

УДК 519.854.2

Задача целочисленного сбалансирования трехмерной матрицы и алгоритмы ее решения

Рублев В. С., Смирнов А. В.¹

Ярославский государственный университет им. П.Г.Демидова

e-mail: roublev@mail.ru, alexander_sm@mail.ru

получена 22 апреля 2010

Ключевые слова: целочисленное сбалансирование, трехмерные матрицы, кратные сети, кратные потоки, обобщенный алгоритм пометок, первый алгоритм Гомори, NP -полнота, минимизация ошибок округления

Рассматривается задача целочисленного сбалансирования трехмерной матрицы, предлагается сведение этой задачи к задаче нахождения максимального потока в кратной сети целочисленного сбалансирования, приводится алгоритм решения задачи о кратном потоке. Также проводится сравнительная характеристика алгоритмов целочисленного сбалансирования на основании вычислительных экспериментов. Кроме того, обосновывается NP -полнота задачи целочисленного сбалансирования трехмерной матрицы и рассматривается задача минимизации ошибок округления в задаче сбалансирования.

1. Введение

Различные задачи целочисленного сбалансирования возникают в сфере управления, экономики, финансов. В частности, подобная задача ставится при планировании железнодорожных грузоперевозок. Имеется матричный план по отправке вагонов, который группируется по некоторым показателям (например, направление, тип вагона, владелец вагона и т. п.). Данный план составляется на месяц и, естественно, является целочисленным. Однако вагоны необходимо отправлять ежедневно. При делении на количество дней в месяце план перестает быть целочисленным. Поэтому возникает проблема такого округления основных параметров, чтобы суммирующие показатели не выходили за определенные рамки. Данный план может быть представлен в виде k -мерной матрицы, где k – это число показателей, по которым ведется суммирование.

В работе [1] была рассмотрена задача целочисленного сбалансирования двумерной матрицы. Пусть имеется матрица A размерности $(n + 1) \times (m + 1)$ с вещественными элементами a_{ij} , для которой выполняются следующие условия баланса:

¹Работа выполнена при поддержке ФЦП «Научные и научно-педагогические кадры инновационной России» на 2009–2013 годы (государственный контракт № П161).

$$a_{00} = \sum_{i=1}^n \sum_{j=1}^m a_{ij}; \quad a_{i0} = \sum_{j=1}^m a_{ij} \quad (i \in \overline{1, n}); \quad a_{0j} = \sum_{i=1}^n a_{ij} \quad (j \in \overline{1, m}).$$

Ищется такая целочисленная матрица D , что выполняются условия:

$$d_{ij} \in \{[a_{ij}], \lceil a_{ij} \rceil\} \quad (i \in \overline{0, n}, j \in \overline{0, m}); \quad d_{00} = \lfloor a_{00} + 0.5 \rfloor;$$

$$d_{00} = \sum_{i=1}^n \sum_{j=1}^m d_{ij}; \quad d_{i0} = \sum_{j=1}^m d_{ij} \quad (i \in \overline{1, n}); \quad d_{0j} = \sum_{i=1}^n d_{ij} \quad (j \in \overline{1, m}).$$

Эта целочисленная матрица D называется сбалансированной для матрицы A . Без ограничения общности можно считать, что элементы исходной нецелочисленной матрицы A принадлежат полуинтервалу $[0, 1)$.

В [1] показано, что задача целочисленного сбалансирования двумерного матричного плана может быть сведена к задаче нахождения максимального потока в транспортной сети по следующим правилам:

1) Каждому элементу a_{ij} матрицы A поставим в соответствие вершину сети x_{ij} , а элементу a_{00} поставим в соответствие еще дополнительную вершину z_{00} .

2) Соединим вход x_0 сети с каждой вершиной x_{i0} ($i \in \overline{1, n}$) дугами пропускной способности $[a_{i0}]$, а также с вершиной x_{00} дугой пропускной способности $d_{00} - \sum_{i=1}^n [a_{i0}]$.

3) Соединим вершину x_{00} с каждой из вершин x_{i0} ($i \in \overline{1, n}$) дугами пропускной способности $\lceil a_{i0} \rceil - [a_{i0}]$.

4) Соединим со стоком z каждую из вершин x_{0j} ($j \in \overline{1, m}$) дугами пропускной способности $[a_{0j}]$, а вершину z_{00} – дугой пропускной способности $d_{00} - \sum_{j=1}^m [a_{0j}]$.

5) Соединим с вершиной z_{00} каждую из вершин x_{0j} ($j \in \overline{1, m}$) дугами пропускной способности $\lceil a_{0j} \rceil - [a_{0j}]$.

6) Соединим каждую из вершин x_{i0} ($i \in \overline{1, n}$) со всеми вершинами x_{ij} ($i \in \overline{1, n}, j \in \overline{1, m}$) дугами пропускной способности $\lceil a_{ij} \rceil$.

7) Соединим с каждой вершиной x_{0j} ($j \in \overline{1, m}$) все вершины x_{ij} ($i \in \overline{1, n}, j \in \overline{1, m}$) дугами пропускной способности $\lceil a_{ij} \rceil$.

Также в [1] показано, что максимальный поток в этой сети, найденный при помощи алгоритма Форда–Фалкерсона, будет индуцировать решение задачи целочисленного сбалансирования двумерной матрицы. Таким образом, задача целочисленного сбалансирования двумерной матрицы может быть решена за полиномиальное время.

2. Постановка задачи

Рассмотрим теперь задачу целочисленного сбалансирования трехмерной матрицы. Дана вещественная трехмерная матрица A размерности $(n+1) \times (m+1) \times (t+1)$ с элементами a_{ijp} , для которой выполнены условия баланса:

$$a_{000} = \sum_{i=1}^n \sum_{j=1}^m \sum_{p=1}^t a_{ijp}; \quad a_{i00} = \sum_{j=1}^m \sum_{p=1}^t a_{ijp} \quad (i \in \overline{1, n}); \quad a_{0j0} = \sum_{i=1}^n \sum_{p=1}^t a_{ijp} \quad (j \in \overline{1, m});$$

$$\begin{aligned}
a_{00p} &= \sum_{i=1}^n \sum_{j=1}^m a_{ijp} \quad (p \in \overline{1, t}); \quad a_{ij0} = \sum_{p=1}^t a_{ijp} \quad (i \in \overline{1, n}, j \in \overline{1, m}); \\
a_{i0p} &= \sum_{j=1}^m a_{ijp} \quad (i \in \overline{1, n}, p \in \overline{1, t}); \quad a_{0jp} = \sum_{i=1}^n a_{ijp} \quad (j \in \overline{1, m}, p \in \overline{1, t}).
\end{aligned}$$

Ищется целочисленная сбалансированная матрица D той же размерности, для которой выполнены условия:

$$\begin{aligned}
d_{ijp} &\in \{\lfloor a_{ijp} \rfloor, \lceil a_{ijp} \rceil\} \quad (i \in \overline{0, n}, j \in \overline{0, m}, p \in \overline{0, t}); \quad d_{000} = \lfloor a_{000} + 0.5 \rfloor; \\
d_{000} &= \sum_{i=1}^n \sum_{j=1}^m \sum_{p=1}^t d_{ijp}; \quad d_{i00} = \sum_{j=1}^m \sum_{p=1}^t d_{ijp} \quad (i \in \overline{1, n}); \quad d_{0j0} = \sum_{i=1}^n \sum_{p=1}^t d_{ijp} \quad (j \in \overline{1, m}); \\
d_{00p} &= \sum_{i=1}^n \sum_{j=1}^m d_{ijp} \quad (p \in \overline{1, t}); \quad d_{ij0} = \sum_{p=1}^t d_{ijp} \quad (i \in \overline{1, n}, j \in \overline{1, m}); \\
d_{i0p} &= \sum_{j=1}^m d_{ijp} \quad (i \in \overline{1, n}, p \in \overline{1, t}); \quad d_{0jp} = \sum_{i=1}^n d_{ijp} \quad (j \in \overline{1, m}, p \in \overline{1, t}).
\end{aligned}$$

Построение модели, аналогичной двумерному случаю, здесь уже невозможно. В работах [2, 3] рассмотрен вопрос о сводимости в смысле «естественного» сведения. Это такое сведение, при котором по исходной трехмерной матрице будет строиться сеть; при этом каждому элементу a_{ijp} матрицы будет соответствовать вершина сети, а пропускные способности дуг будут устанавливаться в зависимости от значений a_{ijp} , причем все пропускные способности дуг должны быть целочисленными. Потребуем также, чтобы некоторое подмножество компонент оптимального решения потоковой задачи образовывало оптимальное решение задачи сбалансирования и задача сбалансирования не имела допустимого решения, если не имеет допустимого решения потоковая задача.

Там же доказана теорема о том, что при $n \geq 2$, $m \geq 2$, $p \geq 2$ для задачи целочисленного сбалансирования трехмерной матрицы не существует «естественного» сведения к задаче о нахождении наибольшего потока. Тем не менее отсутствие сводимости задачи целочисленного сбалансирования трехмерной матрицы к задаче о наибольшем потоке в «естественном» смысле не означает отсутствия какого-либо другого сведения. Однако другое сведение очевидно будет сложнее, нежели «естественное», поэтому в работах [4, 5] было предложено обобщение теории потоков Форда–Фалкерсона, названное кратными потоками и задачей о нахождении максимального кратного потока. Далее мы подробно рассмотрим эту модель.

В разделе 7 мы докажем NP -полноту задачи целочисленного сбалансирования трехмерной матрицы. Помимо прочего, это повлечет за собой отсутствие эффективной сводимости к потоковой задаче.

3. Кратная сеть целочисленного сбалансирования

Для построения алгоритма решения задачи сбалансирования рассмотрим обобщение теории потоков Форда–Фалкерсона, которое назовем кратными потоками. Для этого введем целое $k > 1$, которое назовем кратностью потока. В качестве сети рассматривается ориентированный мультиграф $G(X, U)$, между вершинами которого могут быть дуги одного из 3 видов:

1) *обычная дуга* u^o с пропускной способностью $c(u^o)$, поток по которой не связан с потоком по другим дугам; множество обычных дуг обозначим через U^o ;

2) *кратная дуга* u^k между двумя вершинами, которая состоит из k дуг одной ориентации с одинаковой пропускной способностью $c(u^k)$ и одинаковым потоком по каждой из них; множество кратных дуг обозначим через U^k ;

3) *связанная дуга* u между двумя вершинами, которая связана с еще $k - 1$ дугой, имеющей одинаковый один из концов; множество связанных дуг, выходящих из одной вершины или входящих в одну вершину, будем называть *мультидугой* u^m ; пропускная способность всех связанных дуг одной мультидуги одинакова; поток по каждой связанной дуге из мультидуги одинаков; множество мультидуг обозначим через U^m .

