EESTI NSV TEADUSTE AKADEEMIA TOIMETISED. XVIII KÕIDE
FÜÜSIKA * MATEMAATIKA. 1969, NR. 2

ИЗВЕСТИЯ АКАДЕМИИ НАУК ЭСТОНСКОЙ ССР. ТОМ XVIII
ФИЗИКА * МАТЕМАТИКА. 1969, № 2

https://doi.org/10.3176/phys.math.1969.2.15

С. УЛЬМ

МЕТОД УСЛОВНОГО ГРАДИЕНТА И ДЕКОМПОЗИЦИЯ ЗАДАЧ ОПТИМАЛЬНОГО УПРАВЛЕНИЯ

S. ULM. TINGLIK GRADIENDIMEETOD JA OPTIMAALSE JUHTIMISE ÜLESANNETE DEKOMPOSITSIOON

S. ULM. THE CONDITIONAL GRADIENT METHOD AND DECOMPOSITION OF OPTIMAL CONTROL PROBLEMS

Для нахождения точки минимума выпуклого функционала $f(u)$ на выпуклом замкнутом ограниченном множестве U в гильбертовом (банаховом) пространстве часто применяется так наз. метод условного градиента. При этом строится следующий итерационный процесс:

1° выбирается начальное приближение $u^{(0)} \in U$;

2° если приближение $u^{(k)}$ найдено, то следующее приближение $u^{(k+1)}$ строится по правилам:

а) решается задача минимизации

$$\min_{u \in U} (\operatorname{grad} f(u^{(k)}), u); \tag{1}$$

б) $u^{(k+1)} = u^{(k)} + \alpha_k (\bar{u}^{(k)} - u^{(k)})$; $0 \leqslant \alpha_k \leqslant 1$, (2)

где $\bar{u}^{(k)}$ — решение задачи (1) $(k = 0, 1, \ldots)$.

Существуют различные возможности для выбора шага α_k.

Метод условного градиента (1)—(2) был предложен для решения задач нелинейного программирования в [1], обобщение для функционалов было дано в работах [2, 3].

В [3] подробно исследуются и условия сходимости метода (1)—(2). Пусть, например, $f(u)$ дифференцируем на U, а $\operatorname{grad} f(u)$ удовлетворяет условию Липшица. Тогда в [3] доказывается, что при подходящем выборе α_k

1) $\lim_{k \to \infty} f(u^{(k)}) = f^* = \inf_{u \in U} f(u)$, 2) $f(u^{(k)}) - f^* \leqslant c/k$ и 3) существует подпоследовательность $u^{(k)}$, слабо сходящаяся к точке минимума u^*.

В [4] метод (1)—(2) использован для декомпозиции больших задач нелинейного программирования. Рассматривается задача

$$\min_{u_i \in U_i} f(u_1, \ldots, u_n), \tag{3}$$

где $f(u) = f(u_1, \ldots, u_n)$ — выпукла; U_i — выпуклые замкнутые ограниченные множества; u_i — в общем векторы.

Обозначим

$$\operatorname{grad} f(u) = (F_1(u), \ldots, F_n(u)). \tag{4}$$

Поскольку в данном случае

$$(\operatorname{grad} f(u^{(k)}), u) = \sum_{i=1}^{n} (F_i(u^{(k)}), u_i), \tag{5}$$

то задача (1) распадается на n независимых задач минимизации:

$$\min_{u_i \in U_i} (F_i(u^{(k)}), u_i) \quad (i = 1, \ldots, n). \tag{6}$$

Это обстоятельство позволяет организовать следующую процедуру решения задачи (3) на двух уровнях.

1° выбирается $u^{(0)} = (u_1^{(0)}, \ldots, u_n^{(0)})$; $u_i^{(0)} \in U_i$;

2° если $u^{(k)}$ найдено, то вычисление $F_i(u^{(k)})$ производится по правилам:

а) второй уровень вычисляет $F_i(u^{(k)})$ и передает его на i-ю подсистему $(i = 1, \ldots, n)$;

б) i-я подсистема решает задачу минимизации

$$\min_{u_i \in U_i} (F_i(u^{(k)}), u_i), \tag{7}$$

решение которой обозначим через $\bar{u}_i^{(k)}$ $(i = 1, \ldots, n)$;

в) по $\bar{u}_i^{(k)}$ второй уровень вычисляет новое приближение по формуле

$$u_i^{(k+1)} = u_i^{(k)} + \alpha_k (\bar{u}_i^{(k)} - u_i^{(k)}) \qquad (i = 1, \ldots, n;\ k = 0, 1, \ldots). \tag{8}$$

Ясно, что рассматриваемую схему можно применить и для решения задачи (3) в гильбертовом пространстве $(U = U_1 \times \ldots \times U_n)$.

