А.А. Карпук, Е.С. Шейнкман
Расчетный центр Национального банка Республики Беларусь,
г. Минск, Республика Беларусь

МАТЕМАТИЧЕСКИЕ МОДЕЛИ И АЛГОРИТМЫ ВЫЧИСЛЕНИЯ ЧИСТЫХ НЕТТИНГОВЫХ ПОЗИЦИЙ ПРИ КЛИРИНГЕ МЕЖБАНКОВСКИХ ПЛАТЕЖЕЙ

Математическую модель задачи вычисления чистых неттинговых позиций при клиринге межбанковских платежей в терминах теории исследования операций можно представить в виде:

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{l=1}^{P_{ij}} a_{ijl}y_{ijl} \to \max, \tag{1}$$

$$\sum_{j=1}^{n}\sum_{i=1}^{P_{ij}} a_{ijl}y_{ijl} - \sum_{j=1}^{n}\sum_{i=1}^{P_{ij}} a_{ijl}y_{ijl} \le r_i, i = \overline{1,n}, \tag{2}$$

где $y_{ijl} \in \{0,1\}$, (3)

$P_{ij} i = \overline{1,n},\ j = \overline{1,n}$ – число платежей из банка с индексом i в банк с индексом j, если $i = j$, то $P_{ij} = 0$

$a_{ijl} > 0,\ i = \overline{1,n},\ j = \overline{1,n},\ l = \overline{1,P_{ij}}$ – величина платежа с индексом l из банка с индексом i в банк с индексом j;

$r_i \geq 0,\ i = \overline{1,n}$ – величина резерва, установленного банком с индексом i для проведения клирингового сеанса;

$y_{ijl},\ i = \overline{1,n},\ j = \overline{1,n},\ l = \overline{1,P_{ij}}$ – искомые переменные, доставляющие максимум целевой функции (1) при выполнении условий (2).

В терминах теории графов задачу клиринга межбанковских платежей можно сформулировать следующим образом. Пусть задан ориентированный граф $G=(B,P)$ с кратными дугами без петель с помеченными вершинами и помеченными дугами. Множество вершин графа $B = \{B_i\}\ i = \overline{1,n}$, множество дуг графа $P = \{(i,j,l)\},\ i = \overline{1,n},\ j = \overline{1,n},\ l = \overline{1,P_{ij}}$. Каждая вершина $B_i\ i = \overline{1,n}$, графа соответствует банку, участвующему в клиринговом сеансе. Меткой вершины $r_i > 0$ является величина резерва банка на проведение клирингового сеанса. Каждая дуга графа $(i,j,l),\ i = \overline{1,n},\ j = \overline{1,n},\ l = \overline{1,P_{ij}}$, соответствует одному платежу имеет в качестве метки величину платежа $a_{ijl} > 0$. Требуется найти множество дуг графа, имеющих максимальную сумму меток и удовлетворяющих следующему условию: для любой вершины графа разность между суммой платежей исходящих дуг и суммой платежей входящих дуг не превосходит метку вершины.

Задача (1) – (3) есть задача линейного программирования с булевыми переменными, число которых измеряется десятками тысяч, что исключает непосредственное применение существующих методов поиска оптимального решения для задач этого типа. Известны два подхода к решению задач оптимизации при невозможности использования точных методов: использование эвристических алгоритмов для решения задачи и упрощение исходной задачи за счет ослабления некоторых условий, снятия ограничений или введения новых ограничений.

Эвристический алгоритм REMAINDER, опубликованный в материалах Немецкого федерального банка, состоит из следующих шагов.

1. Для каждого банка построить список всех исходящих платежей, упорядочив их по не убыванию величин платежей.

2. Для каждого банка с индексом $i = \overline{1,n}$ вычислить текущее сальдо S_i по формуле:

$$S_i = \sum_{j=1}^{n}\sum_{i=1}^{P_{ji}} a_{ijl} - \sum_{j=1}^{n}\sum_{i=1}^{P_{ij}} a_{ijl} + r_i.$$

3. Для всех $i = \overline{1,n},\ j = \overline{1,n},\ l = \overline{1,P_{ij}}$ положить $y_{ijl} = 1$.

4. Если для всех $i = \overline{1,n}$ выполняется условие $S_i \geq 0$, то закончить работу.

5. Организовать цикл по числу банков, для которых $S_i < 0$, и выполнить для каждого из этих банков шаги 6 – 9, просматривая банки в порядке не убывания значений S_i.

