

транзакции 500 байт, таким образом, один блок содержит записи около 2000 транзакций. Отметим, что в блок помещается не сама запись или транзакция, а ее 32-битное представление, называемое корнем Меркла.

Технология блокчейн и распределенные реестры в целом кажутся сегодня очень перспективными с точки зрения хранения данных, а также создания децентрализованного механизма сотрудничества внутри компании, бизнес-сообщества и даже между странами. Уже сегодня многие крупные компании из различных секторов экономики, банковские консорциумы, государственные учреждения начинают свои пилотные проекты на основе технологии распределенного реестра. Однако следует понимать, что с точки зрения корпоративного и государственного сектора это лишь новый тип устройства базы данных, но он имеет потенциал в корне изменить облик бизнес-процессов, лежащих в основе деятельности компаний, и повысить прозрачность деятельности органов власти и их результатов.

Литература

1. Blockchain Vs Distributed Ledger Technology // 101blockchains.com [Электронный ресурс] – Режим доступа: <https://101blockchains.com/blockchain-vs-distributed-ledger-technology/> – Дата доступа: 13.02.2023.
2. Технология распределенного реестра и блокчейн // crypto.ru [Электронный ресурс] – Режим доступа: <https://crypto.ru/raspredelennyy-reestr-i-blockchain/> – Дата доступа: 13.02.2023.
3. Шилов, К. Д. Блокчейн и распределенные реестры как виды баз данных / К. Д. Шилов, А.В. Зубарев. // Инновации. 2018. №12 (242) [Электронный ресурс] – Режим доступа: <https://cyberleninka.ru/article/n/blokcheyn-i-raspredelennye-reestry-kak-vidy-baz-dannyh> – Дата доступа: 13.02.2023.
4. Пряников, М. М. Блокчейн как коммуникационная основа формирования цифровой экономики: преимущества и проблемы / М. М. Пряников, А.В. Чугунов. // International Journal of Open Information Technologies. 2017. №6. [Электронный ресурс] – Режим доступа: <https://cyberleninka.ru/article/n/blokcheyn-kak-kommunikatsionnaya-osnova-formirovaniya-tsifrovoy-ekonomiki-preimuschestva-i-problemy> – Дата доступа: 13.02.2023.

КЛИРИНГОВАЯ СИСТЕМА ВЗАИМОРАСЧЕТОВ МЕЖДУ УЧАСТНИКАМИ ИННОВАЦИОННО-ПРОМЫШЛЕННОГО КЛАСТЕРА

*Сазонова Д. В., Смелов В. В., канд. техн. наук, доцент
Белорусский государственный технологический университет (г. Минск)*

Ключевые слова: инновационно-промышленный кластер, клиринг, кластер.

Инновационно-промышленный кластер – объединение субъектов хозяйствования с целью их эффективного взаимодействия и совместного устойчивого развития [1].

Клиринг – это комплекс взаиморасчётов за оказанные друг другу услуги, проданные товары или ценные бумаги, основанные на безналичных расчётах. Т.е. есть множество компаний, между которыми могут быть встречные обязательства, и задача клиринга сделать их взаиморасчеты как можно более эффективными.

Многосторонний клиринг – это порядок осуществления клиринговой деятельности, при котором данные по всем обязательствам каждого из участников кластера собираются централизованно в клиринговой системе и итоговые обязательства все компании-участники получают также централизованно [2, 3].

Кластер рассматривается как система:

$$S \equiv \langle V, M(t), E, D, P(t) \rangle,$$

где $V = \{v_1, v_2, \dots, v_n\}$ – множество n участников ПК,

$M(t) = (m_{i,j}(t))_{n \times n}$ – квадратная матрица размерности n , значение каждого элемента $m_{i,j}(t) \geq 0$ которой, равно величине обязательств участника V_i перед участником V_j на момент времени t ,

$E = (e_{i,j})_{n \times n}$ – квадратная матрица, значение каждого элемента $e_{i,j} \geq 0$ которой равно величине издержек (например, стоимость банковского перевода) участника V_i при погашении обязательств перед участником V_j ,

$D = \{d_1, d_2, \dots, d_n\}$ – коэффициенты, определяющие величину издержек для участников ПК, возникающие при привлечении денежных средств (например, дневная процентная ставка по банковскому кредиту),

$P(t) = \{p_1(t), p_2(t), \dots, p_n(t)\}$ – потенциалы участников, где каждое значение равно максимальной величине, которую участник V_i может потратить на погашение своих суммарных обязательства перед другими участниками в момент времени t .

