ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ В МЕТОДЕ СЕТЕВОГО ПРОГРАММИРОВАНИЯ

В.Н. Бурков, И.В. Буркова
Институт проблем управления им. В.А. Трапезникова РАН,
Россия, 117997, Москва, Профсоюзная ул., д. 65
E-mail: vlab17@bk.ru; irbur27@mail.ru

В докладе описывается новый метод решения задач многоэкстремальной оптимизации – метод сетевого программирования. Его идея состоит в представлении целевой функции в виде суперпозиции более простых функций. Такую суперпозицию удобно представлять в виде сети, в вершинах которой решаются простые (оценочные) задачи. Решение задачи в конченой вершине дает оценку, а в частном случае – оптимальное решение для исходной задачи. Метод сетевого программирования идеально подходит для параллельных вычислений, поскольку оценочные задачи на одном уровне сетевого представления решаются независимо.
В докладе оценивается сокращение времени решения задач целочисленного линейного программирования за счет параллельных вычислений.

PARALLEL COMPUTING IN NETWORK PROGRAMMING METHOD / V.N. Burkov, I.V. Burkova (Institute of Control Sciences of RAS, 65 Profsoyuznaya st., Moscow, 117997, Russia). The paper describes a new method for solving multiextreme optimization problem – the network programming method. Its idea is to represent the objective function as a superposition of more simple functions. Such superposition can be conveniently represented as a network. At the nodes of this network more simple problems (evaluation tasks) are solved. The solution in the terminal node provides an estimate, and in some cases - the optimal solution for the original problem. The network programming method is well suited for parallel computing, since evaluation tasks at the same level of network representations can be solved independently.
In the paper the estimation of time reduction required to solve integer linear programming problems through parallel computing is given.

1. Введение

Метод сетевого программирования разработан для решения задач многоэкстремальной и дискретной оптимизации [1].

Идея метода состоит в представлении целевой функции в виде суперпозиции более простых функций (в частном случае – функций двух переменных). Такую суперпозицию удобно представлять в виде сети, в вершинах которой решаются простые (оценочные) задачи (сетевое представление). Решение задачи в конченой вершине дает верхнюю (нижнюю) оценку для исходной задачи. Если сеть является деревом, то в конечной вершине получается оптимальное решение исходной задачи. Метод сетевого программирования идеально подходит для параллельных вычислений, поскольку оценочные задачи на одном уровне сетевого представления решаются независимо.

В докладе дается оценка сокращения времени решения задач целочисленного линейного программирования, за счет параллельных вычислений.

2. Сетевое представление функций

Рассмотрим следующую задачу дискретной оптимизации: определить вектор $x \in X$, обеспечивающий

$$\max_{x\in X} f\left(x\right) \tag{1}$$

при ограничении

$$\varphi(x) \le b \tag{2}$$

Далее будем предполагать, что $X = \prod_i X_i$, где X_i – дискретное множество чисел.

Любая функция дискретных переменных допускает сетевое представление, такое, что вычисление значений функции сводится к вычислению значений более простых функций. В частности, любая функция дискретных переменных допускает дихотомическое представление, когда вычисление значения функции сводится к последовательному вычислению значений функций двух переменных.

Рассмотрим функцию четырех переменных:

$$f(x) = (x_1x_2 + x_1^2)x_3 + x_3x_4 + x_1x_4. \tag{3}$$

Ее сетевое представление приведено на рис. 1, где $y_1 = x_1x_2 + x_1^2$; $y_2 = y_1x_3$; $y_3 = x_1x_4$; $y_4 = x_3x_4$; $y_5 = y_2 + y_3 + y_4$.

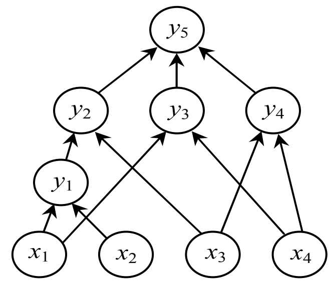

Рис. 1.

