

ПРИМЕНЕНИЕ ИНСТРУМЕНТАРИЯ ВРЕМЕННЫХ СЕТЕЙ ПЕТРИ ДЛЯ УПРАВЛЕНИЯ РЕСУРСНЫМИ ПАРАМЕТРАМИ ПРОЕКТА

© 2024 Т. В. Азарнова✉, Ю. В. Бондаренко, Н. А. Поляков, О. А. Медведева

*Воронежский государственный университет
Университетская пл., 1, 394018 Воронеж, Российская Федерация*

Аннотация. Эффективность проекта во многом зависит от используемой стратегии управления ресурсными параметрами задач проекта, которые могут выполняться последовательно и параллельно. Распределение ресурсов может носить детерминированный и стохастический характер, но на практике проекты, как правило, реализуются в условиях риска и неопределенности, поэтому в процессе планирования ресурсного обеспечения проекта нужно учитывать факторы риска и неопределенности, которые могут носить эндогенный (внутренний организационный) и экзогенный (внешний) характер. Для моделирования влияния на проект факторов риска и неопределенности, а также оценки эффективности используемой стратегии ресурсного обеспечения можно использовать инструментарий имитационного моделирования, удовлетворяющий определенным требованиям адаптивности и возможности реализации обратной связи. Подобный инструментарий имитационного моделирования можно построить на базе аппарата сетей Петри. Цель исследования заключается в разработке алгоритма имитационного моделирования процесса выполнения проекта при определенной стратегии ресурсного обеспечения, базирующегося на построении модели данного процесса в виде временной сети Петри. На начальном этапе строится сетевой граф проекта и на основании средних или зафиксированных по некоторому принципу значений вычисляется критический путь, критическое время выполнения всего проекта, и временные параметры для каждого события и каждой работы. По сетевому графу с рассчитанными значениями временных параметров реализации работ строится цветная временная сеть Петри, переходы маркеров в которой осуществляются при выполнении определенного блока временных и ресурсных ограничений. Моделирование процесса выполнения работ на основе цветных временных сетей Петри позволит оценить эффективность управления ресурсами проекта.

Ключевые слова: сети Петри, календарное планирование, сетевой граф, оценки эффективности распределения ресурсов проекта.

ВВЕДЕНИЕ

Для современных предприятий всё более актуальным становится вопрос разработки информационных аналитических систем оценки эффективности распределения ресурсов в рамках реализации проектов [1]. Существует достаточно много крупных информационных систем управления проектами, которые решают целый ряд задач распределения ресурсов [2]. Microsoft Project в моди-

фикации Professional обеспечивает широкие функции анализа параметров проекта и планирования ресурсов, OpenPlan представляет пользователю мощные средства ресурсного и стоимостного планирования, COMFAR позволяет формировать графическую информацию о результатах проекта при варьировании таких параметров как объем реализации, производственные издержки, проценты за кредит, SpiderProject обладает алгоритмами учета практического опыта при управлении ресурсами проекта, 1С Рарус: Управление проектами содержит специальные инструменты для управления рисками проекта. Интерес

✉ Азарнова Татьяна Васильевна
e-mail: ivdas92@mail.ru



Контент доступен под лицензией Creative Commons Attribution 4.0 License.

The content is available under Creative Commons Attribution 4.0 License.