Матрица обязательств и потенциалы участников зависят от времени.

Состояние системы S на момент времени t описывается матрицей $M(t)$. Будем предполагать далее время дискретным и говорить о начальном состоянии $M(0)$ системы S и пошаговом его изменении $M(0), M(1), M(2) \dots$. При это изменение состояния системы между шагами t и $t + 1$ осуществляется в две фазы: $M(t) \rightarrow M'(t + 1) \rightarrow M(t + 1)$ (рисунок 1).

$$S \equiv \langle V, M(t), E, \Lambda \rangle$$

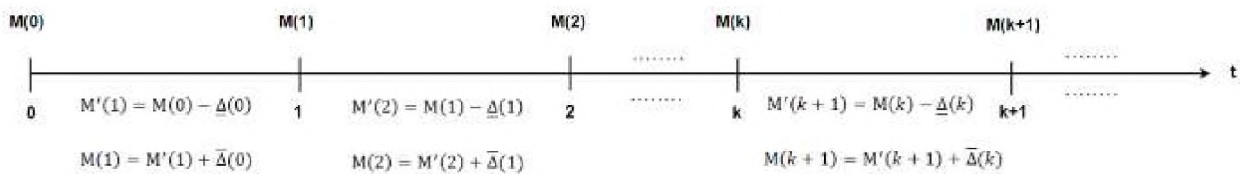


Рис. 1. Схема изменения состояния системы S

Пусть элементы матрицы

$$M = \begin{pmatrix} m_{1,1} & \cdots & m_{1,n} \\ \vdots & \ddots & \vdots \\ m_{n,1} & \cdots & m_{n,n} \end{pmatrix} -$$

обязательства $m_{i,j} = m_{i,j}(t_1)$ участников ПК в фиксированный момент времени $t = t_1$.

Матрицу M можно рассматривать как матрицу смежности взвешенного ориентированного графа, отображающего состояние системы в фиксированный момент времени. На рисунке 2 приведен пример матрицы смежности и соответствующего графа, отображающего обязательства между пятью участниками ПК.

