, \tag{1}$$

$$x(t_*) = x_0, \tag{2}$$

$$t \in [t_*, t^*], \quad t_* < t^* < +\infty,$$

где $x \in \mathbb{R}^n$, $u \in \mathbb{R}^1$. $A(t)$, $b(t)$ — кусочно-непрерывные $(n \times n)$-матричная и n-мерная векторная функции, $t \in [t_*, t^*]$.

Для решения задачи (1)—(2) нужно максимизировать целевую функцию

$$c^T x(t^*) \to \max_u, \tag{3}$$

при ограничении

$$g_* \leqslant Hx(t^*) \leqslant g^*, \tag{4}$$

где $g_*,\ g^* \in \mathbb{R}^m$, $m = rangH < n$.

Для определения класса допустимых управлений, зададим шаг дискретизации $h = \frac{t^*-t_*}{N}$, где N — некоторое натуральное число. Отрезок $[t_*, t^*]$ разобъем на N частей длины h. Управление $u(t)$ будем искать в классе кусочно-постоянных функций с периодом квантования h:

$$u(t) = u(t_* + (k-1)h) = u_k, \quad t \in [t_* + (k-1)h,\ t_* + kh), \quad k = \overline{1, N}. \qquad (5)$$

Таким образом, получили задачу оптимального управления (1)—(4), в которой требуется найти такое допустимое управление $u^0(t)$, что $c^T x(t^*, u^0) \geqslant c^T x(t^*, u)$ для $\forall u$ из (5).

§1.2. Процедура сведения к ИЗЛП

Покажем, как задача (1)—(4) может быть сведена к ИЗЛП. По формуле Коши имеем

$$x(t^*, t_*, x_0) = Y(t^*)\left(Y^{-1}(t_*)x_0 + \int_{t_*}^{t^*} Y^{-1}(\tau)b(\tau)u(\tau)d\tau\right), \qquad (6)$$

где $Y(t)$ — фундаментальная матрица системы (1), нормированная в точке t_* ($Y(t_*) = E$). Подставим (6) в максимизируемое выражение (3):

$$c^T Y(t^*)Y^{-1}(t_*)x_0 + \int_{t_*}^{t^*} c^T Y(t^*)Y^{-1}(\tau)b(\tau)u(\tau)d\tau \longrightarrow \max_u .$$

Первое слагаемое есть константа. Его можно не рассматривать, т. к. оно не изменяет характер стремления функции к максимуму. Согласно (5), получаем

$$\sum_{k=1}^{N} \int_{t_*+(k-1)h}^{t_*+kh} c^T Y(t^*)Y^{-1}(\tau)b(\tau)d\tau u_k \longrightarrow \max_u .$$

Домножим теперь (6) на матрицу H слева:

$$Hx(t^*, t_*, x_0) = HY(t^*)\left(Y^{-1}(t_*)x_0 + \int_{t_*}^{t^*} Y^{-1}(\tau)b(\tau)u(\tau)d\tau\right).$$

С учетом (5) граничное условие задачи (4) даёт систему неравенств относительно u_k:

$$g_* - HY(t^*)Y^{-1}(t_*)x_0 \leqslant \sum_{k=1}^{N} \int_{t_*+(k-1)h}^{t_*+kh} HY(t^*)Y^{-1}(\tau)b(\tau)d\tau u_k \leqslant$$
$$\leqslant g^* - HY(t^*)Y^{-1}(t_*)x_0.$$

Введём следующие обозначения:

$$d_k = \int_{t_*+(k-1)h}^{t_*+kh} c^T Y(t^*)Y^{-1}(\tau)b(\tau)d\tau,\ k = \overline{1, N},\quad D = (d_1, \ldots, d_N)^T,$$

$$v_k = \int_{t_*+(k-1)h}^{t_*+kh} HY(t^*)Y^{-1}(\tau)b(\tau)d\tau,\ k = \overline{1, N},\quad V = (v_1, \ldots, v_N),$$

$$U = (u_1, \ldots, u_N)^T,$$

$$w_* = g_* - HY(t^*)Y^{-1}(t_*)x_0,\quad w_* \in \mathbb{R}^m.$$

$$w^* = g^* - HY(t^*)Y^{-1}(t_*)x_0,\quad w^* \in \mathbb{R}^m.$$

Здесь U и D — N-мерные векторы, V — $(m \times N)$-матрица.

Дополнительно введем ограничения на управление:

$$L_* \leqslant u(t) \leqslant L^*.$$

Окончательно получаем интервальную задачу линейного программирования:

$$\begin{gathered} D^T U \to \max_u, \\ w_* \leqslant VU \leqslant w^*, \\ L_* \leqslant u_k \leqslant L^*,\quad k = \overline{1, N}. \end{gathered} \tag{7}$$

Таким образом, задача оптимального программного управления (1)—(4) сводится к ИЗЛП (7).

§1.3. Описание адаптивного метода

Особенностью адаптивного метода является то, что он работает с ИЗЛП, не требуя введения дополнительных переменных и повышения размерности задачи, как это происходит в классическом симплекс-методе. В рассматриваемых моделях структура ИЗЛП специфична: число столбцов много больше числа строк, т.е. множество планов имеет структуру вытянутого многогранника. В таком случае перебор вершин симплекс-методом нецелесообразен, и адаптивный метод является более эффективным. В ходе реализации алгоритма метода Р. Габасова [6] могут использоваться произвольные точки из множества планов, а не только вершины, поэтому метод способен работать с любыми начальными предположениями. По аналогии с базисом в симплекс-методе, для плана в адаптивном методе строится опора, но в отличие от базиса опора не связана с планом. Это позволяет менять их независимо друг от друга и более эффективно строить оптимальный план.

§1.4. Опора

Рассмотрим подробнее как строится опора. Из множества номеров строк I и столбцов J матрицы V из (7) выделим некоторые подмножества $I_o \subseteq I$ и $J_o \subseteq J$. Так, что $\mid I_o \mid = \mid J_o \mid$. Пара $K_o = (I_o, J_o)$ называется опорой, а матрица $V(I_o, J_o)$ называется опорной матрицей, если $detV(I_o, J_o) \neq 0$. Если $I_o = \emptyset$ и $J_o = \emptyset$, то $K_o = (I_o, J_o)$ называется пустой опорой.

§1.5. Алгоритм метода

Для решения задачи (7) мы используем алгоритм прямого опорного метода. Для начала нужно выбрать начальный план U и опору $K_o = (I_o; J_o)$. Пусть $I_n = I \setminus I_o$ и $J_n = J \setminus J_o$.