Множество выходящих из вершины дуг может быть либо только *кратными* дугами, либо только одной *мультидугой* (k связанных дуг), либо только *обычными* дугами.

Из источника x_0 сети выходят только кратные дуги, а в сток z сети входит только одна мультидуга. Если из вершины выходят связанные дуги мультидуги, то в нее обязательно входит кратная дуга. Если в вершину входит мультидуга, то из нее может выходить только кратная дуга. Определенный таким образом мультиграф $G(X, U)$ с пропускными способностями дуг назовем *кратной (транспортной) сетью*.

Кратным потоком по сети называется целочисленная функция, определенная на множестве дуг $U = U^o \cup U^k \cup U^m$, для которой выполнены условия неотрицательности, ограниченности (пропускными способностями дуг) и неразрывности потока (в каждой вершине).

Величиной кратного потока называется сумма φ_z входящего потока для стока z , равная сумме выходящего из источника потока.

Отметим, что при $k = 1$ кратная сеть превращается в обычную транспортную сеть, а кратный поток превращается в обычный поток по этой сети. Задача о наибольшем кратном потоке для кратной сети является обобщением задачи о наибольшем потоке для обычной транспортной сети.

Покажем теперь, каким образом задача целочисленного сбалансирования трехмерной матрицы может быть сведена к задаче о наибольшем кратном потоке.

Заметим, что без ограничения общности в задаче о целочисленном сбалансировании можно считать, что $a_{ijp} < 1$ для всех ненулевых значений индексов i, j, p . В противном случае исходную матрицу A можно разложить на сумму матриц $B + F$, где B образована целыми частями элементов a_{ijp} ($i \in \overline{1, n}, j \in \overline{1, m}, p \in \overline{1, t}$) и значения элементов b_{ijp} с нулевыми индексами находятся из условий баланса, а $F = B - A$. Если H – целочисленная сбалансированная матрица для F , то $D = B + H$

– целочисленная сбалансированная матрица для A . Это замечание позволяет считать, что $d_{ijp} \in \{0, 1\}$ ($i \in \overline{1, n}$, $j \in \overline{1, m}$, $p \in \overline{1, t}$).

Кратная сеть $G(X, U)$ целочисленного сбалансирования представлена на рис. 1 на следующей странице. Каждому элементу a_{ijp} матрицы A соответствует вершина сети x_{ijp} (на рисунке показаны только вершины со значениями индексов 0, 1 и n, m, t соответственно для индексов i, j, p). Кратные дуги отмечены жирными стрелками (для наглядности кратные дуги вместе с вершинами, которые они соединяют, показаны на рисунке дважды – в частях G_1 и G_2 сети G). Связанные дуги мультидуг отмечены двойными стрелками и при этом вторая стрелка стоит у вершины, которая является общим концом связанных дуг мультидуги (в сток z входит мультидуга, образованная связанными дугами из z_1 и z_2 , а из каждой вершины x_{ijp} ($i > 0$, $j > 0$, $p > 0$) выходит мультидуга, образованная связанными дугами, одна из которых находится в части G_1 , а другая – в части G_2). Каждой вершине x_{ijp} , где более чем 1 индекс имеет нулевые значения, соответствует дополнительная вершина x'_{ijp} , а также вершинам z_k ($k = 1, 2$) соответствуют дополнительные вершины z'_k . Пропускные способности кратных дуг:

$$c(x_{000}, x_{i00}) = 2[a_{i00}] \quad (i \in \overline{1, n}); \quad c(x_{000}, x'_{000}) = 2 \left([a_{000} + 0.5] - \sum_{i=1}^n [a_{i00}] \right);$$

$$c(x'_{000}, x_{i00}) = 2([a_{i00}] - [a_{i00}]) \quad (i \in \overline{1, n}); \quad c(x_{i00}, x_{ij0}) = 2[a_{ij0}] \quad (i \in \overline{1, n}, j \in \overline{1, m});$$

$$c(x_{i00}, x'_{i00}) = 2 \left([a_{i00}] - \sum_{j=1}^m [a_{ij0}] \right) \quad (i \in \overline{1, n});$$

$$c(x'_{i00}, x_{ij0}) = 2([a_{ij0}] - [a_{ij0}]) \quad (i \in \overline{1, n}, j \in \overline{1, m});$$

$$c(x_{ij0}, x_{ijp}) = 2[a_{ijp}] \quad (i \in \overline{1, n}, j \in \overline{1, m}, p \in \overline{1, t}).$$

Пропускные способности связанных дуг для всех мультидуг из вершин x_{ijp} равны $[a_{ijp}]$, а пропускные способности связанных дуг, входящих в вершину z : $c(z_k, z) = [a_{000} + 0.5]$ ($k = 1, 2$).

Пропускные способности обычных дуг:

$$c(x_{0jp}, x_{0j0}) = [a_{0jp}] \quad (j \in \overline{1, m}, p \in \overline{1, t}); \quad c(x_{0jp}, x'_{0j0}) = [a_{0jp}] - [a_{0jp}] \quad (j \in \overline{1, m}, p \in \overline{1, t});$$

$$c(x'_{0j0}, x_{0j0}) = [a_{0j0}] - \sum_{p=1}^t [a_{0jp}] \quad (j \in \overline{1, m}); \quad c(x_{0j0}, z_1) = [a_{0j0}] \quad (j \in \overline{1, m});$$

$$c(x_{0j0}, z'_1) = [a_{0j0}] - [a_{0j0}] \quad (j \in \overline{1, m}); \quad c(z'_1, z_1) = [a_{000} + 0.5] - \sum_{j=1}^m [a_{0j0}];$$

$$c(x_{i0p}, x_{00p}) = [a_{i0p}] \quad (i \in \overline{1, n}, p \in \overline{1, t}); \quad c(x_{i0p}, x'_{00p}) = [a_{i0p}] - [a_{i0p}] \quad (i \in \overline{1, n}, p \in \overline{1, t});$$

$$c(x'_{00p}, x_{00p}) = [a_{00p}] - \sum_{i=1}^n [a_{i0p}] \quad (p \in \overline{1, t}); \quad c(x_{00p}, z_2) = [a_{00p}] \quad (p \in \overline{1, t});$$

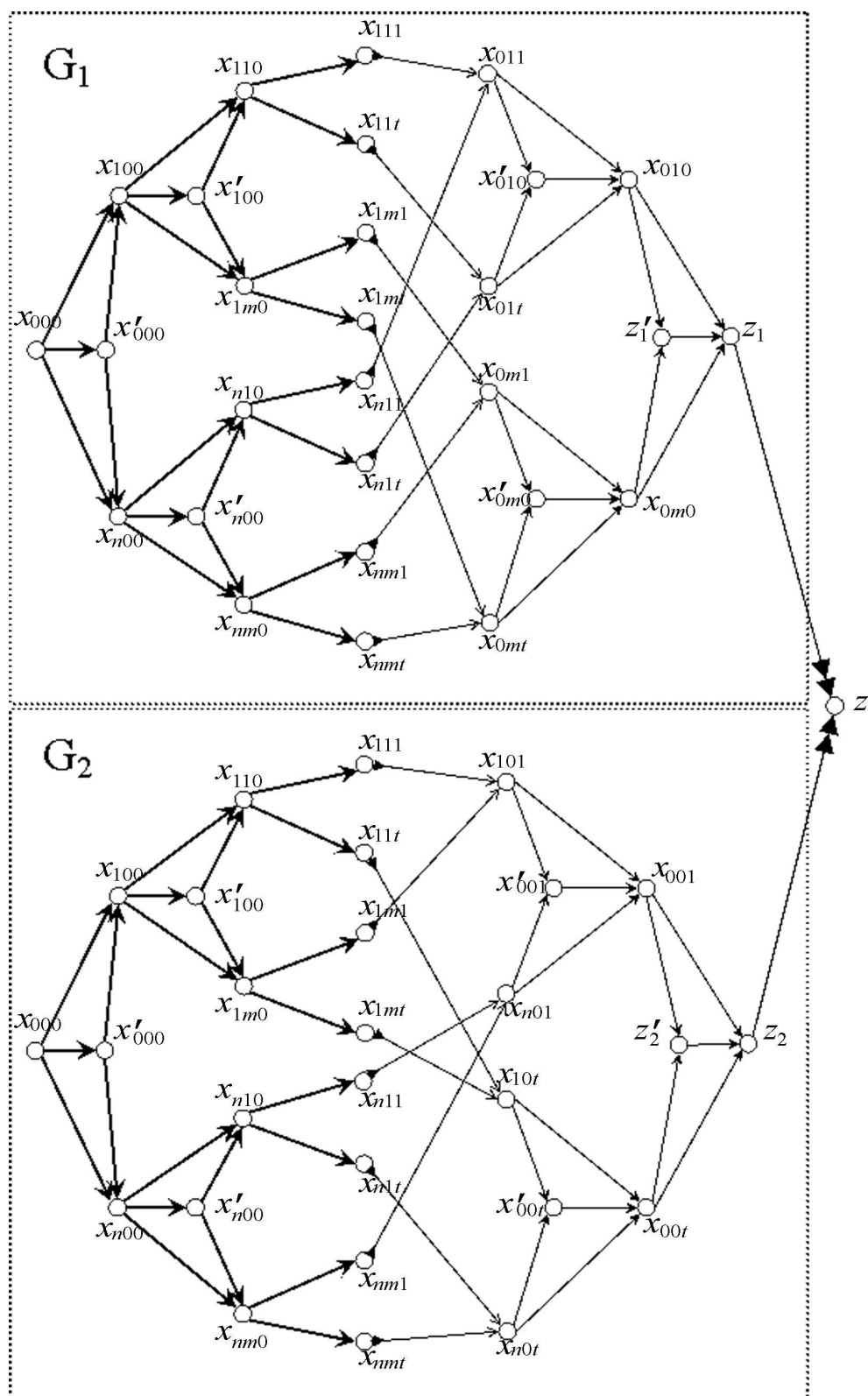


Рис. 1. Кратная сеть целочисленного сбалансирования трехмерной матрицы

$$c(x_{00p}, z'_2) = \lceil a_{00p} \rceil - \lfloor a_{00p} \rfloor \quad (p \in \overline{1, t}); \quad c(z'_2, z_2) = \lfloor a_{000} + 0.5 \rfloor - \sum_{p=1}^t \lfloor a_{00p} \rfloor.$$

Связь задачи целочисленного сбалансирования и задачи о максимальном кратном потоке устанавливается следующей теоремой.

