УДК 681.513.6

Совершенствование алгоритмов оптимизации решения многокритериальных задач

Ретунская Надежда Владиславовна, Самарастат, Самара, Россия.

Цель данной работы заключается в совершенствовании методов оптимизации. В качестве средств анализа использовались Метод динамического программирования, Аппроксимационно-комбинаторный метод.

Ключевые слова: Математические и инструментальные методы, Метод динамического программирования, Аппроксимационно-комбинаторный метод.

Большинство технико-экономических задач характеризуются сложностью технологического расчета, неопределенностью исходных данных. Кроме того, оценка качества их решений осуществляется, как правило, по многим показателям (критериям), при этом могут быть использованы неформализованные правила оценки. Традиционные способы применения методов оптимизации, позволяющие определить лишь оптимальные решения по какому-либо одному критерию, по перечисленным причинам становятся неприменимыми для решения практических задач. Однако может оказаться, что решения, близкие к оптимальному, т.е. незначительно (на некоторую величину $R \leq 0$) отличающиеся по значению критерия от оптимального значения, лучше удовлетворяют каким-либо дополнительным условиям, больше соответствуют физическому содержанию задачи.

Это делает актуальным в задачах оптимизации не ограничиваться определением лишь оптимального решения, а развивать их с целью определения не только оптимального решения, но и множества всех решений, близких к оптимальному. Знание такого множества решений может быть использовано для решения более сложных задач оптимизации.

Этот подход, предложенный в аппроксимационно-комбинаторном методе, позволил найти ряд модификаций известных методов оптимизации и расширить их применение на решение новых классов задач оптимизации. В частности модификация метода динамического программирования, позволяющая определять не только оптимальную траекторию, но и все траектории, близкие к оптимальной.

Исследуем свойства задачи и опишем алгоритмы ее решения. Найти множество $\Omega_0 \subset \Omega$, обладающее свойством: если $\omega \in \Omega_0$, то для каждого $j = \overline{1, m}$

$$F_j(\omega) - F_j(\alpha_j) \le R_j$$

и если $\omega \in \Omega \setminus \Omega_0$, то существует такое $j = \overline{1, m}$, для которого

$$F_j(\omega) - F_j(\alpha_j) > R_j$$

где

$$F_j(\alpha_j) = \min_{\omega \in \Omega} F_j(\omega),$$

$$F_j(\omega) = \sum_{i=1}^{N} f_i^j (X(i-1), U(i)).$$

Управление, принимаемое в качестве решения, может быть найдено среди множества Ω_0 с помощью неформализованных критериев. Заметим, что

$$\Omega_0 = \bigcap_{j=1}^{m} \Omega_0^j (R_j),$$

где $\Omega_0^j(R_j)$ – множество близких управлений по j -му критерию $j = \overline{1, m}$.

Если обозначим через $\Omega_j(R_j)$ множество близких управлений задачи 3л по j - му критерию, то нетрудно видеть, что

$$\Omega_0 \bigcap_{j=1}^{m} \Omega_j (R_j).$$

Таким образом, для отыскания множества Ω_0 достаточно для каждого $j = \overline{1, m}$ найти алгоритмом 1 (либо 2) множество $\Omega_j(R_j)$ и затем определить их пересечение. Однако такой алгоритм обладает следующими недостатками:

- приходится решать m однотипных задач;

- так как число элементов множеств $\Omega_j(R_j), j=\overline{1,m}$, может быть достаточно велико, а каждый элемент этих множеств представляет собой N-мерный вектор, то алгоритм отыскания множества Ω_0 пересечением множеств $\Omega_j(R_j)$ может быть достаточно трудоемким [51].

Введем для каждого j-го критерия функции Беллмана

$$W_i^j(x_i)=\begin{cases}0, & если \quad i=N,\\ \min\sum_{k=i+1}^{N} f_k^j(x(k-1),u(k)), & если \quad i=N,\end{cases}$$

$$x_i\in X_i, i=\overline{0,N}.$$

где минимум берется по всем частичным управлениям $u(k), k=\overline{i+1,N}$, допустимым для начального состояния x_i; если для некоторого начального состояния x_i не существует допустимых частичных управлений, то будем полагать $W_i^j(x_i)=M_i\geq M$, где M достаточно большое число.