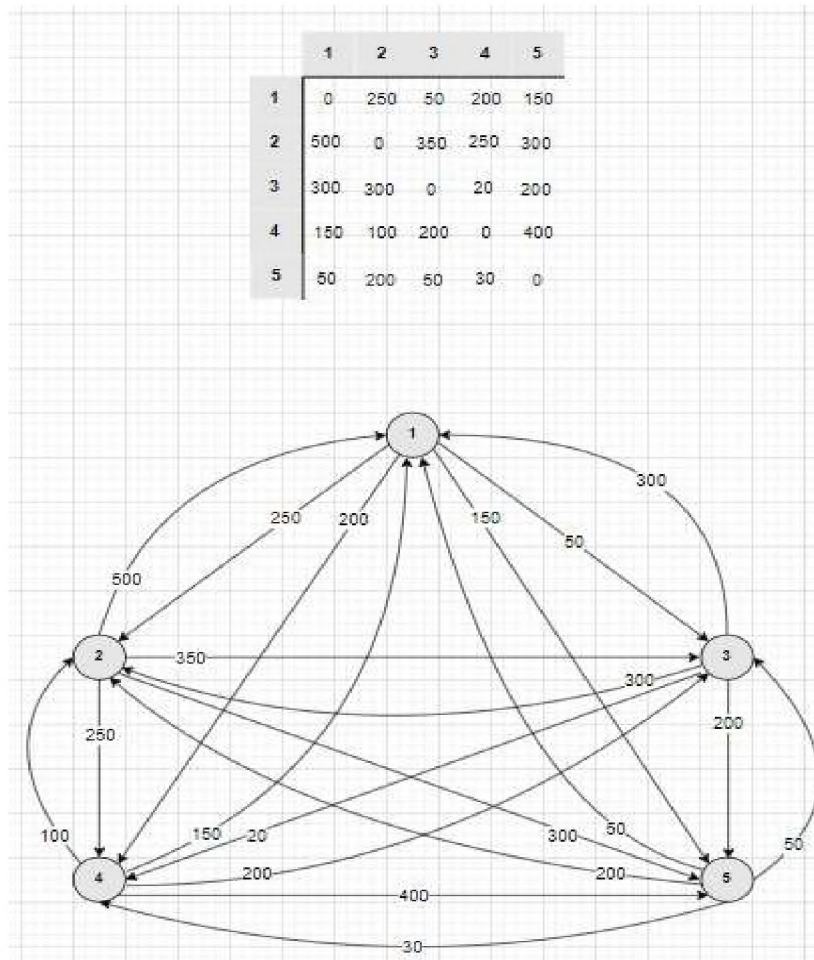


Рис. 2. Граф, отображающий состояние системы S

Вершины графа соответствуют участникам ПК, дуги – обязательствам между ними, а веса дуг – величинам этих обязательств.

На рисунке 3 отображены примеры матриц M – матрицы обязательств, Δ – величины погашения обязательств, M' – матрицы новых обязательств, как разница между матрицами M и Δ , а также граф, соответствующий промежуточному состоянию M' .