Теорема 1. Если задача целочисленного сбалансирования трехмерной матрицы имеет решение, то оно индуцирует максимальный поток в кратной сети G следующим образом:

- 1) для каждой вершины x_{ijp} , являющейся концом кратной дуги, проходящий через нее поток равен $2d_{ijp}$;
- 2) для каждой вершины x_{ijp} , являющейся концом обычной дуги, проходящий через нее поток равен d_{ijp} ;
- 3) для вершины x'_{000} проходящий через нее поток равен $c(x_{000}, x'_{000})$;
- 4) для каждой вершины z_k ($k = 1, 2$) проходящий через нее поток равен $\lfloor a_{000} + 0.5 \rfloor$;
- 5) для каждой вершины z'_k ($k = 1, 2$) проходящий через нее поток равен $c(z'_k, z_k)$.

Справедливость теоремы 1 следует непосредственно из правил построения кратной сети целочисленного сбалансирования. Однако обратное утверждение неверно: не всякому максимальному кратному потоку величины $2\lfloor a_{000} + 0.5 \rfloor$ соответствует решение задачи целочисленного сбалансирования матрицы. Поэтому дополнительно потребуем выполнения следующих условий, которые в дальнейшем мы будем называть *условиями разрешимости*:

$$\begin{aligned} f(x_{i00}, x_{ij0}) + f(x'_{i00}, x_{ij0}) &\geq c(x_{i00}, x_{ij0}) \quad (i \in \overline{1, n}, j \in \overline{1, m}); \\ f(x_{0jp}, x_{0j0}) + f(x_{0jp}, x'_{0j0}) &\geq c(x_{0jp}, x_{0j0}) \quad (j \in \overline{1, m}, p \in \overline{1, t}); \\ f(x_{i0p}, x_{00p}) + f(x_{i0p}, x'_{00p}) &\geq c(x_{i0p}, x_{00p}) \quad (i \in \overline{1, n}, p \in \overline{1, t}). \end{aligned} \quad (1)$$

Здесь $f(a, b)$ – это поток на соответствующей дуге. В дальнейшем дуги (x_{i00}, x_{ij0}) , (x_{0jp}, x_{0j0}) , (x_{i0p}, x_{00p}) ($i \in \overline{1, n}, j \in \overline{1, m}, p \in \overline{1, t}$) мы будем называть *основными*, а дуги (x_{i00}, x'_{i00}) , (x'_{i00}, x_{ij0}) , (x_{0jp}, x'_{0j0}) , (x'_{0j0}, x_{0j0}) , (x_{i0p}, x'_{00p}) , (x'_{00p}, x_{00p}) ($i \in \overline{1, n}, j \in \overline{1, m}, p \in \overline{1, t}$) мы будем называть *дополнительными*.

Справедливо следующее утверждение.

Теорема 2. Пусть максимальный кратный поток φ для сети G имеет величину $\varphi_z = 2\lfloor a_{000} + 0.5 \rfloor$ и выполнены условия разрешимости (1). Тогда данный поток индуцирует решение задачи целочисленного сбалансирования трехмерной матрицы следующим образом: положим d_{ijp} равным

- 1) половине величины потока, проходящего через вершину x_{ijp} , если она является концом кратной дуги;
- 2) величине потока, проходящего через вершину x_{ijp} , если она является концом обычной дуги.

Доказательство. Заметим, что в рассматриваемой ситуации выполнение условия $d_{ijp} \in \{\lfloor a_{ijp} \rfloor, \lceil a_{ijp} \rceil\}$ для элементов d_{ijp} , d_{i00} , d_{0j0} , d_{00p} , d_{000} ($i > 0, j > 0, p > 0$) следует непосредственно из правил построения сети. Выполнение же этого условия для элементов d_{ij0} , d_{0jp} , d_{i0p} ($i > 0, j > 0, p > 0$) будет обеспечено за счет условий разрешимости.

Сохранение условий баланса следует из условия неразрывности кратного потока в каждой вершине.

Таким образом, кратный поток, для которого $\varphi_z = 2[a_{000} + 0.5]$ и выполнены условия разрешимости (1), индуцирует решение задачи сбалансирования, что и требовалось доказать.

Объединяя вышесказанное, получаем следующий критерий разрешимости задачи сбалансирования.

Теорема 3. *Для того, чтобы задача целочисленного сбалансирования имела решение, необходимо и достаточно, чтобы в сети целочисленного сбалансирования существовал максимальный кратный поток φ величины $\varphi_z = 2[a_{000} + 0.5]$, для которого выполняются условия разрешимости (1).*

Необходимость здесь устанавливает теорема 1, достаточность – теорема 2.

Теорема 3 устанавливает сведение задачи целочисленного сбалансирования к задаче о максимальном кратном потоке.

4. Полный поток и обобщенный путь прорыва

Объединение мультидуги u_z^m , идущей из вершин z_1, z_2 в z и двух путей μ_r ($r = 1, 2$), каждый из которых является ориентированным путем в соответствующей части G_r из вершины x_{000} в вершину z_r , назовем *обобщенным путем*, если каждый из путей μ_r проходит через одну и ту же вершину x_{ijp} с ненулевыми индексами $i > 0, j > 0, p > 0$.

Кратный поток назовем *полным*, если любой обобщенный путь из x_{000} в z имеет дугу u (обычную, кратную или мультидугу), поток по которой равен ее пропускной способности $f(u) = c(u)$.

Проекция C_i ($i = 1, 2$) подграфа C на часть сети G_i – это часть подграфа C , образованная его вершинами и дугами, принадлежащими G_i .

Так как части G_i ($i = 1, 2$) сети G представляют собой обычные транспортные сети с источником x_{000} и стоком z_i , то будем называть некоторый путь из x_{000} в z_i *путем прорыва* в части G_i , если $f(u) < c(u)$ на прямых дугах и $f(u) > 0$ на обратных дугах этого пути.

Разрезом R в кратной сети $G(X, U)$, отделяющим x_{000} и z , назовем множество вершин R , где $x_{000} \in R$, а $z \in \bar{R}$ ($\bar{R} = X \setminus R$). Число $c(R) = \sum_{x \in R, y \in \bar{R}} c(x, y)$ назовем *пропускной способностью разреза* R . Разрез $R_0 = \arg \min_R c(R)$ назовем *минимальным разрезом*.

Пусть в кратной сети определен некоторый поток φ величины $\varphi_z \geq 0$. *Кратным циклом* в сети $G(X, U)$, где X – множество вершин, U – множество дуг (обычных, кратных или мультидуг), назовем такой подграф $C(X', U')$, $X' \subset X$, $U' \subset U$, для которого:

1) проекции C_1 и C_2 на части G_1 и G_2 соответственно есть объединение некоторых циклов, причем дуги, поток по которым ненулевой, могут проходиться в обратном направлении;

2) проекции C_1 и C_2 согласованы (одинаковы) на общей части подграфов G_1 и G_2 ;

3) C_1 представим в виде $C_1 = \cup \{C_1^j\}$, где C_1^j – некоторые циклы и $C_1^j \not\subset C_1^k \forall j \neq k$; при этом для любой дуги u из G_1 выполняется неравенство

$$0 \leq f(u) + a^+(u) - a^-(u) \leq c(u),$$

где $a^+(u)$ – это число циклов C_1^j , в которых дуга u проходится в прямом направлении, а $a^-(u)$ – это число циклов C_1^j , в которых дуга u проходится в обратном направлении. Такое же условие должно выполняться и для C_2 .

Обобщенным путем прорыва в сети $G(X, U)$ для некоторого кратного потока φ назовем такой подграф $S(X', U')$, $X' \subset X$, $U' \subset U$, для которого:

1) каждая из проекций S_1 и S_2 на части G_1 и G_2 соответственно есть объединение ровно одного пути прорыва из x_{000} в z и некоторых циклов, причем дуги, поток по которым ненулевой, могут проходиться в обратном направлении;

2) проекции S_1 и S_2 согласованы (одинаковы) на общей части подграфов G_1 и G_2 ;

3) S_1 представим в виде $S_1 = \mu_1 \cup \{C_1^j\}$, где μ_1 – путь прорыва, C_1^j – некоторые циклы и $C_1^j \not\subset C_1^k \forall j \neq k$; при этом для любой дуги u из G_1 выполняется неравенство

$$0 \leq f(u) + a^+(u) - a^-(u) \leq c(u),$$

где $a^+(u)$ – это число элементов множества $\mu_1 \cup \{C_1^j\}$, в которых дуга u проходится в прямом направлении, а $a^-(u)$ – это число элементов множества $\mu_1 \cup \{C_1^j\}$, в которых дуга u проходится в обратном направлении. Такое же условие должно выполняться и для S_2 ;

4) S не содержит кратного цикла.

Рассмотрим теперь обычную сеть $G(X, U)$; X – множество вершин, U – множество дуг. $\varphi(U)$ – поток в сети $G(X, U)$, φ_z – его величина. $c(u)$ – пропускная способность дуги u ; $c(u) \geq f(u) \geq 0$ – поток на дуге. $f_x(u)$ – поток, входящий или исходящий из вершины по дуге u . Пусть x_0 – источник, z – сток сети $G(X, U)$.

$G_\varphi = G(X, U_\varphi)$; $U_\varphi = \{u | f(u) \neq 0\}$.

Лемма 1. Пусть $\varphi^1(U)$, $\varphi^2(U)$ – потоки в сети $G(X, U)$, причем $\varphi_z^1 < \varphi_z^2$. Пусть $\varphi^0(U) = \varphi^2(U) - \varphi^1(U)$. Тогда граф $G_{\varphi^0} = \{S_i\} \cup \{C_j\}$, где $\{S_i\}$ – множество всех путей прорыва из x_0 в z для потока φ^1 , $\{S_i\} \neq \emptyset$, а $\{C_j\}$ – множество всех циклов.

Доказательство. I. Очевидно, что $G_{\varphi^0} = G' \cup G_1'' \cup \dots \cup G_k''$, где G' , G_1'' , ..., G_k'' – компоненты связности; $x_0 \in G'$.

Докажем, что $\forall x \in G_{\varphi^0}$, $x \neq x_0$, $x \neq z$ выполнено

$$\sum f_x^0(u) = 0. \quad (2)$$

Суммирование ведется по всем дугам u , инцидентным вершине x .

Рассмотрим множества дуг

$$U_{in}^1(x) = \{u \in G_{\varphi^1} | u \text{ входит в } x\}; \quad U_{in}^2(x) = \{u \in G_{\varphi^2} | u \text{ входит в } x\};$$

$$U_{out}^1(x) = \{u \in G_{\varphi^1} | u \text{ исходит из } x\}; \quad U_{out}^2(x) = \{u \in G_{\varphi^2} | u \text{ исходит из } x\}.$$

Дополним эти множества дугами u_i с нулевыми потоками на них так, чтобы $U_{in}^1(x) = U_{in}^2(x)$, $U_{out}^1(x) = U_{out}^2(x)$.

$$x \neq x_0, x \neq z \Rightarrow \begin{cases} \sum_{u \in U_{in}^1(x)} f_x^1(u) = - \sum_{u \in U_{out}^1(x)} f_x^1(u) \\ \sum_{u \in U_{in}^2(x)} f_x^2(u) = - \sum_{u \in U_{out}^2(x)} f_x^2(u) \end{cases} \quad (3)$$

Обозначим $U_{in}^0(x) = U_{in}^1(x) = U_{in}^2(x)$, $U_{out}^0(x) = U_{out}^1(x) = U_{out}^2(x)$. Тогда

$$\sum_{u \in U_{in}^0(x)} f_x^0(u) = \sum_{u \in U_{in}^0(x)} (f_x^2(u) - f_x^1(u)) \stackrel{(3)}{=} \sum_{u \in U_{out}^0(x)} (f_x^1(u) - f_x^2(u)) = - \sum_{u \in U_{out}^0(x)} f_x^0(u),$$

откуда следует требуемое равенство.