Теорема 3. Если $\tilde{\omega}=\left[\tilde{u}(k), k=\overline{1,N}\right]$ – управление, принадлежащее Ω_0, $\left[\tilde{x}(k), k=\overline{0,N}\right]$ – соответствующая ему траектория, тогда для каждого $j=\overline{1,m}$ и для каждого $i=\overline{0,N-1}$ справедливо неравенство

$$f_i^j\left(x(i),\tilde{u}^i+1\right)+W_{i+1}^j\left(\tilde{x}(i+1)\right)\leq W_i^j(x(i))+R_j.$$

Доказательство. Так как $\tilde{\omega}\in\Omega_0$, то для всякого $j=\overline{1,m}$ $\tilde{\omega}$ – близкое управление, т.е.

$$\sum_{k=1}^{N} f_k^j\left(\tilde{x}(k-1),\tilde{u}(k)\right) \le F_j\left(a_j\right) + R_j,\ j=\overline{1,m}.$$

Но тогда из теоремы 1 следует, что для каждого $i=\overline{0,N-1}$

$$f_i^j\left(\tilde{x}(i),\tilde{u}(i+1)\right) + W_{i+1}^j\left(\tilde{x}(i+1)\right) \le W_i^j\left(x(i)\right) + R_j,\ j=\overline{1,m}.$$

Теорема доказана.

На основе этой теоремы может быть предложен следующий алгоритм решения задачи 3л.

Алгоритм 3. Первый этап.

Для всякого $x_N \in X_N$ и всякого $j=\overline{1,m}$ полагаем $W_N^j(x_N)=0$. Затем последовательно для $i=N-1, N-2,...,0$ для каждого $x_i \in X_i$ выполняем следующие операции:

- если $U_{i+1}(x_i) \neq \phi$, то определяем и запоминаем для каждого $j=\overline{1,m}$:

$$W_i^j(x_i) = \min_{u \in U_{i+1}(x_i)} \left[f_{i+1}^j(x_i,u) + W_{i+1}^j\left(f_{i+1}(x_i,u)\right)\right],$$

если $U_{i+1}(x_i) \neq \phi$, то полагаем $W_i^j(x_i) = M$, $j=\overline{1,m}$;

- если хотя бы для одного $j=\overline{1,m}$ $W_i^j(x_i) \le M$, то определяем и запоминаем множество таких $u \in U_{i+1}(x_i)$, у которых для всякого $j=\overline{1,m}$

$$f_{i+1}^j(x_i,u) + W_{i+1}^j\left(f_{i+1}(x_i,u)\right) \le W_i^j(x_i) + R_j \qquad (1)$$

и обозначаем их через

$$u_{i+1}(x_i,1),...,u_{i+1}(x_i,\xi_{i+1}),...,u_{i+1}(x_i,l(x_i)),$$

где $l(x_i)$ – число запоминаемых $u \in U_{i+1}(x_i), 1 \le \xi_{i+1} \le l(x_i)$. Если это множество оказалось пустым, то полагаем $l(x_i)=0$.

После получения $W_0^j(x_0)$ для каждого $j=\overline{1,m}$ полагаем

$F_j(\alpha_j)=W_0^j(x_0), j=\overline{1,m}$. Если хотя бы одного $j=\overline{1,m}$ $F_j(\alpha_j)\geq M$ (а отсюда следует, что для всякого $j=\overline{1,m}$ $F_j(\alpha_j)\geq M$), то задача не имеет решения и работа алгоритма заканчивается, в противном случае переходим ко второму этапу алгоритма [17].

Второй этап.

- Полагаем $\tilde{x}_0=\tilde{x}_0, i=1$.

- Если $i\geq N$, то если $l\left(\tilde{x}_{i-1}\right)=0$, то переходим к пункту 4, в противном случае полагаем $\xi_i=1$, для всякого $j=\overline{1,m}$ определяем

$$D_i=\begin{cases}0, & при \quad i=1,\\ \sum_{k=1}^{i=1} f_k^j\left(\tilde{x}_{k-1},\tilde{u}_k\right), & при \quad i=2,\end{cases}$$

определяем и запоминаем

$$\tilde{u}_i=u_i\left(\tilde{x}_{i+1},\xi_i\right), \tilde{x}_i=f_i\left(\tilde{x}_{i+1},\tilde{u}_i\right),$$

для каждого $j=\overline{1,m}$ находим $f_i^j\left(\tilde{x}_{i+1},\tilde{u}_i\right)$, если для каждого $j=\overline{1,m}$

$$D+f_i^j\left(\tilde{x}_{i-1},\tilde{u}_i\right)+W_i^j(x_i)\leq F_j(\alpha_j)+R_j,$$