Определение 1. Структурой сетевого представления называется сеть, каждой вершине которой соответствует подмножество переменных.

Определение 2. Функции $f(x)$ и $\varphi(x)$ называются структурно-подобными (с-подобными), если существуют сетевые представления этих функций такие, что соответствующие сетевые структуры совпадают.

Рассмотрим функцию

$$\varphi(x) = (x_1 + x_2)x_3(x_1 + x_4) + x_3x_4.$$

Нетрудно показать, что одно из сетевых представлений этой функции имеет вид рис. 1.

Замечание. Фраза "в заданном классе сетевых представлений" в определении 1 является существенной. Если нет ограничений на класс сетевых представлений, то любые две функции можно считать с-подобными. Достаточно записать их в виде:

$$f(x) = f(x) + 0\times\varphi(x);$$
$$\varphi(x) = 0\times f(x) + \varphi(x)$$

и объединить сетевые представления этих функций в одно общее. Однако, пользы в таком сетевом представлении мало.

Класс сетевых представлений может задаваться, например, условием, что структура сетевого представления должна содержать $(2n – 1)$ вершин (n – начальных вершин, соответствующих n переменным, и $(n – 1)$ вершин, соответствующих функциям, входящим в суперпозицию).

3. Метод сетевого программирования

Пусть функции $f(x)$ и $\varphi(x)$ в задаче (1), (2) являются с-подобными. Построим их общее сетевое представление. Пусть далее целевая функция f является монотонной функцией обобщенных переменных y (без ограничения общности можно принять, что f – возрастающая функция y). Аналогично примем, что функция φ также является возрастающей функцией y. В сетевом представлении выделим вершины нулевого уровня, которым соответствуют переменные x_i. Вершинам первого уровня соответствуют задачи оптимизации следующего вида: максимизировать

$$y_i = f_i(x) \tag{4}$$

при ограничении

$$z_i = \varphi_i(x) \le p, \tag{5}$$

где p принимает все допустимые значения.

Как уже отмечалось выше, в сетевом представлении те задачи, которые соответствуют вершинам сетевого представления (за исключением вершин первого уровня) имеют более простой вид, такой, что существуют эффективные алгоритмы их решения. В частности, если сетевое представление является дихотомическим, то каждая задача является задачей оптимизации функции двух переменных, и в дискретном случае легко решается на основе матричного представления. Решив задачи первого уровня, переходим к решению задач второго уровня и т. д. Последней решается задача, соответствующая выходу сети. Обозначим $y_k(b)$ – значение целевой функции в оптимальном решении задачи, соответствующей выходу сети.

Теорема 1. Величина $y_k(b)$ является верхней оценкой для исходной задачи (1), (2).

Доказательство. Достаточно заметить, что любое допустимое решение задачи (1), (2) является допустимым решением для всех задач, решаемых в вершинах сетевого представления. Поэтому оптимальное решение задачи в конечной вершине не хуже, чем оптимальное решение исходной задачи. Это доказывает теорему.

Таким образом, метод сетевого программирования для с-подобных функций позволяет получать верхние оценки для задачи (1), (2).

4. Сетевое представление задачи нелинейного программирования

Рассмотрим задачу нелинейного программирования: определить $x = \left\{x_i, i = \overline{1, n}\right\}$, такой, что

$$f(x) \to \max \tag{6}$$

при ограничениях

$$\varphi_j(x) \le b_j, \quad j = \overline{1, m}; \tag{7}$$

$$x \in X_{m+1}. \tag{8}$$

На рис. 2 приведено сетевое представление ограничений (7), (8) (X_j обозначено j-е ограничение (7), $j = \overline{1, m}$).

Для применения метода сетевого программирования необходимо, чтобы целевая функция имела такую же структуру сетевого представления. Для этого представим $f(x)$ в виде

$$f(x) = \sum_{j=1}^{m} h_j(x) + h_{m+1}(x), \tag{9}$$

где $h_j(x)$ – произвольные функции такие, что представленные ниже задачи (10), (11) имеют решение.