II. Докажем, что $z \in G'$. G' – сеть, поэтому

$$\sum_{x \in G'} \sum_{u \in U_{in}^0(x) \cup U_{out}^0(x)} f_x^0(u) = 0.$$

Предположим, что $z \notin G'$. Т.к. $\forall x \in G_{\varphi^0}$, $x \neq x_0$, $x \neq z$ $\sum_{u \in U_{in}^0(x) \cup U_{out}^0(x)} f_x^0(u) = 0$ (по (2)), то

$$\sum_{x \in G'} \sum_{u \in U_{in}^0(x) \cup U_{out}^0(x)} f_x^0(u) = \sum_{u \in U_{in}^0(x_0) \cup U_{out}^0(x_0)} f_{x_0}^0(u) + \sum_{x \in G', x \neq x_0} \sum_{u \in U_{in}^0(x) \cup U_{out}^0(x)} f_x^0(u) = \varphi_z^0 > 0,$$

что приводит к противоречию. Таким образом, $z \in G'$.

III. $x_0 \in G'$, $z \in G'$ (по II), G' – связный граф, любой вершине $x \in G_{\varphi^0}$: $x \neq x_0$, $x \neq z$ инцидентны как входящие, так и исходящие дуги (следует из (2)), поэтому вершина z достижима из x_0 в G' , следовательно, сеть G_{φ^0} содержит хотя бы один путь прорыва.

IV. Докажем теперь, что $G_{\varphi^0} = \{S_i\} \cup \{C_j\}$.

Возьмем какой-то путь прорыва $S \in \{S_i\}$ с максимально возможным потоком на нем φ^{max} . После этого удалим S из G_{φ^0} , уменьшив поток на соответствующих дугах на величину φ^{max} .

Процедуру будем повторять, пока существуют пути прорыва из x_0 в z .

В итоге G' распадется на l компонент связности $G_1''', \dots, G_l'''$, а также останутся $G_1'', \dots, G_k''$. Для каждой вершины из этих $k + l$ компонент связности будет выполняться равенство (2) (в том числе и для вершин x_0 и z), следовательно, каждая компонента связности будет представлять собой объединение некоторых циклов. Поэтому $G_{\varphi^0} = \{S_i\} \cup \{C_j\}$.

Утверждение леммы полностью обосновано.

Лемма 2. Пусть $\varphi^1(U)$, $\varphi^2(U)$ – потоки в сети $G(X, U)$, причем $\varphi_z^1 = \varphi_z^2$. Пусть найдутся такие дуги сети $G(X, U)$, что потоки $\varphi^1(U)$, $\varphi^2(U)$ на этих дугах различны. Пусть $\varphi^0(U) = \varphi^2(U) - \varphi^1(U)$. Тогда граф $G_{\varphi^0} = \{C_j\}$, где $\{C_j\}$ – множество всех циклов.

Доказательство. Заметим сначала, что $G_{\varphi^0} = G_1 \cup \dots \cup G_k$, где $G_1, \dots, G_k$ – компоненты связности.

Повторяя рассуждения из части I доказательства леммы 1, получаем, что $\forall x \in G_{\varphi^0}, x \neq x_0, x \neq z$ выполнено $\sum f_x^0(u) = 0$. Суммирование ведется по всем дугам u , инцидентным вершине x .

При этом обязательно $\varphi_{x_0}^0 = \varphi_z^0 = 0$ (так как $\varphi_z^1 = \varphi_z^2$).

Повторяя далее рассуждения из части IV доказательства леммы 1, получаем, что $G_{\varphi^0} = \{C_j\}$, что и требовалось доказать.

Докажем теперь теорему о разности двух полных потоков в кратной сети. Так же, как и в случае обычной сети, обозначим $G_\varphi = G(X, U_\varphi)$; $U_\varphi = \{u | f(u) \neq 0\}$.

Теорема 4. Пусть $\varphi^1(U), \varphi^2(U)$ – два полных потока в кратной сети $G(X, U)$, причем $\varphi_z^1 \leq \varphi_z^2$. Пусть $\varphi^0(U) = \varphi^2(U) - \varphi^1(U)$. Тогда граф $G_{\varphi^0} = \{S_i\} \cup \{C_j\}$, где $\{S_i\}$ – множество всех обобщенных путей прорыва из x_{000} в z , а $\{C_j\}$ – множество всех кратных циклов. В случае, когда $\varphi_z^1 = \varphi_z^2 = \varphi_z^{max}$, $\{S_i\} = \emptyset$.

Доказательство. Обозначим через G_{φ^1} и G_{φ^2} проекции сети G_φ в частях G_1 и G_2 соответственно. Тогда по леммам 1 и 2

$$\begin{aligned} G_{\varphi^0}^1 &= \{S_{i_1}\} \cup \{C_{j_1}\}, \{S_{i_1}\} - \text{множество путей прорыва из } x_{000} \text{ в } z_1, \\ &\quad \{C_{j_1}\} - \text{множество циклов;} \\ G_{\varphi^0}^2 &= \{S_{i_2}\} \cup \{C_{j_2}\}, \{S_{i_2}\} - \text{множество путей прорыва из } x_{000} \text{ в } z_2, \\ &\quad \{C_{j_2}\} - \text{множество циклов.} \end{aligned}$$

Кратные сети G_{φ^1} и G_{φ^2} можно представить как совокупности обобщенных путей прорыва. Поэтому очевидно, что $G_{\varphi^1}^1$ и $G_{\varphi^2}^1$ совпадают на общей части так же, как и $G_{\varphi^1}^2$ и $G_{\varphi^2}^2$, и потоки на каждой связанной дуге каждой мультидуги равны в каждом из графов G_{φ^1} и G_{φ^2} . Следовательно, графы $G_{\varphi^0}^1$ и $G_{\varphi^0}^2$ совпадают на общей части и потоки на каждой связанной дуге каждой мультидуги графа G_{φ^0} равны. Таким образом, все пути прорыва в графе G_{φ^0} являются обобщенными, а циклы совпадают на общих частях графов $G_{\varphi^0}^1$ и $G_{\varphi^0}^2$, то есть $G_{\varphi^0} = \{S_i\} \cup \{C_j\}$.

Если $\varphi_z^1 = \varphi_z^2 = \varphi_z^{max}$, то дуги, принадлежащие минимальному разрезу, являются насыщенными в графах G_{φ^1} и G_{φ^2} , а следовательно, в графе G_{φ^0} поток на этих дугах нулевой, то есть $\{S_i\} = \emptyset$.

Утверждение теоремы полностью обосновано.

5. Алгоритм нахождения максимального кратного потока для решения задачи сбалансирования

Рассмотрим теперь алгоритм нахождения наибольшего потока в кратной сети. Работа алгоритма разбивается на три больших этапа:

- 1) этап построения полного потока;
- 2) этап увеличения потока. Если полученный полный поток не является максимальным, то увеличиваем поток при помощи обобщенного алгоритма пометок (см. ниже) до тех пор, пока поток не станет максимальным;

3) *этап коррекции потока*. Если полученный максимальный поток имеет величину $2[a_{000} + 0.5]$, но не удовлетворяет условиям разрешимости, то выполняем коррекцию потока при помощи обобщенного алгоритма пометок до тех пор, пока не будут выполняться условия разрешимости (1), либо не будет показано, что подобная коррекция невозможна.

Остановимся теперь подробнее на каждом из этапов.

Алгоритм нахождения полного потока прост. На данном этапе увеличения потока можно добиваться, находя все возможные обобщенные пути прорыва из x_{000} в z через вершины x_{ijp} ($i \in \overline{1, n}$, $j \in \overline{1, m}$, $p \in \overline{1, t}$) без обратных дуг и увеличивая поток по ним. В итоге будет получен полный поток.

Рассмотрим обобщенный алгоритм пометок. Идея алгоритма состоит в следующем: в проекциях G_1 и G_2 поочередно строятся пути прорыва μ_1 и μ_2 (возможно, в объединении с некоторыми циклами) до тех пор, пока пути в обеих проекциях не станут согласованными, либо же не останется вариантов для продолжения построения пути. В первом случае объединение μ_1 и μ_2 с мультидугой с концом в z даст обобщенный путь прорыва, во втором случае производится откат до «точки ветвления», после чего выполнение алгоритма возобновляется. Если точкой ветвления оказывается x_{000} и при этом x_{000} не помечена, то задача целочисленного сбалансирования не имеет решения.

Дадим теперь формальное описание обобщенного алгоритма пометок в случае кратной сети целочисленного сбалансирования:

1. С помощью алгоритма Форда–Фалкерсона находим путь прорыва μ_1 в части G_1 .
2. Части пути μ_1 , соответствующие кратным дугам, добавляем к пути μ_2 в части G_2 . Все пометки вершин x_{ij0} , x'_{i00} ($i \in \overline{0, n}$, $j \in \overline{0, m}$) в части G_2 заменяются соответствующими пометками из части G_1 . Добавляем к пути μ_2 также дуги (x_{ijp}, x_{i0p}) или (x_{i0p}, x_{ijp}) , инцидентные тем x_{ijp} ($i \in \overline{1, n}$, $j \in \overline{1, m}$, $p \in \overline{1, t}$), которые были добавлены к пути μ_2 , ставим соответствующие пометки (в случае добавления дуги (x_{ijp}, x_{i0p}) вершина x_{i0p} получает пометку $\{+x_{ijp}\}$, в случае добавления дуги (x_{i0p}, x_{ijp}) вершина x_{ijp} получает пометку $\{-x_{i0p}\}$).

Вершины x_{i0p} , добавленные на этом шаге к пути μ_2 , образуют два множества: $X^+ = \{x_{i0p} | (x_{ijp}, x_{i0p}) \in \mu_2\}$ и $X^- = \{x_{i0p} | (x_{i0p}, x_{ijp}) \in \mu_2\}$. Каждое вхождение вершины в одно из множеств считаем уникальным. Тогда $|X^+| = |X^-| + 1$.