то увеличиваем i на единицу и переходим к пункту 2, в противном случае переходим к следующему пункту, если $\xi_i=l\left(\tilde{x}_{i=1}\right)$, то переходим к пункту 4, в противном случае увеличиваем ξ_i на единицу и переходим к

$$D_i=\begin{cases}0, & при \quad i=1,\\ \sum_{k=1}^{i=1} f_k^j\left(\tilde{x}_{k-1},\tilde{u}_k\right), & при \quad i=2.\end{cases}$$

Если $i > N$, то распечатываем управление

$$\left[\tilde{u}, i = \overline{1, N}\right],$$

которое будет принадлежать множеству Ω_0, либо заносим его в заранее отведенное место, затем полагаем $i = N$ и переходим к пункту 3.

- Если $\xi_i = l\left(\tilde{x}_{x-1}\right)$, то увеличиваем ξ_i на единицу, определяем для каждого $j = \overline{1, m}$

$$D_i = \begin{cases} 0, & при \quad i = 1, \\ \sum_{k=1}^{i-1} f_k^j\left(\tilde{x}_{k-1}, \tilde{u}_k\right), & при \quad i \geq 2, \end{cases}$$

находим и запоминаем

$$\tilde{u}_i = u_i\left(\tilde{x}_{i-1}, \xi_i\right), \tilde{x}_i = f_i\left(\tilde{x}_{i-1}, \tilde{u}_i\right),$$

для каждого $j = \overline{1, m}$ определяем $f_i^j\left(\tilde{x}_{i-1}, \tilde{u}_i\right)$, если существует $j = \overline{1, m}$, для которого

$$D_j + f_i^j\left(\tilde{x}_{i-1}, \tilde{u}_i\right) + W_i^j(x_i) \leq F_j(\alpha_j) + R_j,$$

то переходим к 3, в противном случае увеличиваем ξ_i на единицу и переходим к 2. Если $\xi_i = l\left(\tilde{x}_{x-1}\right)$, то переходим к пункту 4.

- Если $i > 1$, то уменьшаем i на единицу и переходим к пункту 3, если $i = 1$, то конец работы.

Если в процессе работы второго этапа не будет получено ни одного управления, то задача 3л не имеет решения, т.е. $\Omega_0 = \phi$.

Рассмотрим случай, когда удается повысить эффективность алгоритма 3

[54]. Предположим, что известна функция $g_i(x(i),u(i))$, такая, что если $x(i)=f_i(x(i-1),u(i))$, то $x(i-1)=g_i(x(i),u(i))$, и наоборот, если $x(i-1)=g_i(x(i),u(i))$, то $x(i)=f_i(x(i-1),u(i)), i=\overline{1,N}$. В этом случае показатели $F_j(\omega)$ можно записать в виде

$$F_j(\omega)=\sum_{i=1}^{N} g_i(x(i),u(i)),\ j=\overline{1,m},$$

где

$$g_i(x(i),u(i))=f_i^j(g_i(x(i),u(i)),u(i)),\ f_i^j(x(i-1),u(i)).$$

Обозначим

$$\sigma_i(x_i)=\begin{cases}0, & \textit{если} \quad i=0,\\ \min\sum_{k=1}^{i} g_k^j(x(k),u(k)), & \textit{если} \quad i=\overline{1,N},\end{cases}$$

$$x_i\in X_i, i=\overline{0,N},\ j=\overline{1,m},$$

где минимум взят по переменным $u(k), k=\overline{1,i}, x(k), k=\overline{0,i},$ удовлетворяющих условиям (12). Если для некоторого x_i таких переменных не существует, то для определенности положим

$$\sigma_i^j(x_i)=M\geq M.$$

Справедливы следующие соотношения Беллмана:

$$\sigma_i^j(x_i)=\min_{u\in G_i(x_i)}\left[g_i^j(x_i,u)+\sigma_{i-1}(g_i(x_i,u))\right],$$

$$x_i\in X_i, i=\overline{1,N},\ j=\overline{1,m}.$$

Теорема 4. Для того чтобы для некоторого фиксированного $\tilde{x}_i\in X_i, i=\overline{0,N-1},$ значение $\tilde{u}_{i+1}\in U_{i+1}\left(\tilde{x}_i\right)$ являлось компонентой хотя бы одного управления $\tilde{\omega}=\left[\tilde{u}(k), k=\overline{1,N}\right],$ входящего в Ω_0 и для которого

