Рекуррентный алгоритм решения задач размещения

Ревотюк М.П., Тиханович Т.В.
Кафедра ИТАС, факультет информационных технологий и управления
Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь
e-mail: rmp@bsuir.by

***Аннотация* – Предложен рекуррентный алгоритм решения комбинаторных задач размещения транспортного типа с использованием наследования решений предшествующих подзадач и коррекцией решений методом потенциалов.**

Ключевые слова: задачи размещения; комбинаторные алгоритмы; метод потенциалов

Задача размещения транспортного типа возникает при выборе мест размещения пунктов производства для удовлетворения потребности в некоторой продукции потребителей с фиксированными объемами потребления. Такая задача в виде

$$\min\left\{\sum_{i\in M}\sum_{j=1}^{n} c_{ij}x_{ij}\ \middle|\ \begin{array}{l}\sum\limits_{i\in M} x_{ij}=b_j,\ j=\overline{1,n};\\ \sum\limits_{j=1}^{n} x_{ij}=a_i,\ i\in M;\ M\in\overline{1,m}\end{array}\right\} \qquad (1)$$

практически может решаться методом перебора среди классических транспортных задач для всех сочетаний строк. Перебор позволяет учесть дополнительные ограничения на варианты размещения, не вписывающиеся в линейную модель (1).

Однако процесс перебора здесь имеет экспоненциальную сложность. Предмет рассмотрения – способ эффективного разбиения задачи (1) на подзадачи с возможностью их параллельного решения. Основная идея предлагаемого способа разбиения – учет взаимозависимости последовательно порождаемых транспортных подзадач

$$\min\left\{\sum_{i\in M}\sum_{j=1}^{n} c_{ij}x_{ij}\ \middle|\ \begin{array}{l}\sum\limits_{i\in M} x_{ij}=b_j,\ j=\overline{1,n};\\ \sum\limits_{j=1}^{n} x_{ij}=a_i,\ i\in M\end{array}\right\} \qquad (2)$$

Пусть конкретный вариант размещения представлен сочетанием $\{v(i)\in M, i=\overline{1,m}\}$. Отдельная подзадача в (2) после новой нумерации строк становится классической транспортной задачей:

$$\min\left\{\sum_{i=1}^{m}\sum_{j=1}^{n} c_{v(i),j}x_{v(i),j}\ \middle|\ \begin{array}{l}\sum\limits_{i=1}^{m} x_{v(i),j}=b_j,\ j=\overline{1,n};\\ \sum\limits_{j=1}^{n} x_{v(i),j}=a_i,\ i=\overline{1,m}\end{array}\right\} \qquad (3)$$

Для решения подзадач (3) предлагается выбрать метод потенциалов, что обусловлено намерением замены процедуры решения отдельной задачи (2) пересмотром решения предшествующей задачи [1,2]. Более быстродействующий для независимого решения транспортных задач венгерский метод менее пригоден для такого пересмотра.

Метод потенциалов, как известно, основан на переходе от задачи (3) к двойственной задаче

$$\max\left\{\sum_{i=1}^{m} u_i+\sum_{j=1}^{n} v_j\ \middle|\ c_{ij}-u_i-v_j\ge 0,\ i=\overline{1,m},\ j=\overline{1,n}\right\} \qquad (4)$$

Схема алгоритма метода потенциалов, как известно, включает начальный этап формирования базисного плана и итерационный процесс уточнения плана [2]. Покажем, что начальный этап после решения задачи (4) для второго и последующих вариантов можно исключить.

Базисный план на начальном этапе с номером $k=0$, а также планы на остальных итерациях, когда $k>0$, должны содержать $m+n-1$ элементов, для которых выполняются условия

$$P^k=\left\{(i,j)\,|\,(v_j^k-u_i^k=c_{ij})\wedge(x_{ij}^k>0),\ i=\overline{1,m},\ j=\overline{1,n}\right\}.$$

Здесь и далее верхний индекс соответствует номеру этапа, когда изменяются множества потенциалов строк и столбцов, а также матрицы корреспонденций.

Построение множества P^k не всегда возможно, но может быть формально выполнено после возмущения исходных данных. Для элементов, не входящими в план перевозок, должно выполняться

$$\overline{P^k}=\left\{(i,j)\,|\,(v_j^k-u_i^k\le c_{ij})\wedge(x_{ij}^k=0),\ i=\overline{1,m},\ j=\overline{1,n}\right\}.$$

Условие оптимальности плана – выполнение условия

$$\left|P^k\right|=m+n-1 \qquad (5)$$

Рекуррентный процесс поиска оптимального плана строится на решении системы уравнений $(v_j^k-u_i^k=c_{ij}),(i,j)\in P^k\wedge v_0^k=0$. Далее вместо (5) достаточно проверять другое условие завершения процесса: $\left|\overline{P^k}\right|\ne mn-m-n+1$. Если оно выполнено, то индексы $(i^{k+1},j^{k+1})=\left\{(i,j)\middle|\max(v_j^k-u_i^k-c_{ij},(i,j)\notin P^k)\right\}$ определяют элемент, вводимый в опорный план P^{k+1}. Из плана P^k при этом выводится элемент с индексами $(i^k,j^k)=\left\{(i,j)\middle|\max(x_{ij}^k,\ (i,j)\in P^k)\right\}$.

Так как на каждой итерации выполнение условия (5) не зависит от значений матрицы, то при переходе к новому варианту сочетания из итераций поиска оптимума можно исключить этапы формирования базисного плана. Как показывают эксперименты, при порождении сочетаний методом вращающейся двери время решения задачи (1) сокращается в *m* раз.

[1] Юдин, Д.Б. Задачи и методы линейного программирования/ Д.Б. Юдин, Е.Г. Гольштейн//М.: Советское радио, 1964. – 736 с.

[2] Brenner, U. A faster polynomial algorithm for the unbalanced Hitchcock transportation problem/U. Brenner//Operations Research Letters, vol. 36(4), 2008. – pp. 408-413.