3. В части G_2 будем строить поток из X^+ в $X^- \cup \{z_2\}$. Для этого мы будем поочередно брать вершины из множества X^+ и строить путь из выбранной вершины до тех пор, пока не будет достигнута какая-либо вершина из множества $X^- \cup \{z_2\}$, либо же это не окажется невозможным (в таком случае переходим на шаг 5). После построения очередного пути вершины, являющиеся его началом и концом, удаляются из множеств X^+ и $X^- \cup \{z_2\}$.

На шаге 3 в путь могут включаться вершины, уже помеченные при построении предыдущих путей, кроме вершин x_{ijp} ($i \in \overline{1, n}$, $j \in \overline{1, m}$, $p \in \overline{1, t}$). Таким образом, вершины могут иметь более одной пометки. При построении очередного

пути дуги ранее полученных путей могут проходиться в обратном направлении, соответствующие пометки взаимно уничтожаются. Также при построении очередного пути разрешается проходить дуги в том направлении, в котором они были пройдены в уже построенных путях, если не будет нарушено условие

$$0 \leq f(a, b) + a^+(a, b) - a^-(a, b) \leq c(a, b),$$

где (a, b) – дуга, $a^+(a, b)$ – количество пометок $\{+a\}$ у вершины b , $a^-(a, b)$ – количество пометок $\{-b\}$ у вершины a .

После получения какого-либо пути на шагах 1–3 все вершины этого пути, от которых могут быть помечены вершины, не принадлежащие этому пути, помечаются как *точки ветвления*, а все вершины, которые могут быть помечены от нее, включаются в *список вариантов* данной вершины. Если вершина, помечаемая как точка ветвления, уже была таковой в одном из ранее построенных путей, то с ней связывается новый список вариантов. Таким образом, с вершиной может быть связано несколько списков вариантов.

4. Повторяем шаги 2–3 поочередно для частей G_1 и G_2 (строим теперь потоки из X^+ в X^-), пока части G_1 и G_2 не станут согласованными, либо же не станет невозможным построение какого-либо пути. В первом случае переходим на шаг 6, во втором случае переходим на шаг 5.
5. Выполняем откат до ближайшей точки ветвления. После этого переходим к построению пути на шаге 1 или на шаге 3 в зависимости от ситуации. При этом запрещается добавлять в данный путь дуги, соединяющие точку ветвления с вершинами, не входящими в список вариантов. В случае, если рассматриваемый путь удастся построить, из списка вариантов выбранной точки ветвления удаляется вершина, получившая в этом пути пометку от точки ветвления. В противном случае текущий список вариантов данной вершины удаляется и выполняется откат до следующей точки ветвления.

Дополнительное правило для задачи сбалансирования. Если во время отката была достигнута вершина x_{000} и вершина x_{000} не имеет пометок, то задача целочисленного сбалансирования не имеет решения, выполнение алгоритма прекращаем. Если при этом список вариантов x_{000} пуст, то полученный кратный поток является максимальным.

6. Объединение μ_1 , μ_2 и мультидуги с концом в z образует обобщенный путь прорыва. Изменяем поток на каждой дуге (a, b) , входящей в этот обобщенный путь прорыва, на величину $a^+(a, b) - a^-(a, b)$, где $a^+(a, b)$ – количество пометок $\{+a\}$ у вершины b , $a^-(a, b)$ – количество пометок $\{-b\}$ у вершины a .

Рассмотрим теперь алгоритм этапа коррекции:

1. Во всех случаях, когда это возможно, переносим поток с дополнительных дуг на основные.

2. Если все основные дуги являются *насыщенными* (т.е. $f(u) = c(u)$), то выполнение алгоритма прекращаем. Иначе временно считаем нулевой пропускную способность основных дуг, являющихся насыщенными.
3. Возьмем произвольную ненасыщенную основную дугу и будем искать кратный цикл C , содержащий эту дугу. Для этого используем обобщенный алгоритм пометок без дополнительного правила для задачи сбалансирования, считая источником конечную вершину выбранной дуги, а стоком считая ее начальную вершину. Если рассматриваемая дуга имеет вид (x_{i0p}, x_{00p}) , то алгоритм начинаем с построения пути в части G_2 .

В итоге будет получен обобщенный путь прорыва из конца выбранной дуги в ее начало, либо же будет установлено, что такого пути нет (в этом случае задача целочисленного сбалансирования трехмерной матрицы не имеет решения; выполнение алгоритма прекращаем).

4. Объединение построенного обобщенного пути прорыва и выбранной дуги образует искомый кратный цикл C . Изменяем поток на дугах этого цикла в соответствии с шагом 6 обобщенного алгоритма пометок. Затем восстанавливаем пропускную способность всех дуг и переходим на шаг 1.

6. Обоснование алгоритма

Теорема 5. *Если задача целочисленного сбалансирования трехмерной матрицы имеет решение, то оно может быть найдено при помощи алгоритма нахождения максимального кратного потока для решения задачи сбалансирования.*

Доказательство. По теореме 3 задача целочисленного сбалансирования трехмерной матрицы имеет решение тогда и только тогда, когда в кратной сети целочисленного сбалансирования существует поток величины $2d_{000}$, удовлетворяющий условиям разрешимости.

Отметим также, что максимальный поток в сети целочисленного сбалансирования не может превышать $2d_{000}$, так как существует разрез $R = \{x_{000}\}$, пропускная способность которого $c(R) = 2d_{000}$.

Наконец, заметим, что величина любого потока в сети целочисленного сбалансирования должна быть кратна 2, так как данная сеть имеет кратность $k = 2$.

Поэтому для доказательства утверждения теоремы мы сначала покажем, что на этапах 1–2 алгоритма нахождения максимального кратного потока может быть построен поток величины $2d_{000}$, либо же может быть установлено, что такого потока нет. Затем мы покажем, что 3 этап позволяет скорректировать, если это возможно, поток величины $2d_{000}$ таким образом, чтобы выполнялись условия разрешимости (1).

I. На первом этапе алгоритма, очевидно, будет получен полный поток величины $2K$. Если $2K < 2d_{000}$, то нам необходимо выполнить обобщенный алгоритм пометок не более, чем $d_{000} - K$ раз. После любого применения обобщенного алгоритма пометок поток может уже не быть полным. Поэтому рассмотрим две ситуации:

1) Обобщенный алгоритм пометок был применен l раз ($l \in \overline{1, d_{000} - K - 1}$). Поток величины $2K + 2l$ не является полным.

В этом случае $\exists x_{i_0 j_0 p_0}$ ($i_0 \in \overline{1, n}$, $j_0 \in \overline{1, m}$, $p_0 \in \overline{1, t}$): существуют пути прорыва μ_1 и μ_2 в частях G_1 и G_2 , проходящие через $x_{i_0 j_0 p_0}$, согласованные на общей части (т.к. все изменения на предыдущих шагах алгоритма производились согласованно в частях G_1 и G_2) и не содержащие обратных дуг.

Объединение μ_1 , μ_2 и мультидуги с концом в z даст обобщенный путь прорыва, который и будет найден обобщенным алгоритмом пометок (т.к. он самый короткий из всех возможных).

2) Обобщенный алгоритм пометок был применен l раз ($l \in \overline{0, d_{000} - K - 1}$). Поток величины $2K + 2l$ является полным. Рассмотрим случаи:

а) Поток является максимальным. В этом случае алгоритм переберет все варианты и установит, что в задаче целочисленного сбалансирования решений нет (т.к. невозможно построить обобщенный путь прорыва).

б) Поток не является максимальным.

Обозначим имеющийся поток через φ^1 . По теореме 4 существует полный поток φ^2 величины $\varphi_z^2 = \varphi_z^1 + 2$ такой, что $G_{\varphi^0} = \{S_i\} \cup \{C_j\}$, где $\varphi^0(U) = \varphi^2(U) - \varphi^1(U)$, $\{S_i\}$ – множество всех обобщенных путей прорыва из x_{000} в z , а $\{C_j\}$ – множество всех кратных циклов, причем $|\{S_i\}| = 1$.

Изменим поток φ^2 таким образом, чтобы $\{C_j\} = \emptyset$. Это всегда можно сделать, так как изменение потока на кратном цикле не приведет к нарушению условий неразрывности потока в вершине и неотрицательности потока. Однако теперь поток φ^2 может уже не быть полным.

В результате указанных действий G_{φ^0} будет состоять из единственного обобщенного пути прорыва S_0 . Следовательно, для того, чтобы увеличить на 2 поток φ^1 , достаточно найти какой-либо обобщенный путь прорыва.

Докажем теперь, что обобщенный путь прорыва может быть найден обобщенным алгоритмом пометок.

Заметим, что проекции μ_1 и μ_2 обобщенного пути прорыва S_0 на части G_1 и G_2 сети есть объединение одного пути прорыва из x_{000} в z и некоторых циклов, причем никакие множества циклов $\{C_q^1\} \subset \mu_1$ и $\{C_r^2\} \subset \mu_2$ не образуют кратного цикла (иначе $S_0 = S'_0 \cup C_0$). Поэтому возможно построить S_0 , применяя шаги 1–4 обобщенного алгоритма пометок. При выполнении алгоритма может возникнуть ситуация, когда на шагах 1–4 будет построен участок пути, не входящий ни в один из возможных обобщенных путей прорыва. Тогда применяется система откатов шага 5, которая в худшем случае позволит перебрать все возможные варианты. Так как количество этих вариантов конечно и один и тот же вариант не может быть рассмотрен дважды (по построению алгоритма), то обобщенный путь прорыва будет найден за конечное число шагов.

Указанное на шаге 5 дополнительное правило для задачи сбалансирования позволяет прекратить досрочно выполнение алгоритма в случае, когда задача целочисленного сбалансирования не имеет решения. Действительно, возврат по откату в начальную вершину фактически означает, что существует дуга (x_{000}, x_{i00}) ($i \in \overline{1, n}$) или (x_{000}, x'_{000}) , поток по которой не максимален, но она не входит ни в один из обобщенных путей прорыва. Следовательно, максимальный поток в сети целочисленного

сбалансирования будет заведомо меньше $2d_{000}$.

II. Пусть теперь в сети получен максимальный поток величины $2d_{000}$, не удовлетворяющий условиям разрешимости. Будем выполнять этап коррекции потока. Если не существует потока величины $2d_{000}$, отвечающего условиям разрешимости, то выполнение алгоритма, очевидно, приведет к отрицательному ответу. Поэтому будем считать, что такой поток существует. Обозначим его φ^2 . Исходный поток обозначим φ^1 .

По теореме 4 $G_{\varphi^0} = \{C_j\}$, где $\varphi^0(U) = \varphi^2(U) - \varphi^1(U)$, $\{C_j\}$ – множество всех кратных циклов. Разбиение G_{φ^0} на циклы можно провести таким образом, что при удалении любого количества дуг любой C_j перестает быть циклом.