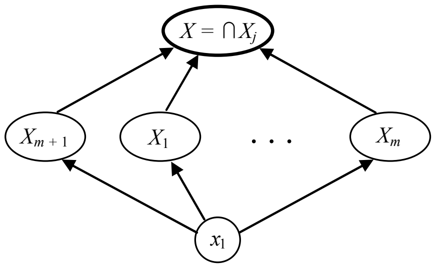

Рис. 2.

В каждой вершине сетевого представления решаются подзадачи оптимизации с одним ограничением. Первые m подзадач имеют вид:

$$\max h_j(x); \quad \varphi_j(x) \le b_j. \tag{10}$$

Последняя $(m + 1)$-я подзадача имеет вид:

$$\max_{x \in X_{m+1}} h_{m+1}(x) = \max_{x \in X_{m+1}} \left[f(x) - \sum_{j=1}^{m} h_i(x) \right]. \tag{11}$$

Обозначим $F_j(h)$ – значение целевой функции в оптимальном решении j-ой подзадачи.

Теорема 1. Функционал

$$F(h) = \sum_{j=1}^{m} F_j(h_j) + F_{m+1}(h) \tag{12}$$

дает верхнюю оценку для исходной задачи.

Доказательство. Все допустимые решения задачи (6)–(8) являются допустимыми для всех подзадач, причем для любого допустимого решения x имеет место

$$\sum_{j=1}^{m+1} h_j(x) = f(x)$$

.

Поэтому $F(h) \ge f(x)$ для любого допустимого x.

Естественно поставить задачу выбора функций $h_j(x)$, $j = \overline{1, m}$, минимизирующих верхнюю оценку (12). Эта задача называется *обобщенной двойственной* для исходной задачи нелинейного программирования. Обоснованием для такого названия являются два факта. Во-первых, как показано ниже (см. пример 1), одним из допустимых решений обобщенной двойственной задачи является минимакс функции Лагранжа, а задачу нахождения минимакса функции Лагранжа часто называют двойственной для задачи нелинейного программирования. Во-вторых, для задачи линейного программирования без требования целочисленности обобщенная двойственная задача является обычной двойственной задачей линейного программирования.

Теорема 2. Функционал $F(h)$ является выпуклым [2].

Таким образом, двойственная задача является задачей выпуклого программирования.

Пример 1. Возьмем одно из допустимых решений двойственной задачи $h_j(x) = \lambda_j \varphi_j(x)$, $j = (\overline{1, m})$.

Первые m подзадач принимают вид: $\lambda_j \varphi_j(x) \to \max$ при ограничении $\varphi_j(x) \le b_j$.

Очевидно, что $F_j(h_j) \le \lambda_j b_j$, $j = \overline{1, m}$. Подставляя в (12), получаем

$$F(\lambda) \le \max_{x \in X_{m+1}} \left(f(x) - \sum_{j=1}^{m} \lambda_j \left(\varphi_j(x) - b_j \right) \right) = \max_{x \in X_{m+1}} L(\lambda, x). \tag{13}$$

Максимизация правой части (13) по λ есть ни что иное как метод множителей Лагранжа. Таким образом, метод множителей Лагранжа дает одно из допустимых решений двойственной задачи (которое в общем случае, не является оптимальным).

В качестве примера рассмотрим задачу целочисленного линейного программирования: определить целочисленный неотрицательный вектор x, максимизирующий

$$C(x) = \sum_{i=1}^{n} c_i x_i \tag{14}$$

при ограничениях

$$\sum_{i=1}^{n} a_{ij} x_i \le b_j\,,\ j = \overline{1, m+1}\,. \tag{15}$$

Примем последнее ограничение в качестве множества X_{m+1}. Разобьем коэффициенты c_i, $i = \overline{1,m}$, на m частей s_{ij}. Примем

$$s_{i;m+1} = c_i - \sum_{j=1}^{m} s_{ij}\,, \quad i = \overline{1,n} \tag{16}$$

Решаем (m+1) подзадач: определить целочисленный неотрицательный вектор x, максимизирующий

$$S_j\left(x\right) = \sum_i s_{ij} x_i \tag{17}$$

при ограничении

$$\sum_{i=1}^{n} a_{ij} x_i \le b_j \tag{18}$$

Обозначим $F_j(s)$ значение $S_j(x)$ в оптимальном решении j-ой подзадачи. Согласно теореме 1

$$F\left(s\right) = \sum_{j=1}^{m} F_j\left(s_j\right) + F_{m+1}\left(s\right) \tag{19}$$