1 шаг Вычисление векторов потенциалов ξ и оценок Δ:

$$\xi^T(I_o) = D^T(J_o)V^{-1}(J_o, I_o), \quad \xi(I_n) = 0, \tag{8}$$

$$\Delta(J_n) = D(J_n) - V^T(I_o, J_n)\xi(I_o), \quad \Delta(J_o) = 0. \tag{9}$$

2 шаг Проверка критерия оптимальности:

$$u_j = \begin{cases} L_* \ , & \text{при } \Delta(j) < 0; \\ L^* \ , & \text{при } \Delta(j) > 0; \\ \in [L_*, L^*] \ , & \text{при } \Delta(j) = 0, j \in J_n; \end{cases} \tag{10}$$

$$V(i, J)u = \begin{cases} w_* \ , & \text{при } \xi(j) < 0; \\ w^* \ , & \text{при } \xi(j) > 0; \\ \in [w_*, w^*] \ , & \text{при } \xi(j) = 0, i \in I_o. \end{cases} \tag{11}$$

В случае его выполнения, искомое решение найдено и следует завершить алгоритм. Если же он не выполнен, то нужно запомнить индекс $i_0 \in I_o$ или $j_0 \in J_n$, на котором нарушается критерий оптимальности и перейти к следующему шагу.

3 шаг Построение направления изменения плана:

Если на шаге 2 критерий оптимальности (10)—(11) нарушается при каком-то индексе j_0, то полагаем:

$$l(j_0) = sgn(\Delta(j_0)), \quad l(J_n \setminus j_0) = 0,$$

$$l(J_o) = -V^{-1}(J_o, I_o)V(I_o, j_0)l(j_0).$$

Если на шаге 2 критерий оптимальности (10)—(11) нарушается при каком-то индексе i_0, то полагаем:

$$l(J_n) = 0, \quad \psi(i_0) = sgn(\xi(i_0)), \quad \psi(I_o \setminus i_0) = 0,$$

$$l(J_o) = -V^{-1}(J_o, I_o)\psi(I_o).$$

4 шаг Вычисление максимального шага вдоль направления:

$$\lambda^0 = min(\lambda(k), k \in J_o \cup j_0 \cup I_n \cup i_0),$$

$$\lambda(k) = \begin{cases} (L^*(k) - u(k))/l(k)\ , & \text{при } l(k) > 0; \\ (L_*(k) - u(k))/l(k)\ , & \text{при } l(k) < 0; \\ \infty\ , & \text{при } l(k) = 0, k \in J_o \cup j_0; \end{cases}$$

$$\lambda(k) = \begin{cases} (\omega^*(k) - V(k, J)u)/V(k, J)l\ , & \text{при } V(k, J)l > 0; \\ (\omega_*(k) - V(k, J)u)/V(k, J)l\ , & \text{при } V(k, J)l < 0; \\ \infty\ , & \text{при } V(k, J)l = 0, k \in I_n \cup i_0. \end{cases}$$

Если на шаге 2 критерий оптимальности (10)—(11) нарушается при каком-то индексе j_0, то $\lambda(i_o) = \infty$ и, наоборот, если нарушается при индексе i_o, то $\lambda(j_o) = \infty$.

Если $\lambda^0 = \infty$, то целевая функция неограничена и работу нужно завершить.

5 шаг Вычисление нового плана:

$$\widehat{u} = u + \lambda^0 l.$$

6 шаг Замена опоры:

Если $\lambda^0 = \lambda(k^*)$, где $k^* \in J_o \cup I_n$, то опору нужно заменить.

Возможен один из четырех случаев:

1) Если на шаге 2 выбран индекс $j_0 \in J_n, \quad k^* = j^* \in J_o$, то

$$\widehat{I_o} = I_o, \quad \widehat{J_o} = (J_o \setminus j^*) \cup j_0.$$

2) Если на шаге 2 выбран индекс $j_0 \in J_n, \quad k^* = i^* \in I_n$, то

$$\widehat{I_o} = I_o \cup i_0, \quad \widehat{J_o} = J_o \cup j_0.$$

3) Если на шаге 2 выбран индекс $i_0 \in I_o, \quad k^* = j^* \in J_o$, то

$$\widehat{I_o} = I_o \setminus i_0, \quad \widehat{J_o} = J_o \setminus j^*.$$

4) Если на шаге 2 выбран индекс $i_0 \in I_o, \quad k^* = i^* \in I_n$, то

$$\widehat{I_o} = (I_o \setminus i_0) \cup i^*, \quad \widehat{J_o} = J_o.$$

Перейти к 1 шагу.

Глава 2. Приложение адаптивного метода в задачах оптимального управления

§2.1. Неоклассическая модель экономического роста

Рассмотрим модель макроэкономического роста в агрегированной замкнутой экономике [6]. Предполагается, что производится только один продукт. Весь выпуск и затраты находятся внутри системы в любой момент времени. Используются только два фактора производства — капитал (K) и трудовые ресурсы (L); z — горизонт планирования; n — темп роста трудовых ресурсов, постоянный для любого момента времени; μ — норма амортизации капитала; δ — норма дисконтирования, с помощью которой капитал и потребление в различные моменты приводятся к начальному моменту времени $t = 0$; $c = c(t)$ — потребление на одного рабочего; $k = k(t)$ — капиталовооруженность в момент t (количество капитала на одного рабочего $\frac{K}{L}$); $\lambda = \alpha - \mu - n$; k_0 — начальная капиталовооруженность; k^* — минимальный капитал на одного рабочего в конечный момент времени, который необходим для дальнейшего функционирования экономики; Y — объем выпуска производства; α и β — параметры производственной функции:

$$Y = \alpha K + \beta L.$$

Требуется выбрать такую экономическую политику, чтобы максимизировать сумму из потребления за период $T = [0, z]$ и конечного капитала. Тогда математическая постановка задачи выглядит следующим образом:

$$J = e^{-\delta z} k(z) + \int_0^z e^{-\delta t} c(t) dt \to \max, \tag{12}$$

$$\dot{k} = \lambda k - c + \beta, \quad k(0) = k_0, \quad k(z) \geqslant k^*, \tag{13}$$

$$0 \leqslant c(t) \leqslant \alpha k(t) + \beta, \quad t \in T. \tag{14}$$

Эту модель можно рассматривать как линейную задачу оптимального управления. Здесь управлением является $c(t)$, а фазовой переменной — $k(t)$. Управление будем искать в классе кусочно-постоянных функций с периодом квантования $h = \frac{z}{N}$, где N — некоторое натуральное число.

$$c(t) = c(ih) = c_i,\ t \in [ih, (i+1)h),\ i = \overline{0, N-1}. \tag{15}$$

Сведение к ИЗЛП

С помощью преобразований из §1.2 главы 1, задачу (12) — (14) можно свести к интервальной задаче линейного программирования. Заметим, что в задаче имеются динамические ограничения (14) на управление $c(t)$. Рассмотрим правое неравенство из (14). По формуле Коши имеем:

$$k(t) = e^{\lambda t}(k_0 + \int_0^t e^{-\lambda\tau}(\beta - c(\tau))d\tau). \tag{16}$$

