ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Май—июнь 2008. Том 15, № 3. 22–30

УДК 519.87+519.854

О ДЕЦЕНТРАЛИЗОВАННОЙ ТРАНСПОРТНОЙ ЗАДАЧЕ*)

В. Т. Дементьев, А. В. Пяткин

Аннотация. Рассматривается децентрализованная транспортная задача, когда потребители действуют индивидуально, максимизируя каждый свою собственную выгоду, а производитель определяет только очерёдность их обслуживания. Показывается, что данная задача NP-трудна, и предлагается эффективный приближённый алгоритм с оценкой точности решения для случая одинаковых объёмов спроса.

Ключевые слова: транспортная задача, двухуровневое программирование, алгоритмическая сложность, NP-полнота, приближённый алгоритм.

Введение

Классическая транспортная задача заключается в максимизации функции

$$S_C = \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} x_{ij} \tag{1}$$

при ограничениях

$$\sum_{j=1}^{m} x_{ij} = a_i, \ \sum_{i=1}^{n} x_{ij} = b_j, \quad x_{ij} \geqslant 0; \ i = 1, 2, \ldots, n; \ j = 1, 2, \ldots, m. \tag{2}$$

Она моделирует централизованное распределение m видов производимой продукции в количествах $b_j, \ j = 1, 2, \ldots, m$, среди n потребителей с объёмами спроса $a_i, \ i = 1, 2, \ldots, n$. Здесь и далее предполагается, что $\sum\limits_{i=1}^{n} a_i = \sum\limits_{j=1}^{m} b_j$. Через $c_{ij} \geqslant 0$ обозначается прибыль, получаемая производителем при продаже одной единицы j-го вида продукции i-му потребителю. Классическая транспортная задача является полиномиально разрешимой [2].

*)Исследование выполнено при финансовой поддержке Российского фонда фундаментальных исследований (проекты 08–01–00516 и 08–01–00370).

В статье рассматривается децентрализованная транспортная задача, в которой предполагается, что каждый потребитель действует индивидуально, максимизируя свою собственною выгоду. Производитель может контролировать лишь очерёдность обслуживания (доступа) потребителей. В настоящей работе предполагается, что потребители имеют ту же матрицу доходов (c_{ij}), что и производитель. Ясно, что выгода каждого потребителя зависит от очерёдности, в которой он будет допущен к производителю. При заданной очерёдности обслуживания π, которую можно интерпретировать как перестановку строк матрицы (c_{ij}), доходы потребителей легко находятся из последовательного решения следующих задач

$$S_i(\pi) = \max_{x_{\pi(i)j}} \left\{ \sum_{j=1}^{m} c_{\pi(i)j} x_{\pi(i)j} \;\middle|\; \sum_{j=1}^{m} x_{\pi(i)j} = a_{\pi(i)}, \right.$$
$$\left. 0 \leqslant x_{\pi(i)j} \leqslant b_j - \sum_{l=1}^{i-1} x^*_{\pi(l)j} \right\}, \quad (3)$$

где через x^*_{ij} обозначен оптимальный выбор i-го потребителя. Тогда доход производителя равен суммарному доходу потребителей, т. е.

$$S_D(\pi) = \sum_{i=1}^{n} S_i(\pi) = \sum_{i=1}^{n} \sum_{j=1}^{m} c_{\pi(i)j} x^*_{\pi(i)j}. \quad (4)$$

Требуется найти $S^*_D = \max\limits_{\pi} S_D(\pi)$. Поскольку каждое допустимое решение задачи (3)–(4) является допустимым решением задачи (1)–(2), имеем $S^*_D \leqslant S^*_C$, где через S^*_C обозначен оптимум классической транспортной задачи (1)–(2).

В следующем разделе показано, что децентрализованная транспортная задача в отличие от классической является NP-трудной даже в том случае, когда $a_i = 2,\ i = 1, 2, \ldots, n$, и $b_j = 1,\ j = 1, 2, \ldots, m$. В разд. 3 приводится приближённый алгоритм решения задачи (3)–(4) в случае, когда $a_i = k,\ i = 1, 2, \ldots, n$, и $b_j = 1,\ j = 1, 2, \ldots, m$. Алгоритм имеет трудоёмкость $O(kn^3)$ и гарантирует получение решения, не более чем в k раз худшего оптимального. Построены два класса матриц. Для матриц одного класса выполняется соотношение $S^*_D = S^*_C$, но алгоритм находит решение, в k раз худшее оптимального. Для матриц другого класса алгоритм находит оптимальное решение, но имеет место соотношение $S^*_D \approx S^*_C/k$. Тем самым показано, что константа в оценке точности

алгоритма неулучшаема, а рассматриваемая дисциплина обслуживания потребителей может оказаться в наихудшем случае во много раз невыгоднее централизованной системы закупок (поставок).