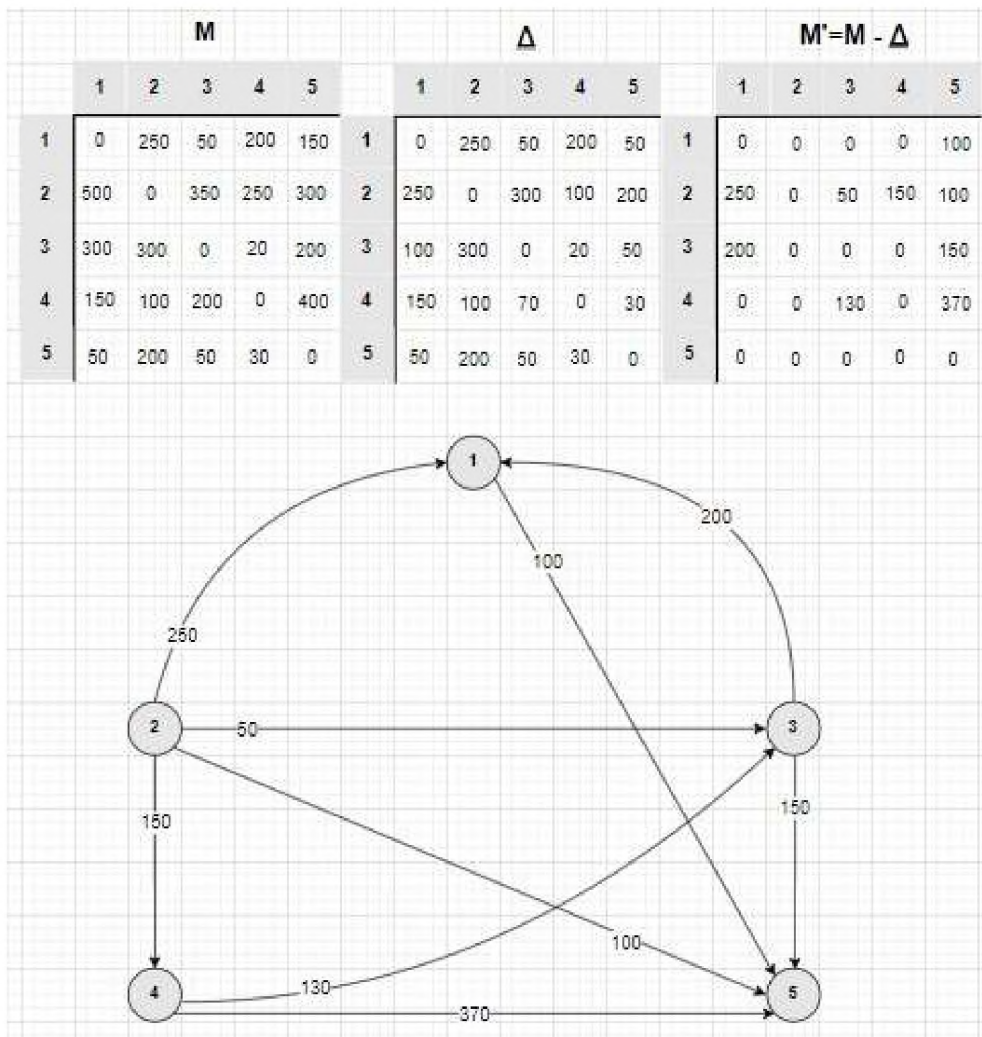


Рис. 3. Граф, отображающий состояние системы S после погашения обязательств

Сформулируем задачу клиринга расчетов в ПК, как задачу математического программирования:

$$F = \sum_i \sum_j \chi_{i,j} e_{i,j} + d_i r_{i,j}, \quad i, j = \overline{1, n}. \quad (1)$$

При следующих ограничениях:

$$\chi_{i,j} = \begin{cases} 0, & r_{i,j} = 0 \\ 1, & r_{i,j} \neq 0 \end{cases}, \quad (2)$$

$$r_{i,j} \geq 0, \quad (3)$$

$$\forall i \sum_j r_{i,j} = \min(p_i, \max(0, \sum_j m_{i,j} - \sum_j m_{j,i})), \quad (4)$$

Задача сводится к минимизации целевой функции, которая равна сумме издержек на перевод и на привлечение денег. Решением $R = (r_{i,j})_{n \times n}$ является план взаиморасчетов, минимизирующий эти издержки.

Предположим, что участники ПК обладают неограниченными потенциалами, а издержки на погашения обязательств для всех участников равны 0.

Тогда целевая функция (1) примет вид:

$$F = \sum_i \sum_j d_i r_{i,j}, \quad i, j = \overline{1, n}, \quad (1')$$

а ограничение (4):

$$\forall i \sum_j r_{i,j} = \max(0, \sum_j m_{i,j} - \sum_j m_{j,i}). \quad (4')$$

Решение задачи сводится к построению $V = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$ - вектора суммарных обязательств,

каждый элемент которого содержит положительные либо отрицательные значения. Положительные означают, что у соответствующих участников имеются обязательства перед другими участниками кластера, а отрицательные – обязательства других участников.

Далее задача сводится к приведению вектора V к нулю и построению матрицы обязательств.

Рассмотрим алгоритм *pure_glade* построения матрицы R . Псевдокод алгоритма представлен на рисунке 4.

```

/* R0 → R1 → … Rk → … Rs */
pure_glade
-----
1.  $\forall (i, j) \Rightarrow r_{i,j}^0 = 0, \forall i \Rightarrow b_i^0 = b_i, k = 0.$ 
2.  $\exists (i, j) | b_i^k > 0 \wedge b_j^k < 0 \Rightarrow stop.$ 
    $k = k + 1$ 
    $\alpha = \min(b_i^{k-1}, |b_j^{k-1}|).$ 
    $r_{i,j}^k = r_{i,j}^{k-1} + \alpha,$ 
    $b_i^k = b_i^{k-1} - \alpha,$ 
    $b_j^k = b_j^{k-1} + \alpha.$ 
3. go to 2.

```

Рис. 4. Псевдокод алгоритма *pure_glade* построения матрицы R

На шаге 1 алгоритма осуществляется инициализация начальной нулевой матрицы R^0 и вектора V^0 вначале равного вектору V .

В рамках каждой итерации выполняются 2 шага (на рисунок 4 – шаги 2 и 3). В результате каждой итерации формируются матрица R^k и вектор V^k . На каждом шаге итерации изменяется один элемент $r_{i,j}^k$ текущей матрицы R^k и два элемента (уменьшаются абсолютные значения обязательств и требований) текущей матрицы V^k . Построение матрицы R завершается, если все элементы вектора V^k равны нулю.