Подставим (16) в ограничение (14), получим:

$$0 \leqslant c(t) \leqslant \alpha e^{\lambda t}(k_0 + \int_0^t e^{-\lambda\tau}(\beta - c(\tau))d\tau) + \beta.$$

Перепишем правое неравенство в виде:

$$c(t) \leqslant q(t) - \alpha e^{\lambda t} \int_0^t e^{-\lambda\tau} c(\tau))d\tau,$$

где

$$q(t) = \alpha e^{\lambda t}(k_0 + \beta \int_0^t e^{-\lambda\tau} d\tau) + \beta.$$

Так как (15)

$$c(t) \leqslant q(t) - \alpha e^{\lambda t}(p(0)c(0) + p(h)c(h) + \dots + p(t-h)c(t-h)),$$

где

$$p(t) = \int_t^{t+h} e^{-\lambda\tau} d\tau.$$

Подставим (16) в (12), получим:

$$J = e^{-\delta z} e^{\lambda z}(k_0 + \int_0^z e^{-\lambda\tau}(\beta - c(\tau))d\tau) + \int_0^z e^{-\delta t} c(t)dt \to \max .$$

Отбросим все слагаемые, не зависящие от управления $c(t)$, получим:

$$\int_0^z (e^{-\delta\tau} - e^{-\delta z} e^{\lambda z} e^{-\lambda\tau})c(\tau)d\tau \to \max .$$

Подставим (16) в граничное условие (13), получим:

$$k(z) = e^{\lambda z}(k_0 + \int_0^z e^{-\lambda\tau}(\beta - c(\tau))d\tau) \geqslant k^*.$$

$$-e^{\lambda z} \int_0^z e^{-\lambda\tau} c(\tau)d\tau \geqslant k^* - e^{\lambda z}(k_0 + \beta \int_0^z e^{-\lambda\tau} d\tau).$$

Учитывая (15) окончательно из (12) — (14) получим систему:

$$D^{\mathrm{T}} C \to \max, \tag{17}$$

$$AC \geqslant B, \tag{18}$$

$$0 \leqslant C \leqslant R, \tag{19}$$

где $D = (D_h(h), \ldots, D_h(Nh))^{\mathrm{T}}; \quad D_h(t) = \int_{t-h}^t (e^{-\delta\tau} - e^{-\delta z} e^{\lambda z} e^{-\lambda\tau})d\tau;$

$A = (A_h(h), \ldots, A_h(Nh)); \quad A_h(t) = -\int_{t-h}^t (e^{\lambda z} e^{-\lambda\tau})d\tau;$

$B = k^* - e^{\lambda z}(k_0 + \beta \int_0^z e^{-\lambda\tau}); \quad C = (c_0, \ldots, c_{N-1})^T;$

$R = (R_0, \ldots, R_{N-1})^{\mathrm{T}};$

$R_k = q(t_k) - \alpha e^{\lambda t} L(t_k), \quad k = \overline{0, N-1};$

$L(0) = 0;\; L(t_i) = L(t_{i-1}) + p(t_{i-1})c(t_{i-1}), \quad i = \overline{1, N-1}.$

Численная реализация

На основе адаптивного метода разработан алгоритм, позволяющий найти требуемое управление $c(t)$. В среде MATLAB написана программа. В качестве тестового примера выбраны следующие значения параметров: $z = 5$; $\alpha = 0{,}1$; $\beta = 0{,}9$; $\mu = 0{,}05$; $n = 2$; $\delta = 0{,}09$; $N = 20$; $k_0 = 3{,}8$; $k^* = 4$. В качестве начальной опоры выбран первый элемент вектора A, а в качестве начального плана выбрана левая граница прямых ограничений (т.е. нулевой вектор). Рассматриваемый интервал времени составляет 5 лет, следовательно, период квантования $h = \frac{z}{N}$ составляет 3 месяца.

Результаты

Полученное округленное значение целевой функции равно 3,553.

Оптимальное управление $c(t)$ выглядит так (см. рис. 1):

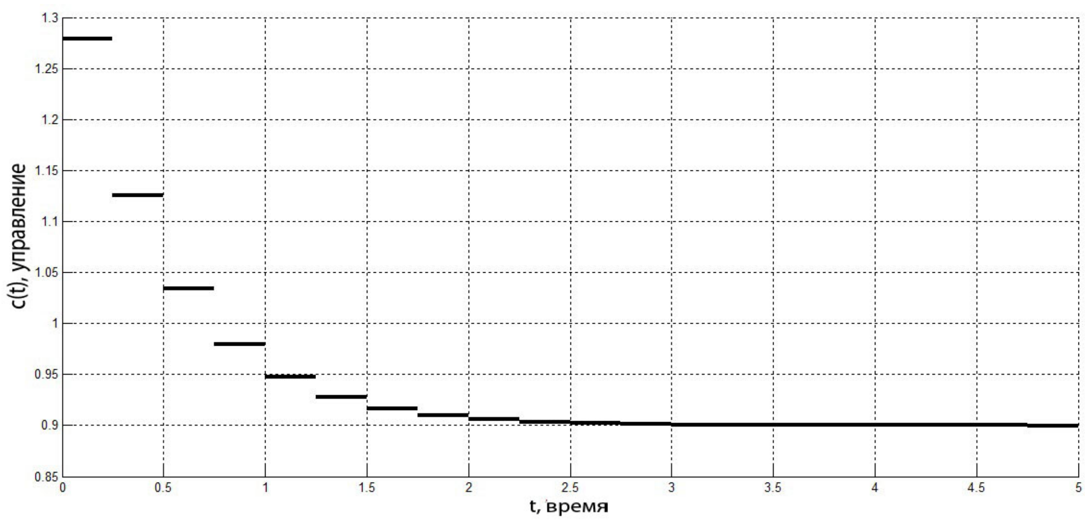

Рис. 1. Оптимальное управление

Изменение целевой функции в ходе работы алгоритма представлено на графике (см. рис. 2):