Следует еще отметить, что в игровой трактовке методов декомпозиции рассматриваемый процесс совпадает с процессом Б нахождения точки равновесия в соответствующей игре нескольких лиц (см. [5]).

Целью данной заметки является применение метода (1)—(2) для декомпозиции больших задач оптимального управления. При этом рассматриваются такие линейные системы, уравнения движения которых приводимы относительно фазовых координат к блочно-треугольному виду *. Отметим, что подобная задача с квадратичным критерием и без ограничений на управление была рассмотрена в [6], где использован метод декомпозиции, предложенный Мезаровичем и др. в [7].

Рассмотрим задачу: минимизировать по $u = (u_1(t), \ldots, u_n(t))$ функционал

$$f(u) = \int_{t_0}^{T} F(x(t), u(t), t)\, dt + \Phi(x(T)) \tag{9}$$

при ограничениях

$$u_i(t) \in U_i, \tag{10}$$

причем фазовые переменные $x(t) = (x_1(t), \ldots, x_n(t))$ и управления $u_i(t)$ связаны системой линейных уравнений

$$\begin{cases} \dfrac{dx_i}{dt} = A_{ii}(t) x_i + \sum\limits_{j=i+1}^{n} A_{ij}(t) x_j + \sum\limits_{j=1}^{n} B_{ij}(t) u_j, & (11) \\ x_i(t_0) = x_{i0} \ (i = 1, \ldots, n). & (12) \end{cases}$$

Отметим, что $x_i(t)$, $u_i(t)$ — в общем векторы размерностью n_i. Матрицы A_{ij}, B_{ij} имеют размерность $n_i \times n_j$. Предположим, что ограничения $u_i(t) \in U_i$ выделяют в пространствах $L_2^{n_i}[t_0, T]$ выпуклые замкнутые ограниченные множества (примеры таких множеств см. в [3]). В той же ра-

* Необходимые и достаточные условия для того, чтобы матрица была перемещением строк и столбцов приводима к блочно-треугольному виду, даны, например, в [8]. Там же предложен и соответствующий алгоритм.

боте приведены и условия, при выполнении которых $f(u)$ выпуклый, дифференцируемый, а его градиент удовлетворяет условию Липшица.

Справедлива формула [3]

$$\operatorname{grad} f = F_u - B^* \psi, \tag{13}$$

где $\psi(t)$ является решением системы

$$\begin{cases} \dfrac{d\psi_i}{dt} = -\sum\limits_{j=1}^{i} A_{ji}^*(t)\psi_j + F_{x_i}, & (14) \\ \psi_i(T) = -\Phi'_{x_i}(x(T)), & (15) \end{cases}$$

а $B^*(t) = (B_{ji}^*(t))$.

Для реализации метода (1) — (2) ((7) — (8)) нам понадобится вычислить $\operatorname{grad} f$ при фиксированных значениях $u(t) = u^{(k)}(t)$. Для этого надо найти соответствующие $x(t)$ и $\psi(t)$. Поскольку системы (11) и (14) блочно-треугольные, то в данном случае для нахождения $x_i(t)$, $\psi_i(t)$ надо последовательно интегрировать линейные неоднородные системы

$$\begin{cases} \dfrac{dx_i}{dt} = A_{ii}x_i + f_i(t);\ x_i(t_0) = x_{i0}\ (i = n, \ldots, 1), \\ \dfrac{d\psi_i}{dt} = -A_{ii}^*\psi_i + g_i(t);\ \psi_i(T) = -\Phi'_{x_i}(x(T))\ (i = 1, \ldots, n), \end{cases} \tag{16}$$

где

$$f_i(t) = \sum_{j=i+1}^{n} A_{ij}x_j(t) + \sum_{j=1}^{n} B_{ij}u_j(t) \tag{17}$$

и

$$g_i(t) = -\sum_{j=1}^{i-1} A_{ji}^*\psi_j(t) + F_{x_i}(x(t), u(t), t). \tag{18}$$

Нетрудно видеть, что

$$\begin{cases} x_i = X_i(t)x_{i0} + X_i(t)\int\limits_{t_0}^{t} X_i^{-1}(s)f_i(s)\,ds \quad (i = n, \ldots, 1), \\ \psi_i = -[X_i^*(t)]^{-1}X_i^*(T)\Phi'_{x_i}(x(T)) + [X_i^*(t)]^{-1}\int\limits_{T}^{t} X_i^*(s)g_i(s)\,ds \\ \qquad (i = 1, \ldots, n), \end{cases} \tag{19}$$

где $X_i (i = 1, \ldots, n)$ — фундаментальные матрицы соответственно для систем

$$\frac{dx_i}{dt} = A_{ii}x_i(t). \tag{20}$$

По формулам (19) можно, зная $X_i(t)$, $X_i^*(t)$, $X_i^{-1}(t)$, $[X_i^*(t)]^{-1}$, последовательно вычислить $x_n, \ldots, x_1, \psi_1, \ldots, \psi_n$ при фиксированном $u(t)$.