На рисунке 5 демонстрируется пример применения алгоритма *pure_glade*.

R ⁰						B ⁰	
	1	2	3	4	5		1
1	0	0	0	0	0	1	-350
2	0	0	0	0	0	2	550
3	0	0	0	0	0	3	170
4	0	0	0	0	0	4	350
5	0	0	0	0	0	5	-720

R ¹						B ¹		R ²						B ²	
	1	2	3	4	5		1		1	2	3	4	5		1
1	0	0	0	0	0	1	-180	1	0	0	0	0	0	1	0
2	0	0	0	0	0	2	550	2	0	0	0	0	0	2	550
3	170	0	0	0	0	3	0	3	170	0	0	0	0	3	0
4	0	0	0	0	0	4	350	4	180	0	0	0	0	4	170
5	0	0	0	0	0	5	-720	5	0	0	0	0	0	5	-720

R ³						B ³		R ⁴						B ⁴	
	1	2	3	4	5		1		1	2	3	4	5		1
1	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0
2	0	0	0	0	0	2	550	2	0	0	0	0	550	2	0
3	170	0	0	0	0	3	0	3	170	0	0	0	0	3	0
4	180	0	0	0	170	4	0	4	180	0	0	0	170	4	0
5	0	0	0	0	0	5	-550	5	0	0	0	0	0	5	0

Рис. 5. Пример применения алгоритма

Имеется исходная матрица, содержащая обязательства участников клиринга. На ее основе строится вектор В содержащий обязательства каждого из участников. В процессе алгоритма этот вектор приходит к нулю, а матрица R заполняется обязательствами, которые участники кластера должны погасить окончательно.

На рисунке 6 представлено два варианта плана погашения обязательств, соответствующих двум вариантам матрицы R.

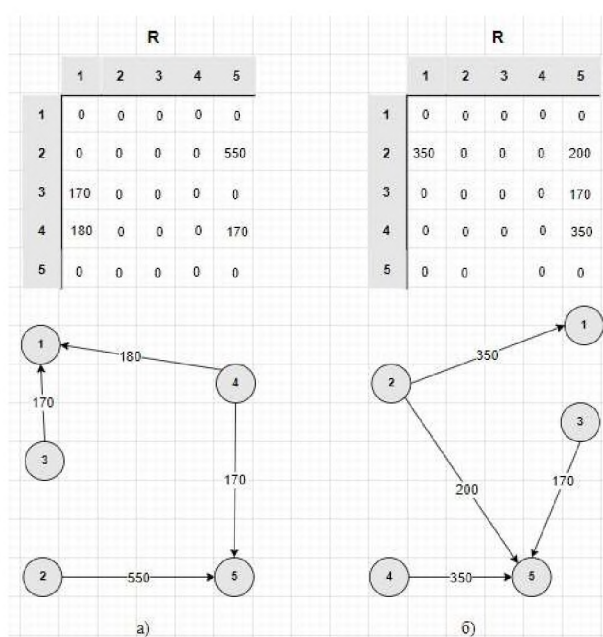


Рис. 6. Два варианта плана погашения обязательств

Следует отметить, что в общем случае результат работы алгоритма зависит от выбора пары (i, j) на шаге 2. Вариант а) совпадает с решением продемонстрированным на рисунке 5, а вариант б) может быть получен при другом порядке выбора пар (i, j) в алгоритме. При этом заметим, что оба решения удовлетворяют условиям (3), (4') и минимизируют целевую функцию (1'), а суммы элементов по строкам и по столбцам в двух вариантах матрицы R совпадают.