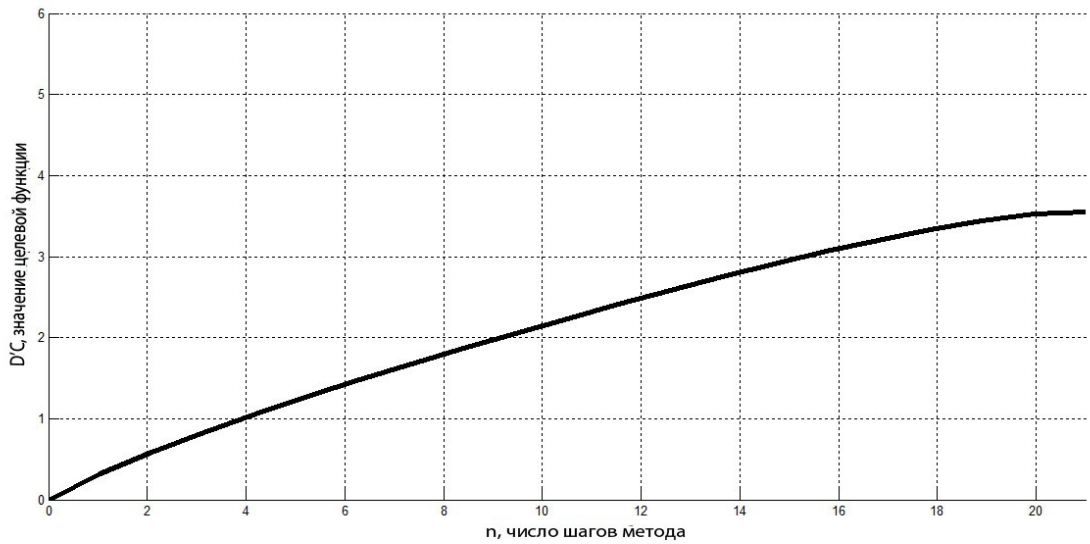

Рис. 2. Изменение целевой функции

Оптимальное управление (см. рис. 1) доставляет максимум функционалу (17), а значит и функционалу (12) исходной задачи (12)—(14).

Потребление убывает с течением времени (см. рис. 1). Это можно объяснить тем, что общественное потребление включает в себя не только заработную плату, но и другие общественные блага (медицинское обслуживание, транспорт, базы отдыха, детские сады, школы и многое другое). В начале требуется большее количество ресурсов, в том числе финансовых (а это налоговая нагрузка), для развития инфраструктуры общественного потребления, а затем ресурсы нужны только на содержание этой инфраструктуры, что потенциально снижает налоговую нагрузку и, как следствие, общественное потребление. При этом заработная плата может оставаться неизменной и даже расти.

Инвестиции

Экономика генерирует добавленную стоимость, частью добавленной стоимости являются запасы товаров и экспортируемая товарная масса. Эти ресурсы можно либо продать и получить внешние инвестиции, либо имитировать внутренние денежные средства и получить внутренние инвестиции. Учитывая это, дополним уравнение (13) постоянной w, получим:

$$J = e^{-\delta z}k(z) + \int_0^z e^{-\delta t}c(t)dt \to \max,$$
$$\dot{k} = \lambda k - c + \beta + w, \quad k(0) = k_0, \quad k(z) \geqslant k^*, \tag{20}$$
$$0 \leqslant c(t) \leqslant \alpha k(t) + \beta, \quad t \in T.$$

Тогда (16) преобразуется в

$$k(t) = e^{\lambda t}(k_0 + \int_0^t e^{-\lambda\tau}(\beta - c(\tau) + w)d\tau).$$

А система (17)—(19) преобразуется в

$$D^{\mathrm{T}}C \to \max,$$
$$AC \geqslant \widehat{B}, \tag{21}$$
$$0 \leqslant C \leqslant \widehat{R},$$

где $\widehat{q}(t) = \alpha e^{\lambda t}(k_0 + (\beta + w)\int_0^t e^{-\lambda\tau}d\tau) + \beta$;
$\widehat{B} = k^* - e^{\lambda z}(k_0 + (\beta + w)\int_0^z e^{-\lambda\tau})$;
$\widehat{R} = (\widehat{R}_0, \ldots, \widehat{R}_{N-1})^{\mathrm{T}}$;
$\widehat{R}_k = \widehat{q}(t_k) - \alpha e^{\lambda t}L(t_k), \quad k = \overline{0, N-1}.$

Результаты

Если взять $w = 10$, а остальные параметры оставить неизменными, то полученное округленное значение целевой функции задачи (20) равно 5,07.

Множество допустимых планов задачи (21) представляет собой вытянутый многоугольник (точками отмечены вершины) (см. рис. 3):

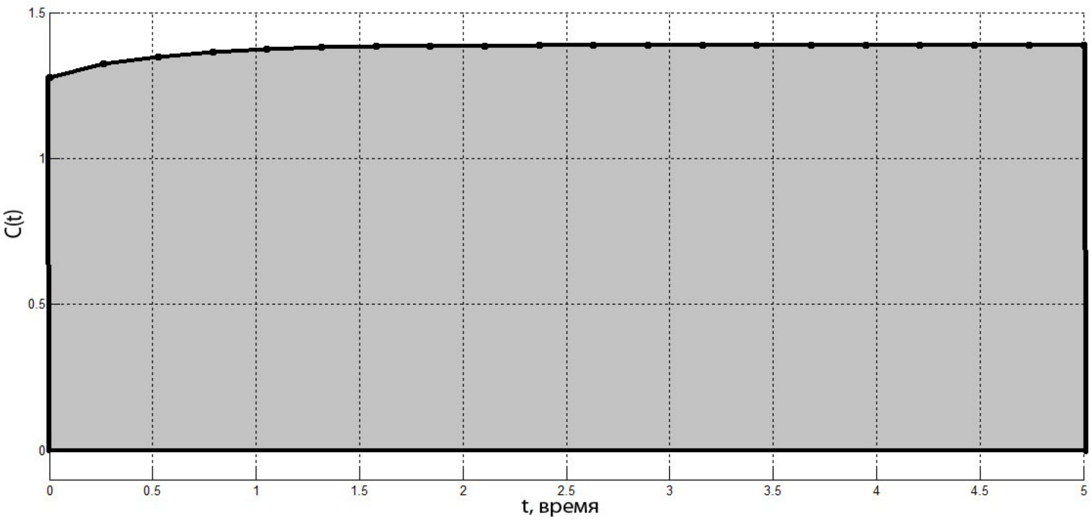

Рис. 3. Множество допустимых планов

Оптимальное управление $c(t)$ задачи (20) выглядит так (см. рис. 4):