представляют аналитические надстройки, применимые к описанным выше системам, которые позволяли бы имитировать выполнение проекта при определенном ресурсном обеспечении и находить узкие места, способные привести к невыполнению проекта или к нарушению сроков его реализации. Хорошо апробированным инструментом организации имитационных расчетов, позволяющим учитывать целый комплекс требований к реализации работ проекта и факторы риска, а также неопределенности в проекте, являются цветные сети Петри. Сети Петри находят активное применение в задачах управления производством. Инструментальные системы Yasper и GPenSIM предоставляют возможность проводить имитационное моделирование на базе сетей Петри [2]. В случае использования сетей Петри на крупных предприятиях с большим количеством состояний интерес предоставляет возможность сокращения графа достижимости, на что направлено множество работ в данной сфере. Часто исследователи используют подход, в соответствии с которым крупная сеть разбивается на несколько мелких. Менее крупные модули позволяют после проведения декомпозиции с достаточно высокой точностью исследовать имеющиеся состояния системы [3]. В работе Градишар Д. и Мусич Г. В работе [4] рассматривается применение синхронизированных сетей Петри для моделирования и анализа параллельных дискретно-событийных производственных систем, также был предложен алгоритм построения сети Петри на основании информации о структуре производственного процесса и характеристиках продуктов, заложенный в систему управления производственными данными. Также описывается симулятор синхронизированной сети Петри, созданный в Matlab, позволяющий вводить эвристики, на основании которых осуществляется планирование производственных операций. В работе [5] была рассмотрена новая методология управления проектами, основанная на методе критического пути и теории сетей Петри, которую рекомендуется использовать для крупных проектов в условиях, когда в процессе планирования нужно учитывать целый ряд ограниченных ресурсов.

1. ПОСТАНОВКА ЗАДАЧИ

В рамках данного исследования рассматривается применение сетей Петри как дополнительного инструмента, направленного на расширение возможностей алгоритма критического пути по анализу ограниченных ресурсов, проверке структурных свойств графа выполнения работ и для получения оптимальных вариантов распределения ресурсов, которые позволят достичь кратчайшей продолжительности проекта с существующими ограниченными ресурсами. Также с их помощью рассматривается проблема определения оптимального распределения дополнительных ресурсов для повышения производительности, которая решается путем увеличения количества токенов в определенных позициях ресурсов. Критический путь для проекта с учетом ограничений ресурсов можно затем найти, преобразуя модель сети Петри обратно в сетевой граф СРМ с дополнительными условиями по приоритету и применяя необходимые правила сокращения. Модифицированную сеть СРМ можно использовать для поиска адекватного ресурсам критического пути и критических действий, которые требуют большего внимания для управления проектом. В работе [6] предложен алгоритм построения процедуры реализации проектов в реальном времени на основе модели сети Петри, графа достижимости и применения алгоритма кратчайшего пути для формирования оптимального расписания. Этот алгоритм использует принцип домашней маркировки, который раньше использовался для синхронных систем задач. Авторы показали, что он может быть расширен без модификации на асинхронные системы задач. В данной работе предложен алгоритм оценки эффективности стратегии ресурсного обеспечения проекта, реализованный в рамках развития алгоритмов, расширяющих возможности метода критического пути за счет применения инструментария сетей Петри. Алгоритм работает в двух вариантах. Первый вариант базируется на предположении, что количество ресурсов, выделяемых на каждую работу в проекте является случайной величиной, распределенной

по равномерному закону, второй на предположении, что все ресурсы находятся в едином ресурсном центре и все работы по необходимости обращаются в данный ресурсный центр за ресурсом.

2. МЕТОДЫ ИССЛЕДОВАНИЯ

Сетевое планирование [7] представляет собой комплекс вычислительных методов, позволяющих составить план выполнения взаимосвязанных работ и оценить временные параметры его реализации. В основе сетевого планирования лежит сетевой граф, вершины которого соответствуют событиям, а дуги работам проекта.

Введём следующие обозначения для разных видов работ:

i — индекс начала работы;

j — индекс окончания работы;

t_{ij} — время на выполнение работы (i, j) ;

r_{ij} — ресурс на выполнение работы (i, j) (для активной работы — $t_{ij} > 0$, $r_{ij} > 0$; для работы-ожидания — $t_{ij} > 0$, $r_{ij} > 0$; для работы-логической связи между работами, — $t_{ij} > 0$, $r_{ij} > 0$).

События соответствуют этапу реализации проекта, который заключается в начале и/или завершении одной или нескольких работ проекта.

Дальнейшее изложение материала будем вести на примере следующего строительного проекта, работы которого описаны в табл. 1.

Сетевой граф данного проекта имеет вид, представленный на рис. 1.

Сетевой граф позволяет вычислить основные временные характеристики работ и событий, которые приведены в табл. 2.