Применим к φ^1 и φ^2 шаг 1 этапа коррекции потока. Затем изменим φ^2 таким образом, чтобы G_{φ^0} не содержал ни одного цикла, не проходящего через основную дугу. Тогда G_{φ^0} будет состоять из циклов, каждый из которых содержит хотя бы одну ненасыщенную в φ^1 основную дугу и не содержит насыщенных в φ^1 основных дуг. Действительно, основные дуги, насыщенные в φ^1 , являются насыщенными и в φ^2 (по построению), поэтому они не принадлежат никакому C_j , а значит, можно временно считать их пропускную способность равной нулю (шаг 2 этапа коррекции потока).

Возьмем произвольный кратный цикл $C_j \in G_{\varphi^0}$. Проекция μ_1 и μ_2 кратного цикла C_j на части G_1 и G_2 сети есть объединение некоторых обычных циклов, причем никакие множества циклов $\{C_q^1\} \subset \mu_1$ и $\{C_r^2\} \subset \mu_2$ ($\{C_q^1\} \cup \{C_r^2\} \neq C_j$) не образуют кратного цикла (в силу построения разбиения G_{φ^0} на циклы).

Цикл C_j обязательно содержит ненасыщенную основную дугу. Удалив ее из цикла, получим C'_j , являющийся обобщенным путем прорыва из конца удаленной дуги в ее начало.

Все вышесказанное позволяет провести доказательство, аналогичное рассуждениям для 2 этапа алгоритма без учета дополнительного правила для задачи сбалансирования. В результате получим, что цикл C_j может быть найден с помощью шагов 3–4 этапа коррекции потока, возможно, с применением системы откатов шага 5.

В силу того, что суммарный поток на основных дугах после отыскания очередного кратного цикла будет увеличиваться по крайней мере на 1, этап коррекции потока выполнится за конечное число шагов.

Таким образом, утверждение теоремы полностью обосновано.

Замечание 1. Хотя величина максимального кратного потока в сети целочисленного сбалансирования может оказаться меньше $2d_{000}$, в каждой из частей G_1 и G_2 сети существует максимальный поток величины d_{000} , однако эти потоки не всегда возможно согласовать.

Замечание 2. Обобщенный алгоритм пометок имеет экспоненциальную трудоемкость, а задача целочисленного сбалансирования трехмерной матрицы является NP -полной (обоснование приведено в следующем разделе данной статьи).

7. NP -полнота задачи целочисленного сбалансирования трехмерной матрицы

В данном разделе мы покажем NP -полноту задачи целочисленного сбалансирования трехмерной матрицы. Заметим, что объем входных данных задачи полиномиален, и для проверки того, является ли целочисленная матрица D решением задачи, нам необходимо выполнить не более $(n+1)(m+1)(t+1)$ операций сравнения элементов d_{ijp} с соответствующими элементами исходной матрицы. Поэтому задача сбалансирования принадлежит классу NP .

Теорема 6. Задача целочисленного сбалансирования трехмерной матрицы является NP -полной.

Доказательство. Рассмотрим NP -полную задачу о 3-сочетаниях (см. [9]).

Задано 3 множества I, J, P с n элементами (можно считать, что $I = J = P = \{1, \dots, n\}$) и подмножество M прямого произведения $I \times J \times P$.

Требуется определить, можно ли из M выбрать подмножество M' , у элементов которого значение каждой координаты i, j или p присутствует ровно 1 раз.

Для того, чтобы установить NP -полноту задачи целочисленного сбалансирования трехмерной матрицы, выполним сведение задачи 3-сочетания к задаче сбалансирования. Для сведения выпишем систему линейных неравенств (каждому набору $(i, j, p) \in I \times J \times P$ ставится в соответствие элемент матрицы a_{ijp}):

$$\left\{ \begin{array}{ll} x_{ijp} = 0, & (i, j, p) \notin M; \\ \delta \leq x_{ijp} \leq 1 - (n^2 - 1)\delta, & (i, j, p) \in M; \\ \delta \leq \sum_{i=1}^n \sum_{j=1}^n x_{ijp} \leq 1, & p \in \overline{1, n}; \\ \delta \leq \sum_{i=1}^n \sum_{p=1}^n x_{ijp} \leq 1, & j \in \overline{1, n}; \\ \delta \leq \sum_{j=1}^n \sum_{p=1}^n x_{ijp} \leq 1, & i \in \overline{1, n}; \\ n - \frac{1}{2} \leq \sum_{i=1}^n \sum_{j=1}^n \sum_{p=1}^n x_{ijp} < n + \frac{1}{2}, \end{array} \right. \quad (4)$$

где $\delta = \frac{1}{4n^3}$.

Полученная система (4) представляет собой систему ограничений задачи линейного программирования. Так как задача линейного программирования имеет полиномиальную сложность, то какое-либо решение этой системы линейных неравенств (или ее неразрешимость) может быть получено полиномиальным алгоритмом.

Справедливы следующие утверждения:

1. Пусть индивидуальная задача 3-сочетания имеет решение M' . Тогда это решение индуцирует решение задачи целочисленного сбалансирования, определяемой любым решением системы (4).

2. Пусть матрица X является некоторым решением системы (4) и задача, определяемая этим решением, имеет решение $M' = \{x_{ijp} | x_{ijp} = 1\}$. Тогда очевидно, что сочетания индексов этого решения являются решением задачи 3-сочетания. Поэтому это решение M' является решением любой задачи целочисленного сбалансирования, матрица которой определяется любым решением системы (4) (согласно предыдущему абзацу).

3. Если индивидуальная задача 3-сочетания не имеет решения, то либо система (4) не имеет решений, либо любая задача целочисленного сбалансирования, матрица которой определяется решением (4), не имеет решения (иначе это решение индуцирует решение задачи 3-сочетания).

4. Если какая-либо задача целочисленного сбалансирования, определяемая решением системы (4), не имеет решения, то задача 3-сочетания также не имеет решения (в противном случае решение последней определяло бы и решение данной задачи целочисленного сбалансирования).

Тем самым мы произвели полиномиальную сводимость задачи 3-сочетания к задаче целочисленного сбалансирования трехмерной матрицы. Поэтому задача о сбалансировании 3-мерной матрицы является NP -полной.

Утверждение теоремы полностью обосновано.

8. Минимизация ошибок округления в задаче целочисленного сбалансирования матрицы

В случае, когда задача целочисленного сбалансирования имеет несколько решений, возникает вопрос о поиске оптимального решения, то есть такого решения, сумма ошибок округления для которого будет минимальной.

Решение задачи о построении сбалансированной матрицы D , наименее отклоняющейся от исходной, для двумерного случая приведено в [1]. В качестве меры отклонения здесь естественно взять либо сумму абсолютных величин отклонений, либо сумму квадратов отклонений. Показано, что эту задачу можно свести к решению некоторой задачи о наибольшем потоке минимальной стоимости для описанной в разделе 1 транспортной сети со следующей функцией стоимости C , определенной на дугах сети:

$$c(x_{i0}, x_{ij}) = 1 - a_{ij} \quad (i \in \overline{1, n}, j \in \overline{1, m});$$

$$c = 0 \text{ для всех остальных дуг сети.}$$

При этом сбалансированная матрица D^0 , соответствующая наибольшему потоку минимальной стоимости, обращает в минимум функционалы

$$\sum_{i=1}^n \sum_{j=1}^m |d_{ij} - a_{ij}|; \quad (5)$$

$$\sum_{i=1}^n \sum_{j=1}^m (d_{ij} - a_{ij})^2. \quad (6)$$

Перейдем теперь к трехмерному случаю. Поставим задачу *минимизации ошибок округления* в задаче целочисленного сбалансирования трехмерной матрицы как задачу поиска такой целочисленной сбалансированной матрицы D , для которой сумма ошибок округления

$$\sum_{i=1}^n \sum_{j=1}^m \sum_{p=1}^t |d_{ijp} - a_{ijp}| \quad (7)$$

будет минимальной. Эта задача впервые была рассмотрена в [6].

Прежде чем привести алгоритм минимизации ошибок округления, дадим несколько определений.

Пусть в кратной сети целочисленного сбалансирования $G(X, U)$, соответствующей некоторой матрице A , имеется максимальный кратный поток $\varphi(U)$ величины $\varphi_z = 2d_{000}$, удовлетворяющий условиям разрешимости (1). Для каждой вершины вида x_{ijp} ($i \in \overline{1, n}$, $j \in \overline{1, m}$, $p \in \overline{1, t}$) определим *ошибку* α_{ijp} :

$$\alpha_{ijp} = \begin{cases} a_{ijp}, & f(x_{ij0}, x_{ijp}) = 0; \\ 1 - a_{ijp}, & f(x_{ij0}, x_{ijp}) = 1. \end{cases}$$

Определим *суммарную ошибку потока* $\varphi(U)$ как величину

$$S(\varphi) = \sum_{i=1}^n \sum_{j=1}^m \sum_{p=1}^t \alpha_{ijp}.$$

Заметим, что значение $S(\varphi)$ совпадает со значением функционала (7) для соответствующей матрицы D .

Определим теперь *стоимость прохода через вершину* x_{ijp} ($i \in \overline{1, n}$, $j \in \overline{1, m}$, $p \in \overline{1, t}$):

$$s(x_{ijp}) = \begin{cases} 1 - 2a_{ijp}, & f(x_{ij0}, x_{ijp}) = 0; \\ 2a_{ijp} - 1, & f(x_{ij0}, x_{ijp}) = 1. \end{cases}$$

Пусть в сети $G(X, U)$ определен максимальный кратный поток $\varphi(U)$ величины $\varphi_z = 2d_{000}$, удовлетворяющий условиям разрешимости (1). Пусть имеется кратный цикл $C(X', U')$ в сети $G(X, U)$, проходящий насыщенные дуги в обратном направлении, а ненасыщенные – в прямом направлении. Цикл $C(X', U')$ назовем *циклом отрицательной стоимости*, если выполняется

$$\Delta S \equiv \sum_{x_{ijp} \in X'} s(x_{ijp}) < 0.$$

При этом при добавлении к потоку $\varphi(U)$ цикла $C(X', U')$ суммарная ошибка $S(\varphi)$ уменьшится на величину $|\Delta S|$.

Теорема 7. Пусть в кратной сети целочисленного сбалансирования $G(X, U)$, соответствующей некоторой матрице A , существуют два максимальных потока $\varphi^1(U)$ и $\varphi^2(U)$, причем $\varphi_z^1 = \varphi_z^2 = 2d_{000}$ и для обоих потоков выполнены условия разрешимости (1). Пусть $S(\varphi^1) > S(\varphi^2)$. Тогда в сети $G(X, U)$ с потоком $\varphi^1(U)$ существует цикл отрицательной стоимости.

Доказательство. Пусть $\varphi^0(U) = \varphi^2(U) - \varphi^1(U)$. По теореме 4 $G_{\varphi^0} = \{C_j\}$, где $\{C_j\}$ – объединение некоторых кратных циклов. Покажем, что среди них есть цикл отрицательной стоимости.