$\tilde{x}_i(i) \in \tilde{x}_i$, необходимо, чтобы

$$\sigma_i^j\left(\tilde{x}_i\right) + f_{i+1}^j\left(\tilde{x}_i, \tilde{u}_{i+1}\right) + W_{i+1}^j\left(f_{i+1}\left(\tilde{x}_i, \tilde{u}_{i+1}\right)\right) \le F_j\left(\alpha_j\right) + R_j,\ j = \overline{1, m}.$$

Доказательство. Пусть $\left[\tilde{u}(k), k = \overline{1, N}\right]$ и $\left[\tilde{x}(k), k = \overline{0, N}\right]$ – такое близкое управление и соответствующая ему траектория, что $\left[\tilde{u}(k), k = \overline{1, N}\right] \in \Omega_0, \tilde{x}_i(i) = \tilde{x}_i, \tilde{u}_i(i+1) = \tilde{u}_{i+1}$. Так как $\Omega_0 = \bigcap_{j=1}^{m} \Omega_j\left(R_j\right)$, то для каждого $j = \overline{1, m}\left[\tilde{u}(k), k = \overline{1, N}\right] \in \Omega_j\left(R_j\right)$, т.е. $\left[\tilde{u}(k), k = \overline{1, N}\right]$ является близким управлением для каждого j-го показателя.

Но тогда в силу теоремы 2 следует справедливость неравенства

$$\sigma_i^j\left(\tilde{x}_i\right) + f_{i+1}^j\left(\tilde{x}_i, \tilde{u}_{i+1}\right) + W_{i+1}^j\left(f_{i+1}\left(\tilde{x}_i, \tilde{u}_{i+1}\right)\right) \le F_j\left(\alpha_j\right) + R_j$$

для всякого $j = \overline{1, m}$.

Заметим, что теорема 4 обобщает необходимые условия теоремы 2 на случай многокритериальных задач. Аналогичное обобщение достаточных условий теоремы 2, вообще говоря, несправедливо.

На основе результатов теоремы 4 приведем следующий алгоритм.

Алгоритм 4. Первый этап.

Для всякого $j = \overline{1, m}$ полагаем $\sigma_0^j(x_0) = 0$. Далее последовательно для $i = 1,2,\dots,N$ для всякого $x_i \in X_i$ выполняем следующие операции: если $G_i(x_i) = \phi$, то полагаем для всякого $j = \overline{1, m}$ $\sigma_i(x_i) = M$ (либо точку x_i исключаем из дальнейшего рассмотрения, т.е. полагаем $X_i = X_i \setminus \{x_i\}$, если $G_i(x_i) = \phi$, то определяем и запоминаем

$$\sigma_i^j(x_i) = \min_{u \in G_i(x_i)} \left[g_i^j(x_i, u) + \sigma_{i-1}^j(g_i(x_i, u)) \right], \; j = \overline{1, m}.$$

После получения $\sigma_N^j(x_N)$ для всякого $x_N \in X_N$ и всякого $j = \overline{1, m}$ полагаем

$$F_j(\alpha) = \min_{x_N \in X_N} \sigma_N^j(x_N).$$

Если хотя бы для одного $j = \overline{1, m}$ $F_j(\alpha_j) \geq M$ (либо $x_N = \phi$); то алгоритм заканчивает работу, в противном случае переходим ко второму этапу.

Второй и третий этапы выполняем так же, как в алгоритме 3 выполнены соответственно первый и второй этапы, заменяя лишь неравенство (1) неравенством

$$\sigma_i^j(x_i) + f_{i+1}^j(x_i, u) + W_{i+1}^j(f_{i+1}(x_i, u)) \leq F_j(\alpha_j) + R_j.$$

Выводы, сделанные при сравнении алгоритмов 1 и 2, остаются в силе и при сравнении алгоритмов 3 и 4.

Список использованных источников:

1. Коваленко А. Г. Элементы выпуклого векторного программирования. // Куйбышев: Куйбышевский государственный университет, 1990. С. 83
2. Изотов Д.А. Эмпирические модели общего экономического равновесия // Пространственная Экономика. 2014. С. 138—167.
3. Полтерович В.М. Кризис экономической теории // Экономическая наука современной России. 1998. № 1. С. 46–66.