1. Доказательство NP-полноты

В этом разделе доказывается NP-полнота следующей задачи распознавания.

Задача 1. *Даны матрица C размера $n \times 2n$ и целое число K. Обозначим через c_{1i} и c_{1j} два наибольших элемента первой строки и назовём весом матрицы C величину $f(C) = c_{1i} + c_{1j} + f(C')$, где матрица C' получается из C удалением первой строки, а также i-го и j-го столбцов. Существует ли такая перестановка строк матрицы C, что получившаяся матрица имеет вес не меньше K?*

Нетрудно заметить, что задача 1 эквивалентна частному случаю задачи (3)–(4) при $a_i = 2,\ i = 1, 2, \ldots, n$, и $b_j = 1,\ j = 1, 2, \ldots, 2n$. Заметим, что в случае, когда $a_i = b_j = 1,\ i, j = 1, 2, \ldots, n$, задача (3)–(4) эквивалентна задаче о назначениях. Это следует из того факта, что в оптимальном решении задачи о назначениях всегда найдётся элемент, являющийся максимальным в своей строке. Отметим, что задача о назначениях разрешима за полиномиальное время [3].

Принадлежность задачи 1 классу NP очевидна. Для доказательства её NP-полноты воспользуемся следующей известной NP-полной задачей Выполнимость [1].

Задача Выполнимость. *Дано q дизъюнкций над множеством из p переменных. Можно ли назначить этим переменным такие значения истинности, чтобы каждая дизъюнкция содержала по крайней мере один истинный литерал (под литералом понимается переменная или её отрицание)?*

Нам потребуется сбалансированная версия этой задачи, где каждый литерал входит ровно в t дизъюнкций. Покажем, что при таком дополнительном условии задача Выполнимость остаётся NP-полной.

Утверждение 1. *Сбалансированная версия задачи Выполнимость NP-полна.*

Доказательство. Рассмотрим произвольный пример задачи Выполнимость и обозначим через t максимальное число вхождений литерала в дизъюнкции. Пусть x — произвольный литерал, входящий в $s < t$ дизъюнкций (если таких литералов нет, то пример уже сбалансирован).

Добавим новую переменную y, а также $t-s$ копий дизъюнкции $x \vee y \vee \overline{y}$ и s копий дизъюнкции $y \vee \overline{y}$. Ясно, что полученный набор дизъюнкций эквивалентен исходному, но литералы x, y и $\overline{y}$ встречаются ровно по t раз. Проведя эту процедуру для каждого литерала, встречающегося менее чем t раз, получим сбалансированный пример задачи Выполнимость. Утверждение 1 доказано.

Теперь можно доказать основной результат данного раздела.

Теорема 1. *Задача* 1 NP-*полна.*

Доказательство. Построим сведение сбалансированной задачи Выполнимость к задаче 1. Рассмотрим некоторый сбалансированный набор из q дизъюнкций над множеством из p переменных, где каждый литерал встречается ровно t раз. Положим $b = 4tp + 2$ и $a = 2tpb + b + 2$. Каждой переменной x_i поставим в соответствие $2t$ строк (по t строк для каждого из литералов x_i и $\overline{x}_i$) и $5t$ столбцов. Столбец матрицы будем называть *большим*, *средним* или *малым*, если максимальный элемент этого столбца равен a, b или 1 соответственно. Каждой переменной x_i соответствуют $2t$ больших, t средних и $2t$ малых столбцов (по t из которых соответствуют литералам x_i и $\overline{x}_i$). Строки и столбцы, соответствующие литералам x_i и $\overline{x}_i$, имеют соответственно вид (элементы всех остальных столбцов в этих строках будут равны 0):

$$(A_{x_i}, B, I_t, 0_t) \text{ и } (A_{\overline{x}_i}, B, 0_t, I_t),$$

где

$$A_{x_i} = \begin{pmatrix} a & a & \frac{a}{2} & \frac{a}{2} & \dots & \frac{a}{2} & \frac{a}{2} \\ \frac{a}{2} & \frac{a}{2} & a & a & \dots & \frac{a}{2} & \frac{a}{2} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \frac{a}{2} & \frac{a}{2} & \frac{a}{2} & \frac{a}{2} & \dots & a & a \end{pmatrix}, \quad B = \begin{pmatrix} b & 0 & \dots & 0 \\ 0 & b & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & b \end{pmatrix},$$