Предположим, что это не так, то есть все циклы имеют нулевую или положительную стоимость. Будем добавлять циклы C_j по одному к потоку $\varphi^1(U)$. В результате получим следующую цепочку неравенств:

$$\begin{aligned} S(\varphi^2) &< S(\varphi^1) = S(\varphi(G_{\varphi^1})) \leq S(\varphi(G_{\varphi^1} \cup C_1)) \leq \\ &\leq S(\varphi(G_{\varphi^1} \cup C_1 \cup C_2)) \leq \dots \leq S(\varphi(G_{\varphi^1} \cup \bigcup_{\{C_j\}} C_j)) = S(\varphi^2). \end{aligned}$$

Полученное противоречие доказывает теорему.

Доказанная теорема фактически означает, что для решения задачи минимизации ошибок округления достаточно найти какое-то решение задачи целочисленного сбалансирования, и затем последовательно улучшать это решение, находя циклы отрицательной стоимости в сети целочисленного сбалансирования и добавляя их к потоку, соответствующему найденному решению задачи сбалансирования.

Алгоритм решения задачи минимизации ошибок округления может выглядеть следующим образом:

1. С помощью описанного в разделе 5 алгоритма нахождения максимального потока в кратной сети целочисленного сбалансирования найдем целочисленную сбалансированную матрицу D .
2. Во всех случаях, когда возможно, перенесем поток с дополнительных дуг на основные (все основные дуги при этом станут насыщенными).
3. Построим *оптимизационную сеть* (кратную сеть минимизации ошибок округления), включая в нее все вершины кратной сети целочисленного сбалансирования, кроме вершин x_{000} , z_1 , z_2 , z . Также в оптимизационную сеть включаются все дуги сети, связывающие добавленные в оптимизационную сеть вершины, кроме основных дуг. При этом ненасыщенным дугам исходной сети соответствуют прямые дуги в оптимизационной сети, а насыщенным дугам соответствуют обратные дуги. Пропускная способность всех дуг полагается равной 1 (так как изменение потока на не равную 1 величину по какой-либо дуге в исходной сети, не связанное с изменением потока по основным дугам и дугам вида (x_{000}, x_{i00}) , (x_{0j0}, z_1) , (x_{00p}, z_2) ($i \in \overline{1, n}$, $j \in \overline{1, m}$, $p \in \overline{1, t}$), невозможно в силу построения сети целочисленного сбалансирования и условий неразрывности потока в каждой вершине).
4. Для вершин x_{ijp} ($i \in \overline{1, n}$, $j \in \overline{1, m}$, $p \in \overline{1, t}$) оптимизационной сети определяем функцию $s(x_{ijp})$ стоимости прохода через вершину.
5. Определяем сумму ошибок округления S .
6. Находим цикл отрицательной стоимости:

- 1) формируем массив *стартовых вершин* из тех вершин, для которых $s(x_{ijp}) < 0$, упорядочиваем его по возрастанию $s(x_{ijp})$;
- 2) выбираем очередную вершину x_{ijp} из списка стартовых вершин. Если такой вершины нет, то полученный поток является потоком минимальной стоимости; выполнение алгоритма прекращаем;
- 3) с помощью обобщенного алгоритма пометок в оптимизационной сети ищем обобщенный путь прорыва из выбранной вершины x_{ijp} в вершину x_{ij0} , если прямая дуга (x_{ij0}, x_{ijp}) принадлежит оптимизационной сети, или из вершины x_{ij0} в вершину x_{ijp} , если обратная дуга (x_{ijp}, x_{ij0}) принадлежит оптимизационной сети. Если этот путь удастся найти, переходим на шаг 4), иначе переходим на шаг 2);
- 4) объединение найденного пути прорыва и дуги (x_{ij0}, x_{ijp}) (прямой или обратной) даст кратный цикл C . Определяем его стоимость

$$\Delta S = \sum_{x_{ijp} \in C} s(x_{ijp}).$$

Если $\Delta S < 0$, то цикл C имеет отрицательную стоимость, переходим на шаг 7. Иначе делаем откат до ближайшей точки ветвления и возвращаемся на шаг 3).

7. Добавляем цикл отрицательной стоимости к потоку:

Для x_{ijp} ($i \in \overline{1, n}$, $j \in \overline{1, m}$, $p \in \overline{1, t}$), входящих в цикл отрицательной стоимости C , меняем значение d_{ijp} с 0 на 1 или с 1 на 0 и пересчитываем соответствующие суммы.

Пересчитываем $S = S + \Delta S$.

Для x_{ijp} ($i \in \overline{1, n}$, $j \in \overline{1, m}$, $p \in \overline{1, t}$), входящих в цикл C , меняем знак $s(x_{ijp})$ на противоположный.

Если все $s(x_{ijp}) \geq 0$ или существует только одна вершина $x_{i_0j_0p_0}$ такая, что $s(x_{i_0j_0p_0}) < 0$ и при этом $|s(x_{i_0j_0p_0})| < |s(x_{ijp})|$ ($i \in \overline{1, n}$, $j \in \overline{1, m}$, $p \in \overline{1, t}$), то матрица D является оптимальной по стоимости, и выполнение алгоритма прекращаем.

В оптимизационной сети меняем ориентацию всех дуг, входящих в цикл C , на противоположную и переходим к шагу 6.

Теорема 8. Если задача целочисленного сбалансирования трехмерной матрицы имеет решение, то имеет решение и задача минимизации ошибок округления. Это решение может быть найдено при помощи указанного алгоритма.

Доказательство. Первая часть утверждения теоремы очевидна: действительно, если задача сбалансирования имеет решение, то среди всех ее решений всегда можно выбрать решения с минимальной ошибкой округления.

Повторяя рассуждения из доказательства теоремы 5, можно доказать, что с помощью обобщенного алгоритма пометок в оптимизационной сети всегда можно

найти кратный цикл (если он существует), проходящий через любую вершину x_{ijp} с отрицательной стоимостью прохода через нее.

Построение алгоритма позволит в худшем случае перебрать все возможные циклы, проходящие через все вершины x_{ijp} с отрицательной стоимостью прохода через них. Если среди них существует цикл отрицательной стоимости, то он будет найден. Более того, если в оптимизационной сети существует цикл отрицательной стоимости, то он содержит одну из таких вершин.

Следовательно, если в оптимизационной сети существует цикл отрицательной стоимости, то он может быть найден при помощи описанного выше алгоритма.

Так как после нахождения очередного цикла отрицательной стоимости суммарная ошибка потока $S(\varphi)$ уменьшается на натуральную величину $|\Delta S|$ и $S \geq 0$, то алгоритм выполнится за конечное число шагов.

Таким образом, утверждение теоремы полностью обосновано.

9. Сравнительный анализ алгоритмов целочисленного сбалансирования

В этом и в следующем пункте для удобства изложения под обобщенным алгоритмом пометок мы будем понимать все три этапа алгоритма нахождения максимального кратного потока для решения задачи сбалансирования.

В работе [2] было получено сведение задачи целочисленного сбалансирования к задаче целочисленного линейного программирования. Эта задача может быть решена при помощи первого алгоритма Гомори (описание см., например, в [8]). В связи с тем, что и алгоритм пометок, и метод Гомори имеют экспоненциальную трудоемкость, возникает вопрос об эффективности этих алгоритмов в той или иной ситуации.

Для выполнения сравнительного анализа двух алгоритмов была написана программа² на языке C#. В ходе исследований было проведено 1000 экспериментов с использованием ПК со следующими характеристиками: процессор – Pentium 4 1.8GHz, RAM – 1 Gb. Испытания проводились сериями по 10 и по 20 экспериментов, внутренние части исходных трехмерных матриц получались при помощи датчиков следующих распределений:

- 1) равномерное;
- 2) экспоненциальное с функцией распределения $f(x) = e^{-x}$, $x \geq 0$;
- 3) нормальное $N(0; 1)$;
- 4) нормальное $N(0.5; 1)$;
- 5) нормальное $N(0.75; 1)$.

Эксперименты формировались из тестов двух типов:

- 1) *Random-тесты* – матрицы с десятичными элементами из интервала $[0, 1)$, получаемыми при помощи одного из указанных распределений и содержащими один знак после запятой;

²Свидетельство о государственной регистрации программы для ЭВМ №2010611519.

2) *0.5-тесты* – матрицы, состоящие из элементов 0 и 0.5, получаемых при помощи одного из распределений.

Всего было проведено 500 экспериментов с тестами каждого типа. В итоге были получены следующие результаты:

1) для Random-тестов: алгоритм пометок выполнен быстрее в 87.8 % случаев; метод Гомори – в 12.2 % случаев;

2) для 0.5-тестов: алгоритм пометок выполнен быстрее в 61.4 % случаев; метод Гомори – в 28.2 % случаев и в 10.4 % случаев время выполнения было равным;

3) результат для всех тестов: алгоритм пометок выполнен быстрее в 74.6 % случаев; метод Гомори – в 20.2 % случаев и в 5.2 % случаев время выполнения было равным.

Заметим, однако, что, хотя алгоритм пометок оказывается лучше на большинстве примеров, общее время выполнения алгоритма пометок существенно больше общего времени выполнения метода Гомори. Это связано с тем, что при возникновении ситуации, когда алгоритму пометок необходимо выполнить перебор, время работы быстро растет с возрастанием глубины этого перебора. Так, даже при достаточно небольшой размерности внутренней части матрицы ($3 \times 3 \times 4$), время выполнения может превысить час.

Вышесказанное подтверждает статистика по сериям экспериментов:

1) для Random-тестов: алгоритм пометок выполнен быстрее в 15 сериях из 40; метод Гомори – в 25 сериях;

2) для 0.5-тестов: алгоритм пометок выполнен быстрее в 10 сериях из 40; метод Гомори – в 29 сериях и в 1 серии время выполнения было одинаковым;

3) результат для всех тестов: алгоритм пометок выполнен быстрее в 25 сериях из 80; метод Гомори – в 54 сериях и в 1 серии время выполнения было одинаковым.

Отметим также, что время выполнения алгоритма пометок превысило 5 минут в 20 из 500 Random-тестов и в 9 из 500 0.5-тестов.

Рассмотрим подробнее статистику по Random-тестам.

Во-первых, заметим, что для размерности внутренней части матрицы $3 \times 3 \times 3$ статистика отличается от средних показателей в пользу алгоритма пометок (93 %). При возрастании же размерности значения стабилизируются возле средних показателей и колеблются от 85 % (равномерное распределение, размерность $3 \times 3 \times 4$) до 90 % ($N(0.75; 1)$, размерность $3 \times 3 \times 4$).