Таблица 1. Обозначение, название и средняя длительность работ проекта
[Table 1. Designation, title and average duration of project work]

Обозначение работы	Название работы	Длительность работы
0–1	подготовка строительной площадки	10
0–2	транспортные работы (завоз оборудования для строительства)	11
0–4	оформление документации	9
1–2	установка необходимого оборудования для начала строительства	8
1–3	установка временных сооружений для рабочих	6
2–3	разметка земли	6
2–4	земельные работы	16
3–5	прокладка слоев	6
4–5	канализация (укладка труб, узлов и др.)	11
4–6	бетонные работы	18
5–6	водоснабжение (укладка труб, узлов и др.)	17
5–7	железобетонные конструкции	16
5–8	кладка стен наружных	8
6–7	отопление (укладка труб, узлов и др.)	8
7–8	установка окон, решеток, дверей	17
7–9	электротехническое оборудование	14
8–10	вентиляция	3
8–11	наружные отделочные работы	17
9–10	укладка полов	10
9–11	внутренние отделочные работы	10
10–11	уборка территории	18

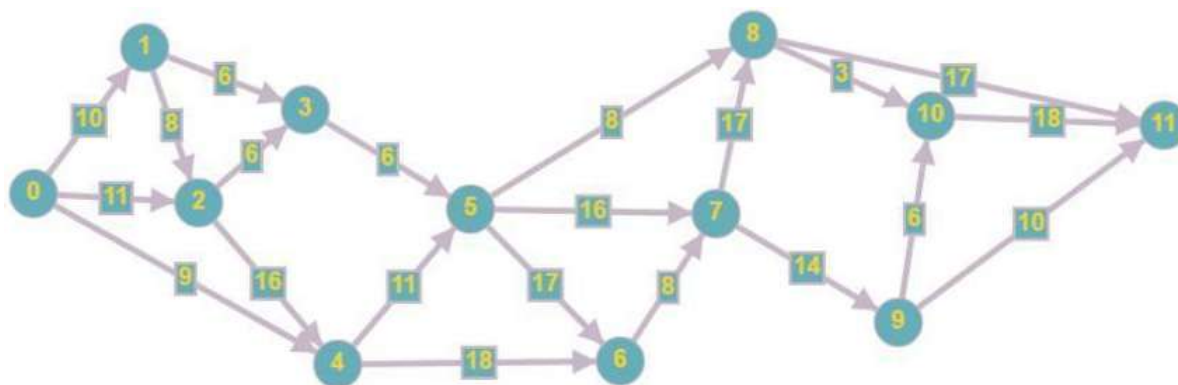


Рис. 1. Сетевой граф строительного проекта
[Fig. 1. Network graph of a construction project]

Таблица 2. Ключевые временные параметры в сетевом графе
[Table 2. Key time parameters in the network graph]

Элемент сети	Наименование параметра	Формальное обозначение параметра
Событие i	Ранний срок свершения события	$t_p(i)$
	Поздний срок свершения события	$t_n(i)$
	Резерв времени события	$R(i)$
Работа (i, j)	Продолжительность работы	t_{ij}
	Ранний срок начала работы	$t_{pn}(i, j)$
	Ранний срок окончания работы	$t_{po}(i, j)$
	Поздний срок начала работы	$t_{nn}(i, j)$
	Поздний срок окончания работы	$t_{no}(i, j)$
	Полный резерв времени работы	$R_n(i, j)$
	Свободный резерв времени	$R_c(i, j)$
Путь L	Продолжительность пути	$T(L)$
	Продолжительность критического пути	$T_{кр}$
	Резерв времени пути	$R(L)$

Вычисления осуществляются по алгоритму нахождения критического пути.

1. Осуществляется топологическая сортировка сетевого графа.

2. Вычисляется ранний срок свершения для каждого события:

$$t_p(j) = \max(t_p(i) + t_{ij}) \quad (1)$$

3. Вычисляется поздний срок свершения для каждого события:

$$t_n(i) = \min(t_n(j) - t_{ij}) \quad (2)$$

4. Вычисляется резерв времени для каждого события:

$$R(i) = t_n(i) - t_p(i). \quad (3)$$

5. Вычисляются ранние и поздние моменты начала и завершения всех работ:

$$t_{pn}(i, j) = t_p(i); \quad (4)$$

$$t_{po}(i, j) = t_p(i) + t_{ij}; \quad (5)$$

$$t_{nn}(i, j) = t_n(j); \quad (6)$$

$$t_{no}(i, j) = t_n(j) - t_{ij}. \quad (7)$$

6. Для каждой из работ производится расчет свободного и полного резервов времени:

$$R_n(i, j) = t_n(j) - t_p(i) - t_{ij}; \quad (8)$$

$$R_c(i) = t_p(j) + t_p(i) - t_{ij}. \quad (9)$$

7. Восстанавливается критический путь.