$$A_{\overline{x}_i} = \begin{pmatrix} \frac{a}{2} & a & a & \frac{a}{2} & \dots & \frac{a}{2} & \frac{a}{2} \\ \frac{a}{2} & \frac{a}{2} & a & a & \dots & \frac{a}{2} & \frac{a}{2} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ a & \frac{a}{2} & \frac{a}{2} & \frac{a}{2} & \frac{a}{2} & \dots & a \end{pmatrix}, \quad I_t = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & 1 \end{pmatrix},$$

0_t — квадратная нулевая матрица порядка t.

Для каждой дизъюнкции назначим по одному малому столбцу, соответствующему каждому входящему в эту дизъюнкцию литералу. Столбцы назначаются произвольным образом, но так, чтобы разным дизъюнкциям были назначены разные столбцы. Каждой дизъюнкции поставим в соответствие одну строку. Её элементы во всех больших и средних

столбцах, соответствующих переменным, положим равными $a/2$ и $b/2$ соответственно. Элементы строки в назначенных для данной дизъюнкции малых столбцах положим равными 1. Добавим также для каждой дизъюнкции один большой столбец с элементом a в соответствующей строке. Все остальные элементы строк, соответствующих дизъюнкциям, положим равными 0.

В результате получим матрицу с $2tp+q$ строками, $2tp+q$ большими столбцами (из которых $2tp$ соответствуют переменным и q дизъюнкциям), tp средними столбцами и $2tp$ малыми столбцами. Добавим в случае необходимости полностью нулевые строки и/или столбцы, чтобы число столбцов матрицы стало ровно вдвое больше числа строк. Положим $K=(2tp+q)a+tpb+tp+q$. Покажем, что набор дизъюнкций выполним тогда и только тогда, когда приведённая выше матрица при некоторой перестановке строк имеет вес не меньше K.

Предположим, что набор дизъюнкций выполним. Тогда переставим строки матрицы в таком порядке: сначала поставим строки, соответствующие истинным литералам, затем — ложным литералам, затем — дизъюнкциям, и наконец, полностью нулевые строки, если таковые имеются. Тогда в каждой строке, соответствующей истинному литералу, дважды выбирается элемент a, после чего все большие столбцы, соответствующие переменным, будут удалены. Поэтому во всех строках, соответствующих ложным литералам, выбираются элементы b и 1, что приводит к удалению всех средних столбцов, а также всех малых столбцов, соответствующих ложным литералам. Заметим, что ни один малый столбец, соответствующий истинному литералу, удалён не был. Поскольку каждая дизъюнкция содержит хотя бы один истинный литерал, во всех строках, соответствующих дизъюнкциям, можно выбрать элементы a и 1. Таким образом, вес матрицы равен $2atp+tp(b+1)+q(a+1)=K$.

Допустим теперь, что при некоторой перестановке строк получается матрица веса $f^*\geqslant K$. Сначала покажем, что строки, соответствующие дизъюнкциям, идут после строк, соответствующих переменным. Оценим величину f^* сверху как сумму максимальных элементов всех столбцов. По выбору a и b имеем

$$f^*\leqslant(2tp+q)a+tpb+2tp<(2tp+q)a+tpb+b/2<(2tp+q)a+a/2.$$

Следовательно, для выполнения неравенства $f^*\geqslant K$ необходимо, чтобы в каждом большом столбце был выбран элемент a, а в каждом среднем столбце — элемент b. В частности, отсюда следует, что выбор максимальных элементов в строке, соответствующей дизъюнкции, дол-

жен производиться после того, как будут удалены все большие и средние столбцы, соответствующие переменным. Рассмотрим произвольную переменную x. Ей соответствуют $2t$ строк. В первых t строках выбираются два элемента из больших столбцов (в результате все большие столбцы, соответствующие x, удаляются). В следующих t строках выбираются по одному элементу из среднего и малого столбцов. Таким образом, все большие и средние столбцы, соответствующие x, будут удалены лишь после выбора двух максимальных элементов в последней из строк, соответствующих x. Следовательно, ни одна строка, соответствующая дизъюнкции, не может встречаться раньше строки, соответствующей переменной.