6. Для рассматриваемого банка с индексом $i' \in \overline{1,n}$ просматривать упорядоченный список исходящих платежей, пока не будет найден платеж (i',j',l'), такой, что $y_{i'j'l'} = 1$ и $S_{j'} - a_{i'j'l'} \geq 0$. Если просмотрен весь список, и такой платеж не найден, то перейти к выполнению шагов 6 – 9 для следующего банка.

7. Удержать найденный платеж, положив $y_{i'j'l'} = 0$, $S_{i'} = S_{i'} + a_{i'j'l'}$, $S_{j'} = S_{j'} - a_{i'j'l'}$. Если $S_{i'} < 0$, то перейти к шагу 6.

8. Просматривать все удержанные ранее исходящие платежи из банка с индексом i' в порядке не возрастания величин платежей и не возрастания весовых коэффициентов платежей, и для каждого очередного удержанного платежа (i',j',l') выполнить действия шага 9.

9. Если $S_{j'} - a_{i'j'l'} < 0$, то перейти к рассмотрению следующего удержанного платежа, иначе удалить платеж (i',j',l') из числа удержанных платежей, положив $x_{i'j'l'} = 1$, $S_{i'} = S_{i'} - a_{i'j'l'}$, $S_{j'} = S_{j'} + a_{i'j'l'}$, и перейти к рассмотрению следующего удержанного платежа. Если просмотрены все удержанные платежи, то перейти к выполнению шагов 6 – 9 для следующего банка.

10. Если для всех банков, для которых выполнялись шаги 6 – 9, выполняется условие $S_i \geq 0$, то закончить работу.

11. Организовать цикл по числу банков, для которых $S_i < 0$, и выполнить для каждого из этих банков шаги 12 – 15, просматривая банки в порядке не убывания значений S_i.

12. Для рассматриваемого банка с индексом $i' \in \overline{1,n}$ просматривать упорядоченный список исходящих платежей пока не будет найден платеж (i',j',l'), такой что $y_{i'j'l'} = 1$. В отличие от шага 6, выполнение условия $S_{j'} - a_{i'j'l'} \geq 0$ здесь не требуется. Если просмотрен весь список, и такой платеж не найден, то перейти к выполнению шагов 12 – 15 для следующего банка.

13. Удержать найденный платеж, положив $y_{i'j'l'} = 0$, $S_{i'} = S_{i'} + a_{i'j'l'}$, $S_{j'} = S_{j'} - a_{i'j'l'}$. Если $S_{i'} < 0$, то перейти к шагу 12.

14. Просматривать все удержанные ранее исходящие платежи из банка с индексом i' в порядке не возрастания величин платежей и не возрастания весовых коэффициентов платежей, и для каждого очередного удержанного платежа (i',j',l') выполнить действия шага 15.

15. Если $S_{j'} - a_{i'j'l'} < 0$, то перейти к рассмотрению следующего удержанного платежа, иначе удалить платеж (i',j',l') из числа удержанных платежей, положив $x_{i'j'l'} = 1$, $S_{i'} = S_{i'} - a_{i'j'l'}$, $S_{j'} = S_{j'} + a_{i'j'l'}$, и перейти к рассмотрению следующего удержанного платежа. Если просмотрены все удержанные платежи, то перейти к выполнению шагов 12 – 15 для следующего банка.

16. Если для всех банков с номерами $i = \overline{1,n}$ выполняется условие $S_i \geq 0$, то закончить работу, иначе перейти к шагу 5.

Традиционный приемом упрощения задач дискретной оптимизации является переход к непрерывным переменным. В нашем случае этот прием состоит в замене ограничений (3) на ограничения

$$0 \le y_{ijl} \le 1,\ i = \overline{1,n},\ j = \overline{1,n},\ l = \overline{1,P_j}\ . \tag{4}$$

После такой замены становится несущественным, какие конкретные платежи проводятся в оптимальном решении задачи (1), (2), (4). Действительно, обозначив

$$x_{ij} = \sum_{l=1}^{P_{ij}} a_{ijl} y_{ijl}\ ,\ i = \overline{1,n},\ j = \overline{1,n}\ , \tag{5}$$

$$S_{ij} = \sum_{l=1}^{P_{ij}} a_{ijl}\ ,\ i = \overline{1,n},\ j = \overline{1,n}\ , \tag{6}$$