Для всех решений (вариантов плана R), полученных с помощью алгоритма *pure_glade*, значения функции (1') совпадают. При этом построение этих матриц не зависит от коэффициентов $d_i, i = \overline{1, n}$. Следовательно, целевую функцию (1') можно упростить:

$$F = \sum_i \sum_j r_{i,j} \quad (1'')$$

Если снять введенное ранее ограничение считать далее, что издержки дебиторов не являются нулевыми $e_{i,j} \geq 0, i, j = \overline{1, n}$, то функция (1) с учетом (1') и (1'') принимает следующий вид:

$$F = \sum_i \sum_j \chi_{i,j} e_{i,j} + r_{i,j}, \quad i, j = \overline{1, n} \quad (1''')$$

Минимизирующий функции (1''') план R при ограничениях (3), (4') и по-прежнему при неограниченном потенциале дебиторов $\forall v_i^+ \Rightarrow p_i \rightarrow +\infty$ может быть получен с помощью алгоритма *pure_expense_glade* (рисунок 7), который является модификацией алгоритма *pure_glade* (рисунок 4).

pure_expense_glade

1. $\forall (i, j) \Rightarrow r_{i,j}^0 = 0, \forall i \Rightarrow b_i^0 = b_i, k = 0.$
2. $\exists (i, j) | b_i^k > 0 \wedge b_j^k < 0 \Rightarrow stop.$
select (i, j) | min(e_{i,j})
 $k = k + 1$
 $\alpha = \min(b_i^{k-1}, |b_j^{k-1}|),$
 $r_{i,j}^k = r_{i,j}^{k-1} + \alpha,$
 $b_i^k = b_i^{k-1} - \alpha,$
 $b_j^k = b_j^{k-1} + \alpha.$
3. *go to 2.*

Рис. 7. Псевдокод алгоритма *pure_expense_glade*

Отличие алгоритма *pure_expense_glade* от алгоритма *pure_glade* заключается в операторе **select** (на рисунке 7 выделен жирным шрифтом), который выбирает пару (i, j) , в порядке возрастания издержек $e_{i,j}$.

На рисунке 8 демонстрируется пример применения алгоритма *pure_expense_glade* для построения плана R в системе S с неограниченными потенциалами участников ПК $p_i \rightarrow +\infty, i = \overline{1, n}$ и издержками на погашения обязательств, заданными матрицей E.