Значение для нормального распределения $N(0.75; 1)$ не случайно – действительно, в случае этого распределения большая часть элементов матрицы находится в интервале $[0.5; 0.9]$, что приводит к большим значениям суммирующих показателей, а следовательно, и к большим величинам пропускных способностей основных дуг сети. Кроме того, при Random-тестах количество целых чисел среди суммирующих показателей невелико, то есть в сети целочисленного сбалансирования присутствуют почти все дополнительные дуги. Такое строение сети сбалансирования позволяет в большинстве случаев получить ответ при помощи алгоритма нахождения полного потока (трудоемкостью $O(nmt)$), либо же при помощи алгоритма пометок без применения правил отката (соответствующая трудоемкость не превышает $O(n^3 m^3 t^3)$). Однако такое строение сети сбалансирования увеличивает и глубину перебора в тех случаях, когда он необходим. Как следствие, время выполнения алгоритма пометок

превышает 5 минут в 10 % экспериментов для $N(0.75; 1)$ и размерности внутренней части матрицы $3 \times 4 \times 4$, в то время как для остальных распределений это число колеблется от 1 % до 6 %.

Отметим, что для всех остальных распределений результаты получились примерно равными друг другу и средним величинам по всем Random-тестам.

Интересными для сравнительной характеристики являются 0.5-тесты. Так как во внутренней части матрицы присутствуют только элементы 0 и 0.5, то большинство сумм, как правило, оказываются целыми. Это, во-первых, сокращает время выполнения метода Гомори (каждая целая сумма влечет исключение одной переменной), а во-вторых, повышает вероятность того, что пример не будет иметь решения (а в этом случае в алгоритме пометок возникнет полный перебор).

В свете вышесказанного закономерна статистика: 61.4 % случаев более быстрого выполнения алгоритма пометок против 87.8 % для Random-тестов. Кроме того, при возрастании размерности процент уменьшается (65 % для размерности $3 \times 3 \times 4$ и 50 % для размерности $3 \times 4 \times 4$ для равномерного распределения).

Однако в силу того, что в сети сбалансирования присутствует мало дополнительных дуг, глубина перебора в алгоритме пометок зачастую оказывается небольшой. Так, только в 9 из 500 примеров время выполнения алгоритма пометок превысило 5 минут, причем 8 случаев были зафиксированы при размерности внутренней части матрицы $3 \times 4 \times 4$ и распределении $N(0.5; 1)$.

Вообще, статистика по трем из пяти рассмотренных распределений (равномерное, экспоненциальное и $N(0.75; 1)$) примерно соответствует средним показателям, в случае же $N(0; 1)$ и $N(0.5; 1)$ наблюдаются некоторые отклонения. Это связано с тем, что распределение $N(0; 1)$ генерирует матрицы, состоящие в основном из нулей, и поэтому глубина перебора в алгоритме пометок не может быть большой, следовательно, процент удачных для этого алгоритма примеров больше, чем в среднем по 0.5-тестам, и убывает медленнее. $N(0.5; 1)$ же генерирует матрицы, состоящие преимущественно из элементов 0.5, поэтому картина здесь получается обратная.

Подведем итоги. Результаты экспериментов подтвердили, что алгоритм пометок, как правило, работает лучше метода Гомори в случаях, когда целых сумм относительно мало, хотя полный перебор может возникнуть и в этих ситуациях. В случае же, когда целых сумм достаточно много, предпочтительнее оказывается алгоритм Гомори. Также лучше оказывается метод Гомори в тех случаях, когда решений нет (в методе пометок тогда неизбежен полный перебор).

Отметим также, что в тех случаях, когда обобщенный алгоритм пометок оказывался быстрее метода Гомори, время выполнения алгоритмов отличалось в среднем на порядок.

Рекомендация по практическому применению рассмотренных алгоритмов может быть следующей:

1. Устанавливаем максимальное допустимое время выполнения алгоритма, зависящее от размерности матрицы и от производительности компьютера.
2. Определяем, какой из алгоритмов будет запущен первым. В случае наличия большого числа целых сумм целесообразно выбрать метод Гомори, в противном случае – алгоритм пометок.

3. Выполняем выбранный алгоритм до тех пор, пока не будет получен ответ, либо не будет превышено максимальное допустимое время.
4. Если было превышено допустимое время, то выполняем шаг 3 для второго алгоритма; если время опять превышает, выдаем ошибку.

10. Результаты вычислительных экспериментов для задач типа (4)

Индивидуальную задачу целочисленного сбалансирования мы будем называть *задачей типа (4)*, если значения элементов ее матрицы удовлетворяют системе (4). Рассмотрим статистику вычислительных экспериментов с задачами такого вида.

Прежде всего отметим, что испытания проводились в тех же условиях и с помощью той же программы, что и эксперименты, рассмотренные в разделе 9. Всего было проведено 400 экспериментов с задачами типа (4). Испытания проводились сериями по 10 тестов. Минимальная размерность внутренней части матрицы в рассматриваемых примерах – $2 \times 2 \times 2$, максимальная – $7 \times 7 \times 7$.

В итоге алгоритм пометок выполнялся быстрее в 84.5 % случаев; метод Гомори – в 4.75 % случаев и в 10.75 % случаев время выполнения было равным. Для серий экспериментов ситуация была похожей: алгоритм пометок выполнялся быстрее в 35 сериях из 40; метод Гомори – в 4 сериях и в 1 серии время выполнения было одинаковым.

Однако для задач типа (4) имеет смысл рассмотреть статистику для различных размерностей отдельно.

Так, при $n = 2$ и $n = 3$ время выполнения обоих алгоритмов, как правило, одинаково, либо же отличается незначительно. При $n = 4$ и $n = 5$ обобщенный алгоритм пометок выполняется быстрее в 100 % случаев, при этом время выполнения алгоритмов отличается в среднем в 30 раз. При $n = 6$ среднее время выполнения алгоритма пометок меньше, чем среднее время выполнения метода Гомори, однако появляются примеры (их 3.33 % среди проведенных экспериментов), которые решаются дольше алгоритмом пометок. При $n = 7$ среднее время выполнения становится вновь примерно одинаковым, однако уже в 27.5 % тестов метод Гомори работает быстрее.

Такая разница в полученных статистиках для различных размерностей объясняется тем, что при $n = 4$ и $n = 5$ даже в случае возникновения перебора в алгоритме пометок время его выполнения меньше минимального времени работы метода Гомори на примерах той же размерности. В случае же $n = 7$ глубина перебора в алгоритме пометок может быть существенно глубже, и, как следствие, время выполнения намного увеличивается.

Так, минимальное время выполнения обобщенного алгоритма пометок для проведенных экспериментов при $n = 7$ составило 00:00:00.0312500, максимальное – 00:06:50.7031250. Минимальное время работы метода Гомори – 00:00:16.5625000, максимальное – 00:01:48.6093750.

Наибольшее отношение времени выполнения алгоритма пометок ко времени выполнения метода Гомори равно 14.15 при времени выполнения алгоритма пометок 00:04:32.4531250 и времени выполнения метода Гомори 00:00:19.2500000.

Наибольшее отношение времени выполнения метода Гомори ко времени выполнения алгоритма пометок равно 2317 при времени выполнения алгоритма пометок 00:00:00.0468750 и времени выполнения метода Гомори 00:01:48.6093750. Следует подчеркнуть, что время выполнения алгоритма пометок меньше времени выполнения метода Гомори на два порядка и более в 50 % случаев при $n = 7$.

Отметим также, что среднее время работы алгоритма пометок увеличивается в тех случаях, когда количество нулевых элементов внутренней части матрицы примерно равно количеству ненулевых элементов. Для метода Гомори среднее время работы растет с увеличением числа ненулевых элементов внутренней части матрицы.

Список литературы

1. Коршунова Н.М., Рублев В.С. Задача целочисленного сбалансирования матрицы // Современные проблемы математики и информатики. Ярославль: ЯрГУ им. П.Г. Демидова, 2000. Вып. 3. С. 145 – 150.
2. Смирнов А.В. Задача целочисленного сбалансирования трехмерной матрицы и сетевая модель // Моделирование и анализ информационных систем. 2009. Т. 16, №3. С. 70 – 76.
3. Смирнов А.В. О несводимости задачи целочисленного сбалансирования трехмерной матрицы к задаче о наибольшем потоке // Дискретные модели в теории управляющих систем: VIII Международная конференция, Москва, 6-9 апреля 2009 г.: Труды. М.: Издательский отдел факультета ВМиК МГУ им. М. В. Ломоносова; МАКС Пресс, 2009. С. 270 – 273.
4. Рублев В.С., Смирнов А.В. Целочисленное сбалансирование 3-мерной матрицы плана // Труды VII международной конференции «Дискретные модели в теории управляющих систем» (Покровское, 4–6 марта 2006 г.). М.: МГУ, 2006. С. 302 – 308.
5. Рублев В.С., Смирнов А.В. Послойный алгоритм целочисленного сбалансирования трехмерной матрицы // Материалы IX Международного семинара «Дискретная математика и ее приложения», посвященного 75-летию со дня рождения академика О. Б. Лупанова (Москва, МГУ, 18-23 июня 2007 г.). М.: МГУ, 2007. С. 351 – 353.
6. Рублев В.С., Смирнов А.В. Минимизация ошибок округления в задаче целочисленного сбалансирования трехмерной матрицы // Материалы XVII Международной школы-семинара «Синтез и сложность управляющих систем» имени академика О.Б. Лупанова (Новосибирск, 27 октября – 1 ноября 2008 г.). Новосибирск: Изд-во Института математики, 2008. С. 153 – 157.

7. Форд Л.Р., Фалкерсон Д.Р. Потоки в сетях. М.: Мир, 1966.
8. Корбут А.А., Финкельштейн Ю.Ю. Дискретное программирование. М.: Наука, 1969.
9. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982.

The Problem of Integer-Valued Balancing of a Three-Dimensional Matrix and Algorithms of Its Solution

Roublev V. S., Smirnov A. V.

Keywords: integer-valued balancing, three-dimensional matrices, multiple networks, multiple flows, generalized labeling algorithm, first Gomory algorithm, *NP*-completeness, minimization of the errors of rounding off

The article is devoted to the problem of integer-valued balancing of a three-dimensional matrix. The reduction of this problem to the problem of finding a maximum flow in the multiple network of integer-valued balancing and the algorithm for this problem are suggested. Also, the comparative characteristic of two algorithms of integer-valued balancing is made according to the results of the computing experiments. *NP*-completeness of the problem of integer-valued balancing of a three-dimensional matrix is proved in the article. The problem of minimization of the errors of rounding off in the problem of integer-valued balancing is explored.

Сведения об авторах:

Рублев Вадим Сергеевич,

Ярославский государственный университет им. П.Г. Демидова,
профессор кафедры теоретической информатики;

Смирнов Александр Валерьевич,

Ярославский государственный университет им. П.Г. Демидова,
аспирант