получим следующую задачу линейного программирования:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij} \to \max, \tag{7}$$

$$\sum_{j=1}^{n} x_{ij} - \sum_{j=1}^{n} x_{ji} \le r_i \text{ для всех } i = \overline{1,n}, \tag{8}$$

$$0 \le x_{ij} \le S_{ij} \text{ при } i \ne j,\ x_{ij} = 0 \text{ при } i = j. \tag{9}$$

Число искомых переменных в задаче (7) – (9) равно $n(n-1)$, поэтому в случае, когда в клиринговом сеансе участвуют менее 50 головных банков, задачу можно решить известными методами решения задач линейного программирования. Для перехода от найденного оптимального решения задачи (7) – (9) к решению задачи (1) – (3) для каждой пары банков, для которой $x_{ij} > 0$, требуется решить задачу о рюкзаке. В отличие от классической задачи о рюкзаке, здесь незаполненное место в рюкзаке оставлять нельзя, т. е. реальная сумма величин платежей из банка с индексом i в банк с индексом j может превосходить x_{ij}. Недостающую величину платежа можно взять только из резерва, но даже если этот резерв был, у многих банков он будет израсходован при решении задачи (7) – (9).

Чтобы сохранить максимальную величину резерва до решения задачи о рюкзаке, положим сначала $r_i = 0$ для всех $i = \overline{1,n}$. Можно показать, что в этом случае все ограничения (8) становятся равенствами и имеют вид

$$\sum_{j=1}^{n} x_{ij} - \sum_{j=1}^{n} x_{ji} = 0 \text{ для всех } i = \overline{1,n}. \tag{10}$$

Полученная задача (7), (10), (9) относится к классу транспортных задач, для решения которых разработаны эффективные алгоритмы. В частности, оптимальное решение задачи (7), (10), (9) можно найти путем выделения и анализа всех контуров в графе, полученном из графа $G = (B,P)$ в результате объединения всех дуг, идущих из вершины с индексом $i = \overline{1,n}$ в вершину с индексом $j = \overline{1,n}$, в одну дугу с пропускной способностью S_{ij}.

Кроме того, задача (7), (10), (9) является частным случаем задачи поиска циркуляции минимальной стоимости в сети, поэтому ее можно назвать задачей поиска максимальной циркуляции в ориентированном графе. Известно, что оптимальное решение задачи поиска циркуляции минимальной стоимости дает так называемый алгоритм дефекта.

Двойственная задача к задаче (7), (10), (9) имеет вид

$$\sum_{i=1}^{n}\sum_{j=1}^{n} S_{ij} y_{ij} \to \min \qquad (11)$$

$p_i - p_j + y_{ij} \geq 1$ для всех $i = \overline{1,n}, j = \overline{1,n}, i \neq j$, (12)

$y_{ij} \geq 0$ при $i \neq j$, $y_{ij} = 0$ при $i = j$, где (13)

p_i, $i = \overline{1,n}$, – двойственные переменные, соответствующие ограничениям (10);

y_{ij}, $i = \overline{1,n}$, $j = \overline{1,n}$ – двойственные переменные, соответствующие ограничениям (9).

Положим $y_{ij} = \max\ \{0,(1 - p_i + p_j)\}$ и $a_{ij} = p_i - p_j - 1$, тогда условия дополняющей нежесткости для рассматриваемой пары двойственных задач линейного программирования можно записать в виде: из $a_{ij} > 0$ следует $x_{ij} = 0$; из $a_{ij} < 0$ следует $x_{ij} = S_{ij}$.

Алгоритм дефекта для поиска оптимального решения задачи (7), (10), (9) состоит из следующих шагов.

1. Положить $x_{ij} = 0$, $p_i = 1$, $i = \overline{1,n}$, $j = \overline{1,n}$, $y_{ij} = 1$ при $i \neq j$. Числа x_{ij} задают поток, удовлетворяющий условиям (9) и (10), а вершинные числа p_i и числа y_{ij} удовлетворяют условиям (12) и (13). Вычислить исходную величину дефекта каждой дуги (i,j), $i \in \overline{1,n}$, $j \in \overline{1,n}$, $i \neq j$ по формуле $k_{ij} = a_{ij}\,(x_{ij} - S_{ij})$. Для всех дуг, для которых $S_{ij} > 0$ величина дефекта положительна.