Для рассматриваемого в работе примера результаты применения алгоритма критического пути представлены на рис. 2.

В условиях риска и неопределенности, когда точно неизвестна длительность каждой

№ Дуги	Начало дуги	Конец дуги	Продолжитель работы	Ранний срок начала	Поздний срок начала	Ранний срок окончания	Поздний срок окончания	Полный резерв времени	Свободный резерв времени
1	0	1	10	0	0	10	10	0	0
2	0	2	11	0	7	11	18	7	7
3	0	4	9	0	25	9	34	25	25
4	1	2	8	10	10	18	18	0	0
5	1	3	6	10	33	16	39	23	8
6	2	3	6	18	33	24	39	15	0
7	2	4	16	18	18	34	34	0	0
8	3	5	6	24	39	30	45	15	15
9	4	5	11	34	34	45	45	0	0
10	4	6	18	34	44	52	62	10	10
11	5	6	17	45	45	62	62	0	0
12	5	7	16	45	54	61	70	9	9
13	5	8	8	45	79	53	87	34	34
14	6	7	8	62	62	70	70	0	0
15	7	8	17	70	70	87	87	0	0
16	7	9	14	70	70	84	84	0	0
17	8	10	3	87	87	90	90	0	0
18	8	11	17	87	91	104	108	4	4
19	9	10	6	84	84	90	90	0	0
20	9	11	10	84	98	94	108	14	14
21	10	11	18	90	90	108	108	0	0

Назад
Построить критический путь

Рис. 2. Результаты алгоритма нахождения критического пути

[Fig. 2. Results of the algorithm for finding the critical path]

работы, которая может непосредственно зависеть от количества ресурсов, выделенных на эту работу, вычисления в данном алгоритме ведутся на основе средних или зафиксированных специальным образом значений длительности работ, а также на основе нечеткого представления информации [13, 14]. В данной работе для оценки параметров проекта в условиях риска и неопределенности предлагается использовать аппарат сетей Петри [8]. Сети Петри представляют собой двудольный ориентированный граф, вершинами которого являются позиции и переходы. Позиции на графе изображаются кружками, а переходы — прямоугольниками или двумя параллельными черточками. Позиции с переходами может связывать одна дуга, или несколько дуг. Дуги, соединяющие две позиции или два перехода, запрещены. Позиции, у которых нет входящих дуг, называются входными, а у которых нет исходящих дуг — выходными. В позициях могут размещаться маркеры активности (метки), способные перемещаться по сети. Размещение меток по позициям сети называется ее размет-

кой. Работа сети Петри начинается с выполнения с начальной разметки.

Простые сети могут быть представлены в виде картежа [9], в котором P — множество позиций, T — множество переходов, E — множество дуг, следующих между переходами и позициями, M_0 — начальное распределение меток, которое описывает число меток в определённой позиции в начальный момент времени:

$$SP = \langle P, T, E, M_0 \rangle, \quad (10)$$

$$M_0 = (m_1^0, \dots, m_N^0).$$

Дополнительно для каждого перехода $t_j, j = 1, \dots, E$ вводится множество выходных и входных позиций $I^0(t_j)$ и $I^1(t_j)$, а для каждой позиции $p_i, i = 1, \dots, N$, множество входных и выходных переходов $J^0(p_i)$ и $J^1(p_i)$ соответственно. Если во всех позициях $I^0(t_j)$ будут маркеры, то переход t_j активизируется, маркеры удаляются из позиций $I^0(t_j)$ и добавляются в позиции $I^1(t_j)$, причем в каждой позиции из множества $I^1(t_j)$ появляется столько маркеров, сколько дуг идет из t_j в данную позицию.