является оценкой сверху $C(x)$: $F\left(s\right) \ge C\left(x\right)$.

***Обобщенная двойственная задача*:** определить

$$\left\{s_{ij}, i=\overline{1,n}, j=\overline{1,m}\right\},$$

минимизирующие (19). Заметим, что если отказаться от требования целочисленности, то задача (19) становится обычной двойственной задачей линейного программирования [1].

Получим необходимые и достаточные условия оптимальности решения двойственной задачи. Пусть s – некоторое решение. Обозначим $P_j(s_j)$ – множество оптимальных решений j-ой подзадачи (17), (18), $j = \overline{1, m+1}$.

Теорема 3. Необходимым и достаточным условием оптимальности решения s является отсутствие решения неравенства

$$\sum_j \max_{x \in P_j\left(s_j\right)} \sum_i y_{ij} x_i < 0 \tag{20}$$

при условиях

$$\sum_{j=1}^{m+1} y_{ij} = 0, \quad i = \overline{1,n}\,. \tag{21}$$

Пример 2. $x_i = 0, 1$; $i = \overline{1,4}$.

$$10x_1 + 8x_2 + 6x_3 + 7x_4 \to \max\,; \tag{22}$$

$$6x_1 + 3x_2 + 2x_3 + 5x_4 \le 11\,; \tag{23}$$

$$3x_1 + 5x_2 + 6x_3 + 3x_4 \le 11\,. \tag{24}$$

Применим метод множителей Лагранжа, т. е. найдем минимум по λ функции

$$11\lambda + \max_{x \in X_2}\left[\left(10-6\lambda\right)x_1 + \left(8-3\lambda\right)x_2 + \left(6-2\lambda\right)x_3 + \left(7-5\lambda\right)x_4\right],$$

где X_2 определяется неравенством (24). При заданном λ это задача "одномерный ранец". Она является NP-трудной, если неизвестна зависимость от n правой части ограничения (24). Однако, на практике, как правило, b_2 либо не зависит от n, либо является линейной функцией n. В этом случае при целочисленных параметрах задача эффективно решается методами динамического либо дихотомического программирования. Имеем оптимальное значение $\lambda_0 = 1\tfrac{2}{9}$, оценка сверху $F_0 = 21\tfrac{1}{3}$. Этому значению λ_0 соответствуют следующие значения s_{ij}, $i = \overline{1,4}$, $j = 1, 2$:

$$\begin{gathered} s_{11} = \lambda_0 a_{11} = 7\tfrac{1}{3}; s_{12} = c_1 - s_{11} = 2\tfrac{2}{3};\\ s_{21} = \lambda_0 a_{21} = 3\tfrac{2}{3}; s_{22} = 4\tfrac{1}{3};\\ s_{31} = \lambda_0 a_{31} = 2\tfrac{4}{9};_{32} = 3\tfrac{5}{9};\\ s_{41} = \lambda_0 a_{41} = 6\tfrac{1}{9}; s_{42} = \tfrac{8}{9}. \end{gathered} \tag{25}$$

Применим метод сетевого программирования.

1 шаг. Выпишем необходимые условия оптимальности для решения (25). Имеем

$$P_1(s_1) = \{(1,1,1,0);(1,0,0,1)\};$$

$$P_2(s_2) = \{(1,1,0,1);(0,1,1,0)\}.$$

Поскольку $y_{i1} + y_{i2} = 0$. то обозначим $y_i = y_{i1} = -y_{i2}$. В этом случае система (20), (21) запишется в виде

$$\max(y_1 + y_2 + y_3; y_1 + y_4) < \min(y_1 + y_2 + y_4; y_2 + y_3).$$

Одно из ее решений: $y_1 = -\varepsilon;\ y_2 = \varepsilon;\ y_3 = -\varepsilon;\ y_4 = 0;\ \varepsilon > 0$.