2. Выбрать произвольную дугу (i,j) с положительным дефектом, если таких дуг нет, то закончить работу. Если $a_{ij} < 0$, то считать вершину B_j источником, а вершину B_i стоком, если $a_{ij} > 0$, то считать вершину B_i источником, а вершину B_j Встоком. Решить задачу поиска максимального потока из источника в сток, используя итерационный алгоритм Форда – Фалкерсона. Если на одной из итераций алгоритма величина потока становится больше либо равной дефекту рассматриваемой дуги, то положить $x_{ij} = S_{ij}$ при $a_{ij} < 0$ и $x_{ij} = 0$ при $a_{ij} > 0$, переведя тем самым эту дугу в бездефектные, и перейти к выполнению шага 2 для следующей дефектной дуги. Увеличение потока по другим дугам, найденное при выполнении алгоритма Форда – Фалкерсона, не выводит поток за рамки ограничений (9) и (10) и не приводит к увеличению величины дефекта каждой дуги. Если полученный максимальный поток из источника в сток недостаточен для перевода дуги в бездефектные, то перейти к шагу 3.

3. Пусть C и $\overline{C}$ – множества помеченных и непомеченных вершин соответственно на последней итерации алгоритма Форда – Фалкерсона. Через A_1

обозначим множество дуг (i,j), таких, что $B_i \in C, B_j \in \overline{C}$, $\alpha_{ji} > 0$, $x_{ij} \le S_{ij}$ а через A_2 – множество дуг (i,j), таких, что $B_i \in C, B_j \in \overline{C}$, $\alpha_{ji} < 0$. Если A_1 пусто, то положить $\delta_1 = \infty$, иначе положить $\delta_1 = \min\limits_{A_1}\{\alpha_{ij}\}$. Если A_2 пусто, то положить $\delta_2 = \infty$, иначе положить $\delta_2 = \min\limits_{A_2}\{-\alpha_{ji}\}$. Положить $\delta = \min\ \{\delta_1, \delta_2\}$. Можно показать, что в отличие от алгоритма дефекта для задачи поиска циркуляции минимальной стоимости в нашем случае всегда $\delta \ne \infty$. Для всех $B_j \in \overline{C}$ положить $p_j = p_j + \delta$ и вычислить новые значения величин a_{ij} и k_{ij} для всех дефектных дуг. При $a_{ij}=0$ бездефектными становятся все дуги, для которых $0 \le x_{ij} \le S_{ij}$, а при $a_{ij} > 0$ бездефектными становятся все дуги, для которых $x_{ij} = 0$. Перейти к шагу 2.

Пусть x_{ij}, $i = \overline{1,n}$, $j = \overline{1,n}$, $i \ne j$ – оптимальное решение задачи (7), (10), (9). Для каждой пары банков с индексами i и j, $i \ne j$, найдем значения булевых переменных y_{ijl}, $l = \overline{1,P_{ij}}$, при которых достигает минимума разность

$$\sum_{l=1}^{P_{ij}} a_{ijl} y_{jil} - x_{ij} \to \min, \tag{14}$$

при условии, что эта разность неотрицательна. Задача (14) есть задача о рюкзаке с минимальным избытком, для решения которой можно модифицировать известные методы решения задачи о рюкзаке. Если для каждого банка с индексом $i = \overline{1,n}$ выполняется условие

$$r_i + \sum_{\substack{j=1 \\ j \ne i}}^{n} \left(\sum_{l=1}^{P_{ji}} a_{jil} y_{jil} - x_{ji} \right) - \sum_{\substack{j=1 \\ j \ne i}}^{n} \left(\sum_{l=1}^{P_{ij}} a_{ijl} y_{ijl} - x_{ij} \right) \ge 0, \tag{15}$$

то найденные значения переменных $y_{ijl}\, i = \overline{1,n}$, $j = \overline{1,n}$, $l = \overline{1,P_{ij}}$, дают решение исходной задачи (1) – (3), которое можно незначительно улучшить за счет использования оставшихся резервов банков. Если для некоторых банков условие (15) не выполняется, и выполнения этого условия нельзя добиться за счет использования оставшихся резервов банков, то для таких банков следует отложить платежи на сумму, равную значению левой части выражения (15) или превосходящую это значение на минимальную величину, и перейти к решению задачи (7), (10), (9).

В настоящее время в автоматизированной системе межбанковских расчетов (АС МБР) Национального банка Республики Беларусь для решения задачи вычисления чистых неттинговых позиций при клиринге межбанковских платежей используется эвристический алгоритм REMAINDER, а также приближенный алгоритм решения задачи (7), (10), (9), основанный на выделении контуров в ориентированном графе. При дальнейшей модернизации АС МБР будет проведена оценка возможности использования описанного алгоритма дефекта, дающего точное решение задачи (7), (10), (9).