Сети Петри при анализе сетевых проектов используются для исследования процесса изменения состояний анализируемого проекта при известных начальных параметрах его реализации. Работы представляются в виде позиций сети Петри, события в виде ее переходов. С помощью таких сетей можно моделировать стохастические характеристики работ и событий, ситуации возникновения циклов при исполнении работ, проводить учет качества выполнения работ, моделировать перераспределение ресурсов между работами. В данной работе для анализа проектов предлагается использовать цветные временные сети Петри [10]. Временной характер данных сетей Петри отражается в том, что некоторым позициям и переходам может приписываться время задержки. Кроме того, позициям сети ставится в соответствие дополнительно несколько специальных маркеров (цветов): время прихода маркеров активности в данную позицию $S(p_i)$ — начальный временной маркер, время завершения работы в данной позиции $D(p_i)$ — завершающий временной маркер, $T(p_i) = T(\tau_1^i, \tau_2^i, \dots, \tau_L^i)$, время реализации работы в позиции — задержка реализации, которая является функцией от количества ресурсов $(\tau_1^i, \tau_2^i, \dots, \tau_L^i)$ выделенных на выполнение работы, $K(p_i) = K(\tau_1^i, \tau_2^i, \dots, \tau_L^i)$, — качество выполнения работы, которое также является функцией от количества ресурсов $(\tau_1^i, \tau_2^i, \dots, \tau_L^i)$.

Вначале рассмотрим случай, когда в процессе первоначального планирования распределения ресурсов по работам ресурсы не являются детерминированными величинами, в таком случае предполагается, что для каждого ресурса указывается интервал возможных значений для позиции p_i — $\tau_k^i \in [\tau_{k,\min}^i, \tau_{k,\max}^i]$, при этом величина ресурса является случайной величиной, равномерно распределенной на введенном интервале. Функция распределения для ресурса τ_k^i имеет вид:

$$F(x) = \begin{cases} 0, & x < 0 \\ \frac{x - \tau_{k,\min}^i}{\tau_{k,\max}^i - \tau_{k,\min}^i}, & \tau_{k,\min}^i \leq \tau_k^i < \tau_{k,\max}^i \\ 0, & \tau_k^i \geq \tau_{k,\max}^i \end{cases} \quad (11)$$

Остановимся кратко на способах построения функций времени и качества. Функцию $T(r_1, \dots, r_L)$ можно строить или на основе определенных нормативных документов (в частности, в строительстве) или на основе ретроспективной базы, используя методы статистического анализа данных для построения регрессионных моделей или методы машинного обучения. Достаточно часто, предполагается, что функция времени является кусочно-линейной от количества ресурсов:

$$T(r_1, \dots, r_L) = \begin{cases} T_1(r_1, \dots, r_L) \in \Omega_1 \\ T_2(r_1, \dots, r_L) \in \Omega_k \end{cases}, \quad (12)$$

В ряде работ также ставится предположение, что функция времени является выпуклой на выпуклом множестве Ω :

$$\begin{aligned} T(a(r_1, \dots, r_L) + (1-a)(h_1, \dots, h_L)) \geq \\ \geq aT(r_1, \dots, r_L) + (1-a)T(h_1, \dots, h_L), \end{aligned} \quad (13)$$

$$\forall (r_1, \dots, r_L), (h_1, \dots, h_L) \in \Omega$$

В рассматриваемом в рамках данной статьи примере строительного проекта функция времени будет вычисляться как:

$$T(r_1, \dots, r_L) = \frac{T_r}{r_1 \cdot \dots \cdot r_L \cdot k \cdot f}. \quad (14)$$

где T_r — трудоемкость, k — коэффициент качества, f — коэффициент перевода рабочих дней в календарные с учетом отпусков.