Приведем общую схему * решения задачи (9) — (12) на двух уровнях:

1° выбирается начальное приближение

$$u^{(0)}(t) = (u_1^{(0)}(t), \ldots, u_n^{(0)}(t));\ u_i^{(0)}(t) \in U_i \quad (i = 1, \ldots, n); \tag{21}$$

* В конкретных случаях эту схему можно модифицировать.

2° если приближение $u^{(k)}(t)$ найдено, то вычисление $u^{(k+1)}(t)$ производится по правилам:

а) подсистемы вычисляют по формулам (19) $x_i^{(k)}(t)$, $\psi_i^{(k)}(t)$ (при фиксированном $u = u^{(k)}$) в следующем порядке:

$$
\begin{array}{lcll}
n\text{-я подсистема:} & x_n^{(k)}(t) & \psi_n^{(k)}(t), & \\
 & \downarrow & \uparrow & \\
(n-1)\text{-я} \quad ,, \quad : & x_{n-1}^{(k)}(t) & \psi_{n-1}^{(k)}(t), & (22)\\
 & \downarrow & \uparrow & \\
 & \vdots & \vdots & \\
 & \downarrow & \uparrow & \\
1\text{-я} \quad ,, \quad : & x_1^{(k)}(t) \rightarrow & \psi_1^{(k)}(t); &
\end{array}
$$

б) по этим функциям второй уровень (центр) вычисляет по формуле (13)

$$\operatorname{grad} f(u^{(k)}) = (G_1(u^{(k)}), \ldots, G_n(u^{(k)})), \tag{23}$$

где

$$G_i(u^{(k)}) = F_{u_i} - (B^* \psi)_i; \tag{24}$$

в) подсистемы решают подзадачи оптимизации *

$$\min_{u_i \in U_i} (G_i(u^{(k)}), u_i), \tag{25}$$

где $G_i = (G_{i1}, \ldots, G_{in_i})$; $u_i = (u_{i1}, \ldots, u_{in_i})$;

$$(G_i(u^{(k)}), u_i) = \int_{t_0}^{T} \sum_{j=1}^{n_i} G_{ij}(u^{(k)}(t)) u_{ij}(t) dt; \tag{26}$$

решения подзадач (25) обозначим через $\overline{u}_i^{(k)}(t)$ $(i = 1, \ldots, n)$;

г) по найденным функциям $\overline{u}_i^{(k)}(t)$ второй уровень вычисляет новое приближение

$$u_i^{(k+1)}(t) = u_i^{(k)}(t) + \alpha_k(\overline{u}_i^{(k)}(t) - u_i^{(k)}(t)) \tag{27}$$

$$(i = 1, \ldots, n).$$

Процесс повторяется $(k = 0, 1, \ldots)$.

ЛИТЕРАТУРА

1. Frank M., Wolfe Ph., Nav. Res. Log. Quart., 3, No. 1—2, 95 (1956).
2. Демьянов В. Ф., Рубинов А. М., Вестник ЛГУ, вып. 19, 5 (1964).
3. Левитин Е. С., Поляк Б. Т., Ж. вычисл. матем. и матем. физики, 6, № 5, 787 (1966).
4. Михалевич В. С., Ермольев Ю. М., Шкурба В. В., Шор Н. З., Кибернетика, АН УССР, № 5, 29 (1967).
5. Волконский В. А., Экон. и матем. методы, 1, вып. 2, 195 (1965).
6. Sato T., Ichikawa A., IEEE Trans. Automat. Control, AC-12, No. 4, 457 (1967).
7. Mesarovic M. D., Pearson J. D., Takahara Y., Joint Automat. Control Conf. Rensselaer Polytech. Inst. (Troy, NY), Preprints Papers, 1965, p. 93.
8. Hararу F., Numerische Mathematik, 4, H. 2, 128 (1962).

Институт кибернетики
Академии наук Эстонской ССР

Поступила в редакцию
13/XI 1968

* Решения подзадач существуют, поскольку линейный функционал достигает минимума на ограниченном замкнутом выпуклом множестве гильбертового пространства (ср. [3], теорема 1.2).