Поскольку в каждом большом столбце, соответствующем переменной x, должен быть выбран элемент a, то либо все строки, соответствующие литералу x, должны идти перед строками, соответствующими литералу $\overline{x}$, либо наоборот, все строки, соответствующие $\overline{x}$, должны идти перед строками, соответствующими x. Положим значение переменной x равным «истина», если имеет место первая альтернатива, и «ложь» в противном случае. Заметим, что все малые столбцы, соответствующие ложным литералам, были удалены. Сумма элементов, выбранных в строках, соответствующих переменным, равна $2tpa + tpb + tp$. Ясно, что в строке, соответствующей дизъюнкции, выбирается элемент a из соответствующего ей большого столбца и элемент 1 из одного из назначенных ей малых столбцов, если хотя бы один из этих столбцов не был удалён. Таким образом, сумма элементов, выбранных в этих строках, не превосходит $q(a+1)$. Поскольку $f^* \geqslant K = 2tpa + tpb + tp + q(a+1)$, в каждой строке, соответствующей дизъюнкции, должен быть элемент 1. А это значит, что каждая дизъюнкция содержит истинный литерал. Теорема 1 доказана.

2. Приближённый алгоритм

Рассмотрим следующий приближённый алгоритм решения задачи (3)–(4) в случае, когда $a_i = k$, $i = 1, 2, \ldots, n$, $k \geqslant 2$, и $b_j = 1$, $j = 1, 2, \ldots, m$.

Описание алгоритма

Шаг 0. Решается классическая транспортная задача (1)–(2) с матрицей (c_{ij}) при $a_i = k$, $i = 1, 2, \ldots, n$, и $b_j = 1$, $j = 1, 2, \ldots, m$. Пусть U_1 — множество элементов матрицы, соответствующих ненулевым значениям переменных в оптимальном решении задачи (1)–(2). Тогда U_1 содержит по k элементов каждой строки и по одному элементу каждого столбца матрицы (c_{ij}).

Шаг i (для $i = 1, 2, \ldots, n$). Обозначим через x_i максимальный элемент в U_i и положим $\pi(i)$ равным номеру строки, содержащей x_i. Пусть $y_1^i = x_i \geqslant y_2^i \geqslant \ldots \geqslant y_j^i$ — элементы строки $\pi(i)$, лежащие в U_i (очевидно, $j \leqslant k$). Выберем k максимальных элементов в строке $\pi(i)$. Обозначим их через $z_1^i \geqslant z_2^i \geqslant \ldots \geqslant z_k^i$. Ясно, что $z_l^i \geqslant y_l^i$ для всех $l = 1, 2, \ldots, j$. Удалим из U_i элементы $y_1^i, y_2^i, \ldots, y_j^i$, а также элементы тех столбцов, которые содержат $z_1^i, z_2^i, \ldots, z_k^i$. Обозначим полученное множество через U_{i+1}. Если $U_{i+1} \neq \varnothing$, то переходим на шаг $i+1$. В противном случае, значения $\pi(i+1), \pi(i+2), \ldots, \pi(n)$ назначаются произвольно из числа тех, которые не были выбраны ранее, и алгоритм заканчивает свою работу.

Оценку качества работы алгоритма даёт следующая

Теорема 2. *Алгоритм за время $O(kn^3)$ находит приближённое решение децентрализованной транспортной задачи с матрицей (c_{ij}) при $c_{ij} \geqslant 0$, $a_i = k$ и $b_j = 1$, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, m$, отличающееся от оптимального не более чем в k раз.*

Доказательство. Трудоёмкость алгоритма определяется решением классической транспортной задачи на шаге 0, поскольку остальные шаги алгоритма выполняются за линейное время. Шаг 0 требует $O(kn^3)$ элементарных операций [2].

Оценим погрешность алгоритма. Через s_i обозначим сумму удалённых на i-м шаге элементов из U_i. Достаточно показать, что для каждого i выполняется соотношение $s_i \leqslant k\left(\sum\limits_{l=1}^{k} z_l^i\right)$. В этом случае

$$S_D(\pi) = \sum_{i=1}^{n}\sum_{l=1}^{k} z_l^i \geqslant \left(\sum_{i=1}^{n} s_i\right)/k = \Big(\sum_{u \in U_1} u\Big)/k = S_C^*/k \geqslant S_D^*/k.$$

Чтобы оценить s_i, рассмотрим два случая.