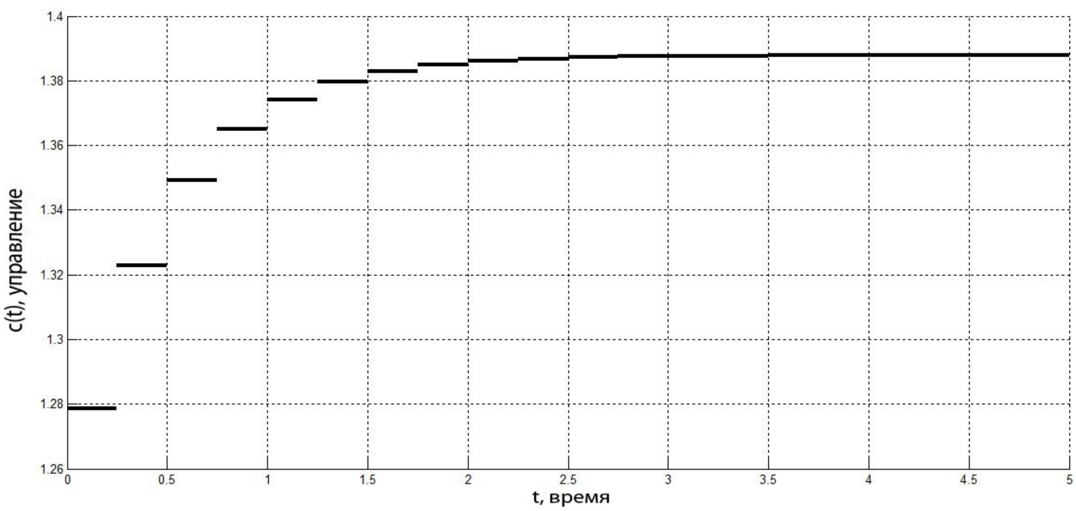

Рис. 4. Оптимальное управление

Изменение целевой функции в ходе работы алгоритма представлено на графике (см. рис. 5):

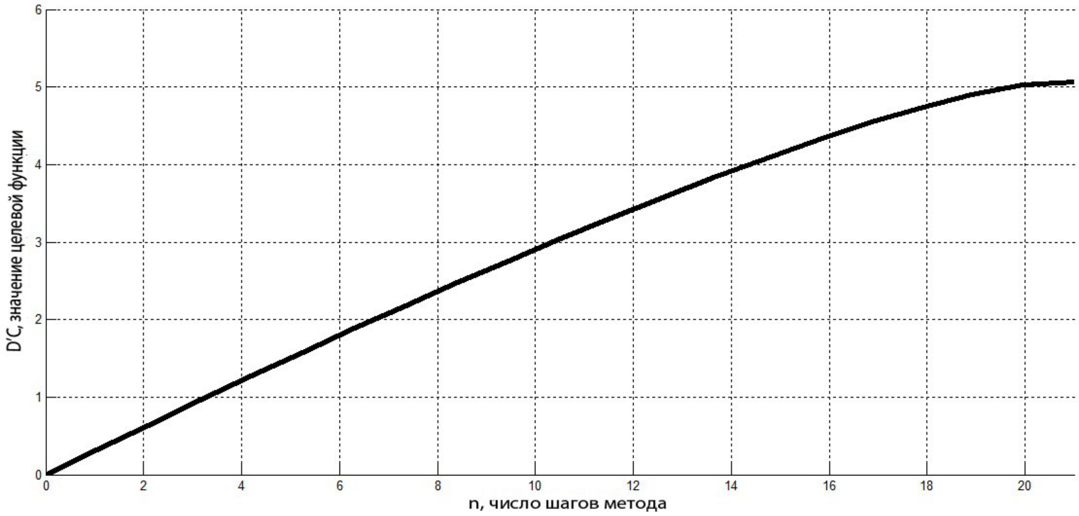

Рис. 5. Изменение целевой функции

Как видно из графика (см. рис. 4) потребление сначала растет, потом уходит в стационарный режим. Таким образом введение переменной w благотворно отразилось на экономике (повысилось значение целевой функции) и благосостоянии людей.

§2.2. Задача построения оптимального распределения капитальных вложений в отрасли

Рассмотрим экономику некоторой отрасли [11]. Обозначим через $K(t)$ величину основных производственных фондов в году t. Величина ΔK прироста основных производственных фондов за промежуток времени Δt будет равна

$$\Delta K = K(t + \Delta t) - K(t).$$

Обозначим через $V(t)$ интенсивность ввода основных производственных фондов, т. е. количество вводимых фондов за единицу времени, например за год. За счет капитальных вложений происходит рост основных производственных фондов $K(t)$, но за счет физического и морального износа их количество уменьшается с течением времени. Будем считать, что интенсивность выбытия основных производственных фондов в году равна $\mu K(t)$. Тогда за промежуток времени Δt будет выведено из производства $\mu K(t)\Delta t$ единиц фондов, и введено $V(t)\Delta t$ единиц новых фондов. Таким образом, уравнение баланса основных производственных фондов будет иметь вид

$$K(t+\Delta t) - K(t) = V(t)\Delta t - \mu K(t)\Delta t.$$

Поделим обе части этого равенства на Δt и устремим Δt к 0. Получим:

$$\frac{dK}{dt} = -\mu K(t) + V(t). \tag{22}$$

Получили модель роста основных производственных фондов отрасли (22). Пусть k_0 — заданное начальное значение основных фондов, k_1 — заданное значение основных фондов в конечный момент, R — известная постоянная величина и $T = [t_0, t_1]$. Введем дополнительные ограничения на $K(t)$ и $V(t)$:

$$K(t_0) = k_0, \tag{23}$$

$$K(t_1) \geqslant k_1, \tag{24}$$

$$0 \leqslant V(t) \leqslant R, \quad t \in T. \tag{25}$$

Введем в рассмотрение критерий качества:

$$F(K(t), V(t)) = \alpha \int_{t_0}^{t_1} V(t)dt - \beta K(t_1) \to \min. \tag{26}$$

Получили линейную задачу оптимального управления (22)—(26), в которой требуется найти такую пару $(K(t), V(t))$, удовлетворяющую (22)—(25) и доставляющую минимум функционалу (26). Здесь управлением является $V(t)$, а фазововой переменной — $K(t)$. Заметим, что функционал (26) состоит из двух слогаемых, значит его минимизация означает, во-первых, экономию капиталовложений, а во-вторых, максимизацию $K(t_1)$ основных фондов в конце рассматриваемого отрезка времени. Числа α и β – это весовые коэффициенты, $\alpha > 0$, $\beta > 0$. Если $\beta > \alpha$, то приоритет отдается первому требованию, если $\alpha > \beta$ — то второму.