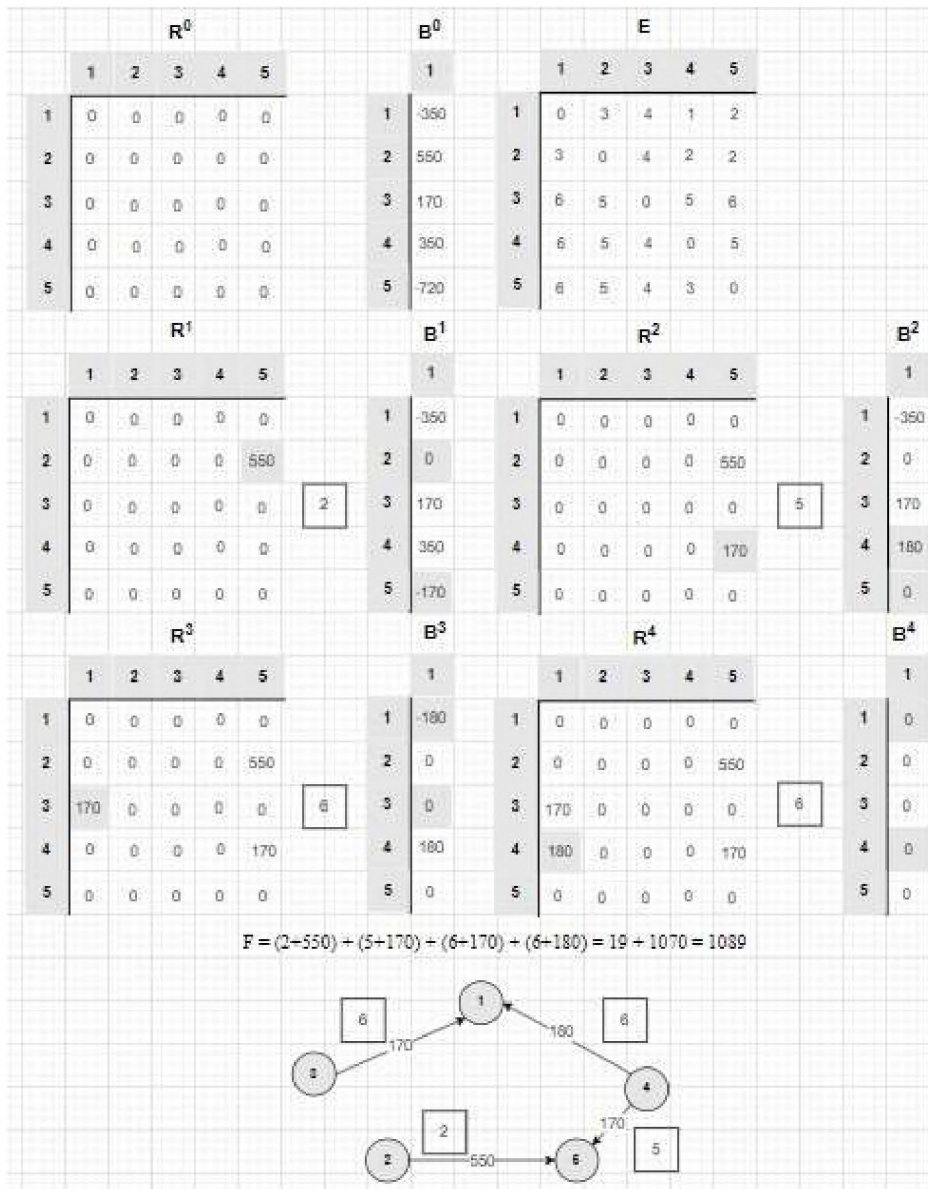


Рис. 8. Пример выполнения алгоритма *pure_expense_glade*

Т. о. в результате получаем решение, которое минимизируют целевую функцию с учетом минимизации издержек по доставке денежных средств.

На рисунке 9 представлено, как может выглядеть последовательность работы клиринговой системы.



Рис. 9. Последовательность работы клиринговой системы

Администратор клиринговой системы открывает сессию.

Пользователи отправляют запросы на авторизацию.

После получения подтверждения об авторизации, пользователи отправляют запрос на участие в клиринге.

В ответ им приходит форма для участия в клиринге в которой они указывают свои обязательства, различные издержки и потенциалы.

После заполнения данных пользователем происходит их обработка и отправка клиринговой сессией данных об обязательствах, которые пользователь должен погасить окончательно.

После того как все пользователи получили данные об обязательствах администратор клиринговой системы закрывает клиринговую сессию.

Выводы:

1. Сложность рассмотренных алгоритмов `pure_glade` и `pure_expense_glade` не превышает $O(n^2)$ поэтому они вполне могут быть применимы для большого количества участников кластера.

2. Предложенные алгоритмы клиринга взаиморасчетов между участниками ПК позволяют получить оптимальный план погашения обязательств в рамках одной клиринговой сессии при существенном допущении – неограниченности ресурсов у дебиторов.

Литература

1. Ташенова, Л. В. Цифровая платформа системообразующего инновационно-активного промышленного кластера: понятие, особенности и структура / Л. В. Ташенова, А. В. Бабкин // Кластеризация цифровой экономики: Глобальные вызовы. Сборник трудов национальной научно-практической конференции с зарубежным участием. В 2-х томах, Санкт-Петербург, 18–20 июня 2020 года / Под редакцией Д.Г. Родионова, А.В. Бабкина. – Санкт-Петербург : ПОЛИТЕХ-ПРЕСС, 2020. – С. 196-208.

2. Маманович, П. А. Клиринг как элемент системы расчетов по операциям с ценными бумагами / П. А. Маманович // Банкаўскі веснік. – 2009. – №10/447. – С.4-11.

3. Немцева, Ю. В. Клиринг обязательств как средство оптимизации системы внутривладельческих расчетов / Ю. В. Немцева, А. В. Беккер // Вестник Алтайской академии экономики и права. – 2019. – № 6-1. – С. 85-91.