Остановимся также кратко на используемом в работе способе вычисления качества выполнения работ $K(r_1, \dots, r_L)$. Под качеством понимается степень удовлетворения требованиям заинтересованных сторон [11], предполагается, что степень удовлетворения требованиям принадлежит промежутку $[0, 1]$, (0 — совсем не удовлетворяет требованиям, 1 — полностью удовлетворяет требованиям). Также предполагается, что требования к качеству формулируются как требования к ресурсам, и качество определенного ресурса зависит от соотношения имеющегося ресурса с базовым, идеальным, ожидаемым количеством $K_l = \varphi(r_l, r_l^{\text{баз}}), K_l = \varphi(r_l, r_l^{\text{ожид}})$, например [11]:

$$K_l = \frac{r_l}{r_l^{\text{баз}}}; \quad (15)$$

$$K_j = \frac{r_l - r_l^{\text{мин}}}{r_l^{\text{баз}} - r_l^{\text{мин}}}; \quad (16)$$

$$K_j = e^{-(r^0)^{m_j}}. \quad (17)$$

где $0 < m_j \leq M$ — положительная константа, $r^0 = \frac{2r_l - (r_l^{\max} + r_l^{\min})}{r_l^{\max} - r_l^{\min}}$ — линейная функция от r_j .

При оценке качества выполнения работы, в зависимости от качества отдельных ресурсов, используется подход Руссмана И. Б. [12], основанный вычислении трудности достижения целей:

$$K(r_1, \dots, r_L) = \prod_{l=1}^L K_l^{\beta_l}. \quad (18)$$

Таким образом, при известных ресурсах, выделенных для выполнения работы, расположенной в определенной позиции можно вычислить маркеры времени $T(r_1, \dots, r_L)$ и качества $K(r_1, \dots, r_L)$. Предполагается, что начальная маркировка сети такова, что единственный маркер находится в начальной позиции p_1 , которая не имеет входящих дуг и соответствует начальной реальной или фиктивной работе.

С учетом введенных предположений и обозначений сформулируем в целом алгоритм оценки реализуемости проекта:

1. Вводится количество повторений алгоритма g . Счетчик количества повторений алгоритма полагается $M = 1$, счетчики несрабатывания переходов полагаются $H_1(t_j) = 0$, $H_2(t_j) = 0$, $j = 1, \dots, E$, счетчик успешного выполнения проекта $R = 0$.

2. Для каждой работы (позиции сети Петри) p_i вводятся интервалы по ресурсам $[\tau_{k,\min}^i, \tau_{k,\max}^i] \forall k = 1, \dots, L$.

3. Вводятся временные характеристики проекта, вычисленные на основании алгоритма нахождения критического пути при средней длительности выполнения работ проекта $T(\tau_1^{i, \text{сред}}, \tau_2^{i, \text{сред}}, \dots, \tau_L^{i, \text{сред}})$, где $\tau_k^{i, \text{сред}}$ — середина интервала $[\tau_{k,\min}^i, \tau_{k,\max}^i] \forall k = 1, \dots, L$.

4. В соответствии с равномерным законом распределения для каждой работы (позиции сети Петри) p_i генерируется вектор ресурсов $\tau_1^i, \tau_2^i, \dots, \tau_L^i$ и вычисляется время

$$T(p_i) = T(\tau_1^i, \tau_2^i, \dots, \tau_L^i)$$

и качество

$$K(p_i) = K(\tau_1^i, \tau_2^i, \dots, \tau_L^i).$$

5. Для начальной позиции p_1 определяются маркеры-переменные (цвета): маркер начала работы $S(p_1) = 0$ и маркер окончания работы $D(p_1) = S(p_1) + T(p_1)$.

6. Для начального перехода t_1 проверяется правило срабатывания:

$$D(p_1) \leq t_n(1); \quad (19)$$

$$K(p_1) \geq \varepsilon_1. \quad (20)$$

где ε_1 — требования к качеству реализации работы в начальной позиции p_1 .

Если хотя бы одно условие не выполнено, то переход не срабатывает, проект не будет выполнен. Если не выполнено первое условие, то счетчик несрабатывания перехода $H_1(t_1) = H_1(t_1) + 1$, если не выполняется второе условие, то $H_2(t_1) = H_2(t_1) + 1$. Счетчик количества повторений алгоритма $M = M + 1$. Если $M > g$, то алгоритм заканчивает работу, иначе осуществляется переход на шаг 4. Если все условия выполнены, то в каждой позиции $I^1(t_1)$ появляются маркеры активности. Время срабатывания перехода $Ts(t_1) = D(p_1)$.