Сведение к ИЗЛП

С помощью формулы Коши и некоторых преобразований, эту задачу можно свести к интервальной задаче линейного программирования. Управление $V(t)$ будем искать в классе кусочно-постоянных функций с периодом квантования $h = \frac{t_1-t_0}{N}$, где N — некоторое натуральное число.

$$V(t) = V(t_0 + ih) = V_i,\ t \in [t_0 + ih, t_0 + (i + 1)h),\ i = \overline{0, N-1}. \tag{27}$$

По формуле Коши имеем:

$$K(t) = e^{-\mu t}(e^{\mu t_0}k_0 + \int_{t_0}^{t} e^{\mu\tau}V(\tau)d\tau). \tag{28}$$

Подставим (28) в (26), получим:

$$F(K(t), V(t)) = \alpha \int_{t_0}^{t_1} V(t)dt - \beta e^{-\mu t_1}(e^{\mu t_0}k_0 + \int_{t_0}^{t_1} e^{\mu\tau}V(\tau)d\tau) \to \min.$$

Отбросим все слагаемые, не зависящие от управления $V(t)$, получим:

$$\int_{t_0}^{t_1} (\alpha - \beta e^{-\mu t_1}e^{\mu\tau})V(\tau)d\tau \to \min.$$

С помощью формулы Коши (28) преобразуем (24):

$$K(t_1) = e^{-\mu t_1}(e^{\mu t_0}k_0 + \int_{t_0}^{t_1} e^{\mu\tau}V(\tau)d\tau) \geqslant k_1.$$

Перенесем все слагаемые без управления $V(t)$ в правую часть неравенства, получим:

$$\int_{t_0}^{t_1} e^{-\mu t_1} e^{\mu \tau} V(\tau) d\tau \geqslant k_1 - e^{-\mu t_1} e^{\mu t_0} k_0.$$

Учитывая (27) получим следующее:

$$C^{\mathrm{T}} V \to \min, \tag{29}$$

$$AV \geqslant B, \tag{30}$$

$$0 \leqslant V(t) \leqslant R, \qquad t \in T, \tag{31}$$

где $V = (V_0, \ldots, V_{N-1})$; $B = k_1 - k_0 e^{-\mu(t_1 - t_0)}$;
$C = (C_h(t_0), \ldots, C_h(t_0 + (N-1)h))^{\mathrm{T}}$;
$C_h(t) = \alpha - \beta e^{-\mu t_1} \int_t^{t+h} e^{\mu \tau} d\tau$;
$A = (A_h(t_0), \ldots, A_h(t_0 + (N-1)h))$; $A_h(t) = e^{-\mu t_1} \int_t^{t+h} e^{\mu \tau} d\tau$;
Используя то, что $\max F(x) = -\min(-F(x))$, можем решить задачу (29)—(31) адаптивным методом.

Численная реализация

Подставим в задачу (22)—(26) конкретные значения параметров и найдем оптимальное управление $V(t)$, доставляющее минимум функционалу (26). $t_0 = 0$; $t_1 = 5$; $k_0 = 2$; $k_1 = 6$; $\mu = 0{,}05$; $N = 20$; $\alpha = 2$; $\beta = 1$; $R = 23$.

В качестве начальной опоры выбран первый элемент вектора A, а в качестве начального плана выбрана правая граница основных ограничений (т.е. R). Рассматриваемый интервал времени составляет 5 лет, следовательно, период квантования $h = \frac{t_1 - t_0}{N}$ составляет 3 месяца.

Результаты

Полученное округленное значение целевой функции равно 113,03.

Множество допустимых планов задачи (29)—(31) представляет собой вытянутый многоугольник (точками отмечены вершины) (см. рис. 6):

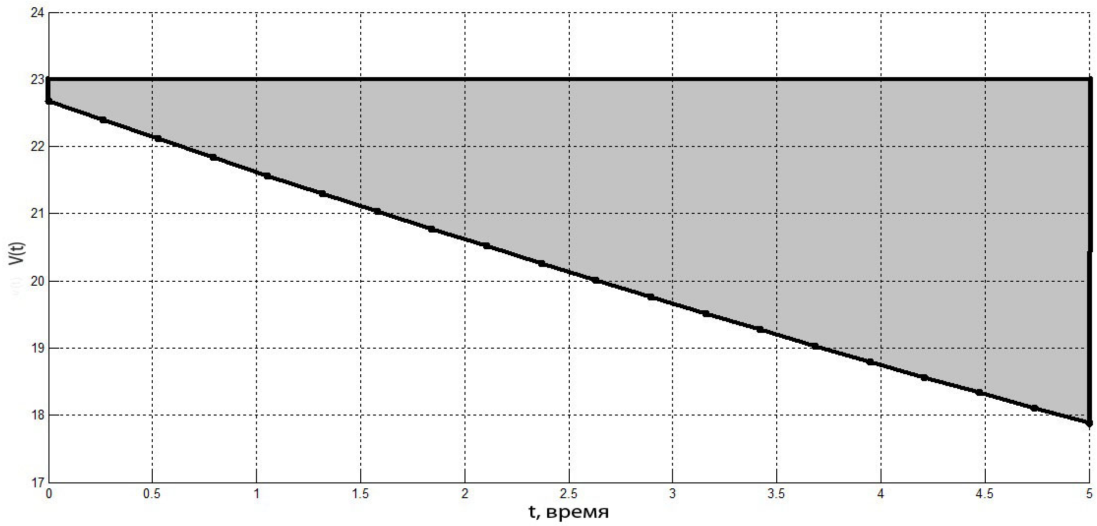

Рис. 6. Множество допустимых планов

Оптимальное управление $V(t)$ задачи (22)—(26) (см. рис. 7):

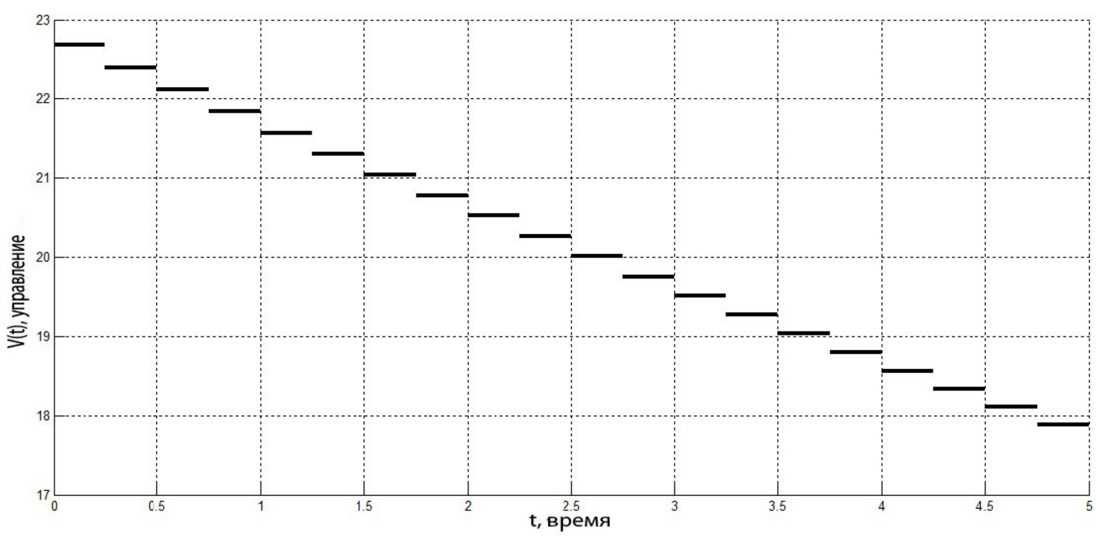

Рис. 7. Оптимальное управление

Изменение целевой функции в ходе работы алгоритма представлено на графике (см. рис. 8):