1. Если $x_i \in \{z_1^i, z_2^i, \ldots, z_k^i\}$, то из U_i помимо $y_1^i, y_2^i, \ldots, y_j^i$ удаляется не более чем $k-1$ элементов, каждый из которых не превосходит x_i. Следовательно,

$$s_i \leqslant (k-1)x_i + \sum_{l=1}^{j} y_l^i \leqslant (k-1)z_1^i + \sum_{l=1}^{k} z_l^i \leqslant k\left(\sum_{l=1}^{k} z_l^i\right).$$

2. Если $x_i \notin \{z_1^i, z_2^i, \ldots, z_k^i\}$, то $z_l^i \geqslant x_i$ для каждого $l = 1, 2, \ldots, k$. Поскольку всего из U_i удалено не более чем $2k$ элементов, имеем

$$s_i \leqslant 2kx_i \leqslant 2\left(\sum_{l=1}^{k} z_l^i\right) \leqslant k\left(\sum_{l=1}^{k} z_l^i\right),$$

так как $k \geqslant 2$. Теорема 2 доказана.

Покажем неулучшаемость доказанной в теореме 2 константы. В первом примере показано, что существуют матрицы, для которых оптимальные решения классической и децентрализованной транспортных задач отличаются почти в k раз. Во втором примере показывается, что решение, найденное описанным выше алгоритмом, может быть почти в k раз хуже оптимального. Чтобы избежать громоздкости, оба примера приведены для случая $k = 2$, но их нетрудно обобщить и для произвольного k.

Пример 1. Рассмотрим матрицу (c_{ij}) с чётным числом строк n:

$$\begin{pmatrix} a & 2 & \dots & 2 & 1 & 0 & \dots & 0 \\ 2 & a & \dots & 2 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 2 & 2 & \dots & a & 0 & 0 & \dots & 1 \end{pmatrix}.$$

Нетрудно проверить, что при $a > 2$ имеем

$$S_C^* = n(a+1), \quad S_D^* = (a+2)n/2 + n/2.$$

Ясно, что при достаточно большом значении a, выполняется соотношение $S_C^*/S_D^* \approx 2$.

Пример 2. Пусть матрица $C = (C_a, I_n)$ имеет чётное число строк n, где

$$C_a = \begin{pmatrix} a+n & 0 & \dots & 0 & 2 & 0 & \dots & 0 \\ 0 & a+n-2 & \dots & 0 & 0 & 2 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \\ 0 & 0 & \dots & a+2 & 0 & 0 & \dots & 2 \\ 0 & 0 & \dots & 0 & a+n-1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & a+n-3 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & a+1 \end{pmatrix},$$

I_n — квадратная единичная матрица порядка n.

Оптимальным решением децентрализованной транспортной задачи с указанной матрицей является перестановка $\pi_1 = n(n-1)\dots 1$, так что

$$S_D^* = S_D(\pi_1) = \sum_{i=1}^{n}((a+i)+1) = na + \frac{n^2}{2} + 3\frac{n}{2}.$$

Алгоритм же находит перестановку $\pi_2 = 12\dots n$, для которой имеет место соотношение

$$S_D(\pi_2) = \sum_{i=1}^{\frac{n}{2}}(a+2i+2) + \frac{n}{2} = a\frac{n}{2} + \frac{n^2}{4} + 2n.$$

Таким образом, $S_D(\pi_1)/S_D(\pi_2) \approx 2$ при достаточно большом значении a, т. е. решение, найденное алгоритмом, отличается от оптимального почти в два раза.

В заключение отметим, что приведённый в статье алгоритм можно легко модифицировать на случай произвольных a_i. Тогда алгоритм найдёт решение, которое не более чем в k раз хуже оптимального, где $k = \max\{a_i \mid i = 1, 2, \ldots, n\}$. В случае же произвольных b_j необходимо сначала для каждого j скопировать j-й столбец b_j раз. Получится эквивалентная задача с единичными требованиями потребителей, однако для неё приведённый выше алгоритм будет псевдополиномиальным.

ЛИТЕРАТУРА

1. **Гэри М., Джонсон Д.** Вычислительные машины и трудноpешаемые задачи. — М.: Мир, 1982. — 416 с.
2. **Диниц Е. А.** Алгоритм поразрядного сокращения невязок и транспортные задачи // Исследования по дискретной математике. — М.: Наука, 1973. — С. 46–57.
3. **Кристофидес Н.** Теория графов. Алгоритмический подход. — М.: Мир, 1978. — 432 с.

Дементьев Владимир Тихонович,
e-mail: orlab@math.nsc.ru

Пяткин Артём Валерьевич,
e-mail: artem@math.nsc.ru

Статья поступила
10 октября 2007 г.

Переработанный вариант —
3 марта 2008 г.