Возьмем $\varepsilon = 5/6$, так как при $\varepsilon = 5/6$ появляется новое решение во второй подзадаче. Имеем

$$s_{11} = 6\tfrac{1}{2};\quad s_{21} = 4\tfrac{1}{2};\quad s_{31} = 1\tfrac{11}{18};\quad s_{41} = 6\tfrac{1}{9};$$

$$s_{12} = 3\tfrac{1}{2};\quad s_{22} = 3\tfrac{1}{2};\quad s_{32} = 4\tfrac{7}{18};\quad s_{42} = \tfrac{8}{9};$$

$$P_1(s_1) = \{(1,0,0,1);\ (1,1,1,0)\};\ F_1 = 12\tfrac{11}{18};$$

$$P_2(s_2) = \{(1,1,0,1);\ (0,1,1,0);\ (1,0,1,0)\};\ F_2 = 7\tfrac{8}{9};$$

$$F = 20\tfrac{1}{2}.$$

2 шаг. Выпишем условия оптимальности:

$$\max(y_1 + y_2 + y_3; y_1 + y_4) < \min(y_1 + y_2 + y_4; y_2 + y_3; y_1 + y_3).$$

Нетрудно видеть, что это неравенство не имеет решений. Действительно, из условия $y_1 + y_2 + y_3 < y_1 + y_2 + y_4$ следует $y_3 < y_4$, а из условия $y_2 + y_4 < y_2 + y_3$ следует $y_4 < y_3$, что противоречиво. Следовательно, получено оптимальное решение двойственной задачи. Полученную оценку сверху можно использовать в методе ветвей и границ. Возьмем для ветвления временную x_1. Если $x_1 = 1$, то решив соответствующую двойственную задачу, получим ту же оценку $F(x_1 = 1) = 20^1/_2$. В случае $x_1 = 0$ оценка $F(x_1 = 0) = 14$. Выбираем значения $x_1 = 1$ и производим ветвление по переменной x_2. Если $x_2 = 1$, то получаем достижимую оценку $F(x_1 = 1, x_2 = 1) = 18$. Если $x_2 = 0$, то получаем также достижимую оценку $F(x_1 = 1, x_2 = 0) = 17$. Следовательно, оптимальное решение $x_1 = 1$, $x_2 = 1$, $x_3 = 0$, $x_4 = 0$, $C_{\max} = 18$.

5. Оценка вычислительной сложности параллельных вычислений

Оценим выигрыш во времени за счет распараллеливания вычислений для задачи ЦЛП. Во-первых, параллельно решается m оценочных подзадач. Во-вторых, в каждой оценочной задаче решаются $(n–1)$ подзадач с двумя переменными. При максимально симметричной структуре дихотомического представления оценочных подзадач и параллельном решении подзадач одного уровня время решений пропорционально числу уровней, т.е. $\log_2 n$. Таким образом, выигрыш во времени составит

$$q = \frac{m(n-1)}{\log_2 n}$$

Более точные оценки можно получить, если учесть различия в трудоемкости решения оценочных задач при разных b_j. Если решать обобщенную двойственную задачу на основе неравенства (20), то выигрыш во времени уменьшается, что в определенной степени компенсируется получением более точных верхних оценок. Это, в свою очередь, уменьшает число ветвлений в методе ветвей и границ.

Литература

1. Бурков В.Н., Буркова И.В. Метод сетевого программирования. – Проблемы управления № 3, 2005. С. 25-27.
2. Буркова И.В. Метод сетевого программирования в задачах нелинейной оптимизации. – «Автоматика и телемеханика», журнал. 2009. № 10. С. 15-21.