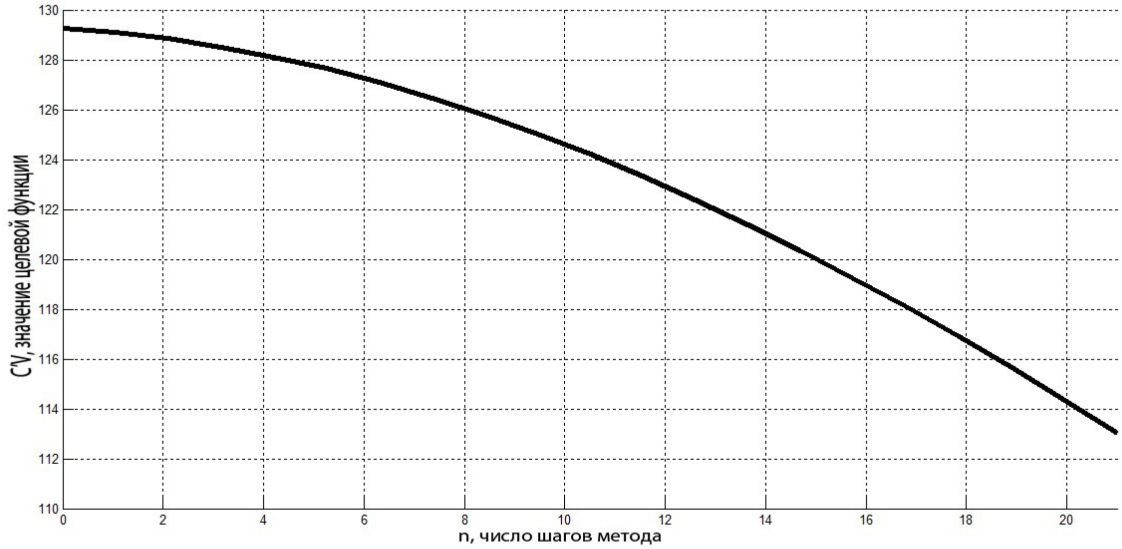

Рис. 8. Изменение целевой функции

Мы нашли оптимальное управление (см. рис. 7), доставляющее минимум функционалу (29), а значит и функционалу (26) исходной задачи (22)—(26).

Заключение

В ходе работы над данным проектом:

1. Изучен адаптивный метод, применяемый к задачам оптимального управления.
2. Метод применен к задаче построения оптимального управления для модели макроэкономического роста. Построено оптимальное управление для этой задачи, результаты приведены на графиках.
3. Метод опробован на задаче построения оптимального распределения капитальных вложений в отрасли, для которой тоже построено оптимальное управление и приведены графики.
4. Для обеих задач разработаны программы в среде MATLAB, фрагменты которых приведены в приложении.
5. Результаты докладывались на конференциях: XLVII Международная научная конференция аспирантов и студентов «Процессы управления и устойчивость» (доклад с публикацией [12]) и XLVIII Международная научная конференция аспирантов и студентов «Процессы управления и устойчивость» (доклад, статья принята к публикации).

Список литературы

1. Калман Р. Е. Об общей теории систем управления. Труды I Междунар. конгресса ИФАК. М.: Изд-во АН СССР, 1961. Т. 2. С. 521–547.

2. Понтрягин Л. С., Болтянский В. Г., Гамкрелидзе Р. В., Мищенко Е. Ф. Математическая теория оптимальных процессов. М.: Наука, 1969. 384 с.

3. Зубов В. И. Математические методы исследования систем автоматического регулирования. Л.: Машиностроение, 1974. 336 с.

4. Беллман Р., Гликсберг И., Гросс О. Некоторые вопросы математической теории процессов управления. М.: ИЛ, 1962. 336 с.

5. Петросян Л. А., Захаров В. В. Математические модели в экологии. СПб.:Изд-во С.-Петербург. ун-та, 1997. 253 с.

6. Альсевич В. В., Габасов Р., Глушенков В. С. Оптимизация линейных экономических моделей. Минск: Изд-во БГУ, 2000. 211 с.

7. Балашевич Н. В., Габасов Р., Кириллова Ф. М. Численные методы программной и позиционной оптимизации линейных систем управления// Журн. вычисл. математики и мат. физики. 2000. Вып. 40, № 6. С. 838–859.

8. Габасов Р. Методы оптимизации: пособие. Минск: Четыре четверти, 2011. 472 с.

9. Попков А. С., Баранов О. В. Об оптимальном управлении вращательным движением вала электродвигателя // Процессы управления и устойчивость. 2014. Т. 1. № 1. С. 31-–36.

10. Клюенков А. Л. Реализация адаптивного метода в одной задаче оптимального управления // Процессы управления и устойчивость. 2015. Т. 2. № 1. С. 53–58.

11. Трошина Н. Ю. Теория оптимального управления [Электронный ресурс]: URL:http://nto.immpu.sgu.ru/sites/default/files/3/__54992.pdf (дата обращения: 16.03.2017).

12. Бойко А. В., Зубаков А. В. Применение адаптивного метода в неоклассической модели экономического роста // Процессы управления и устойчивость. 2016. Т. 3. № 1. С. 607–611.

Приложение

В приложении приведен код функций метода (MatLab).

Алгоритм адаптивного метода

Первый шаг алгоритма

```
function [U, Delta] = step1(c, A, IOP, JOP, INOP, JNOP)
[M, N] = size(A);
U(1:M, 1) = 0;
Delta(1:N, 1) = 0;
[tmp, NOP] = size(IOP);
Atran = A';
Arev = A(IOP, JOP)^-1;
U(INOP) = 0;
Delta(JNOP) = c(JNOP);
if NOP > 0
    crev = (c(JOP));
    U;
    Delta;
    U(IOP) = Arev' * crev;
    Delta(JOP) = 0;
    Delta(JNOP) = Delta(JNOP) - (A(IOP, JNOP))' * U(IOP);
end
end
```

Второй шаг алгоритма

```
function [opt, k0, flag] = step2(x, A, IOP, JNOP, DN, DV, BN, BV, U, Delta)
[tmp, NOP] = size(IOP);
[tmp, NNOP] = size(JNOP);
for p = 1 : 1 : NNOP
    % индекс столбца в опоре
    j = JNOP(p);
    if (x(j) == DN(j) && Delta(j) >= 0) || (x(j) == DV(j) && Delta(j) <= 0) ||
    (x(j) > DN(j) && x(j) < DV(j) && Delta(j) ~= 0)
        opt = false;
        k0 = p;
        flag = 2;
        return
    end
end
for p = 1 : 1 : NOP
    %индекс строки в опоре
```

```
        i = IOP(p);
        tmp = A(i,:) * x;
        if (tmp == BN(i) && U(i) >= 0) || (tmp == BV(i) && U(i) <= 0) ||
        (tmp > BN(i) && tmp < BV(i) && U(i) ~= 0)
            opt = false;
            k0 = p;
            flag = 1;
            return
        end
end
opt = true;
flag = 0;
k0 = 0;
end
```

Третий шаг алгоритма

```
function [L] = step3(flag, k0, A, IOP, JOP, JNOP, U, Delta)
%flag = 1 if k = i0 from IOP
%flag = 2 if k = j0 from JNOP
[M, N] = size(A);
L = zeros(N, 1);
AOP = A(IOP, JOP);
Arev = AOP ^ -1;
if AOP == 0
    Arev=0;
end
if flag == 2
    j0 = JNOP(k0);
    L(j0) = sign(Delta(j0));
    L(JOP) = - Arev * A(IOP, j0) * L(j0);
elseif flag == 1
    i0 = IOP(k0);
    L(i0) = sign(U(i0));
    L(JOP) = - Arev * L(IOP);
end
```

Четвертый шаг алгоритма

```
function [Tetak, zamOpFlag, k] = step4(x, A, IOP, JOP, INOP, JNOP, L, DN, DV, BN, BV, flag, k0)
[tmp, NOP] = size(JOP);
[tmp, NNOP] = size(INOP);
TetaI = []; % первая строка - значения, вторая - индекс
TetaJ = [];
for p = 1:1:NOP
    k = JOP(p);
    TmpTeta = Inf;
    if L(k) > 0
```

```
            TmpTeta = (DV(k) - x(k)) / L(k);
        end
        if L(k) < 0
            TmpTeta = (DN(k) - x(k)) / L(k);
        end
        TetaJ = [TetaJ [TmpTeta; k]];
end
if flag == 2
        k = JNOP(k0);
        TmpTeta = Inf;
        if L(k) > 0
             TmpTeta = (DV(k) - x(k)) / L(k);
        end
        if L(k0) < 0
             TmpTeta = (DN(k) - x(k)) / L(k);
        end
        TetaJ = [TetaJ [TmpTeta; k]];
end
for p = 1:1:NNOP
        k = INOP(p);
        TmpTeta = Inf;
        tmp = A(k, :) * L;
        tmpx = A(k,:) * x;
        if  tmp > 0
            TmpTeta = (BV(k) - tmpx)/tmp;
        end
        if tmp < 0
            TmpTeta = (BN(k) - tmpx)/tmp;
        end
        TetaI = [TetaI [TmpTeta; k]];
end
if flag == 1
        k = IOP(k0);
        TmpTeta = Inf;
        tmp = A(k, :) * L;
        tmpx = A(k,:) * x;
        if tmp > 0;
            TmpTeta = (BV(k) - tmpx)/tmp;
        end
        if tmp < 0
            TmpTeta = (BN(k) - tmpx)/tmp;
        end
        TetaI = [TetaI [TmpTeta; k]];
end
TetaI=double(TetaI);
TetakI = inf;
TetakJ = inf;
if ~isempty(TetaI)
```

```
[TetakI, indexI] = min(TetaI(1,:));
end
if ~isempty(TetaJ)
[TetakJ, indexJ] = min(TetaJ(1, :));
end
zamOpFlag = 0;
k = 0;
if TetakI >= TetakJ;
    Tetak = TetakJ;
    index = TetaJ(2, indexJ);
    if index == JNOP(k0)
        return
    end
    zamOpFlag = 2;
    [k] = find(JOP == index, 1);
else
    Tetak = TetakI;
    index = TetaI(2, indexI);
    if index == IOP(k0)
        return
    end
    zamOpFlag = 1;
    [k] = find(INOP == index, 1);
end
```

Шестой шаг алгоритма

```
function [NEWIOP, NEWJOP, NEWINOP, NEWJNOP] =
step6(IOP, JOP, INOP, JNOP, M, N, flag, zamOpFlag, k0, k)
I = 1:1:M;
J = 1:1:N;
if flag == 2 && zamOpFlag == 2
    NEWIOP = IOP;
    I(NEWIOP) = [];
    NEWINOP = I;
    JOP(k) = [];
    NEWJOP = sort([JOP JNOP(k0)]);
    J(NEWJOP) = [];
    NEWJNOP = J;
end
if flag == 2 && zamOpFlag == 1
    NEWIOP = sort([IOP INOP(k)]);
    NEWJOP = sort([JOP JNOP(k0)]);
    I(NEWIOP) = [];
    J(NEWJOP) = [];
    NEWINOP = I;
    NEWJNOP = J;
end
```

```
if flag == 1 && zamOpFlag == 2
    IOP(k0) = [];
    NEWIOP = IOP;
    I(NEWIOP) = [];
    NEWINOP = I;
    JOP(k) = [];
    NEWJOP = JOP;
    J(NEWJOP) = [];
    NEWJNOP = J ;
end
if flag == 1 && zamOpFlag == 1
    IOP(k0) = [];
    NEWIOP = sort([IOP INOP(k)]);
    I(NEWIOP) = [];
    NEWINOP = I;
    NEWJOP = JOP;
    J(NEWJOP) = [];
    NEWJNOP = J;
end
end
```

Main

```
while 1
    opt = false;
    zamOpFlag = false;
    Tetak = 0;
    L = zeros(N, 1);
    k0 = 1;
    k = 1;
    [U, Delta] = step1(c, A, IOP, JOP, INOP, JNOP);
    [opt, k0, flag] = step2(x, A, IOP, JNOP, DN, DV, BN, BV, U, Delta);
    if opt
        break
    end
    [L] = step3(flag, k0, A, IOP, JOP, JNOP, U, Delta);
    [Tetak, zamOpFlag, k] = step4(x, A, IOP, JOP, INOP, JNOP,L, DN, DV, BN, BV, flag, k0);
    x = x + Tetak * L; %5-й шаг алгоритма
    if zamOpFlag ~= 0
        [IOP, JOP, INOP, JNOP] = step6(IOP, JOP, INOP, JNOP, M, N, flag, zamOpFlag, k0, k);
      end
end
```