7. Для позиций $I^1(t_1)$, в которых появились маркеры активности:

$$S(p_i) = Ts(t_1), \quad (21)$$

$$D(p_i) = S(p_i) + T(p_i). \quad (22)$$

Проверяются условия:

$$D(p_i) \leq t_n(i), \quad (23)$$

$$K(p_i) \geq \varepsilon_i, \quad (24)$$

где ε_i — требования к качеству реализации работы в позиции p_i . Если хотя бы одно условие не выполнено, то не срабатывает переход t_j , для которого $p_i \in I^1(t_j)$, проект не будет выполнен. Если не выполнено первое условие, то счетчик несрабатывания перехода $H_1(t_1) = H_1(t_1) + 1$, если не выполняется второе условие, то $H_2(t_1) = H_2(t_1) + 1$. Счетчик количества повторения алгоритма $M = M + 1$. Если $M > g$, то алгоритм заканчивает работу, иначе осуществляется переход на шаг 4.

8. Для каждого перехода t_j проверяется наличие маркеров активности в каждой позиции из множества $I^0(t_j)$. Если маркеры активности присутствуют во всех анализируемых позициях, то переход срабатывает, время срабатывания перехода $Ts(t_j) = \max_{p_i \in I^0(t_j)} D(p_i)$. Если $I^1(t_j) \neq \emptyset$, то в каждой позиции $I^1(t_j)$ появ-

ляются маркеры активности. Если $I^1(t_j) = \emptyset$, то делается вывод, что проект успешно выполнен и $R = R + 1$, $M = M + 1$. Если $M > g$, то алгоритм заканчивает работу, иначе осуществляется переход на шаг 4. Если все условия выполнены, то осуществляется переход на шаг 9.

9. Оценивается вероятность успешной реализации проекта:

$$P = \frac{R}{g}. \tag{25}$$

3. РЕЗУЛЬТАТЫ И ИХ ОБСУЖДЕНИЕ

Продemonстрируем работу алгоритма на примере приведенного выше строительного проекта. В качестве недетерминированных ресурсов будем рассматривать только трудовые ресурсы. На рис. 3 приведены данные для работы с недетерминированными ресурсами.

Результаты расчетов по приведенному алгоритму приведены в табл. 3.

Алгоритм был запущен 100000 раз, на каждом этапе генерировались данные по ресурсам. Из 100000 раз проект был успешно выполнен 86136 раз и 13864 раза был не выполнен из-за несрабатывания 6 перехода, ко-

Таблица 3. Результаты работы алгоритма
[Table 3. Results of the algorithm]

Статус проекта	№ условия	№ перехода	Количество повторных случаев
Выполнен			86136
Не выполнен	1		0
Не выполнен	2	6	13864

торое связано с невыполнением требования по качеству одной из работ проекта.

С помощью имитационных механизмов, предложенных сетями Петри, оценивается вероятность срабатывания различных переходов, а, следовательно, вероятность наступления различных событий проекта. Если вероятность несрабатывания переходов будет высокой, выше некоторого порога, то возможно перераспределение ресурсов между работами. Перераспределение заключается в установлении новых плановых интервалов выделения ресурса $((r_1^0 - \Delta r_1, r_1^0 + \Delta r_1), \dots, (r_L^0 - \Delta r_L, r_L^0 + \Delta r_L))$. Ресурсы могут перераспределяться с работ, не лежащих на критическом пути на работы и лежащих на критическом пути.

Non-deterministic project resources

Заполнить из файла

№ Работы	Мин. количество рабочих, чел.	Макс. количество рабочих, чел.	R1баз	Трудоемкость, чел. дней	Мин. коэфф. выполнения работ	Макс. коэфф. выполнения работ
1	6	12	9	36	1	1,3
2	11	17	14	42	1	1,3
3	13	19	17	48	1	1,3
4	31	37	34	340	1	1,3
5	29	35	33	128	1	1,3
6	2	8	8	20	1	1,3
7	30	36	38	468	1	1,3
8	51	57	54	448	1	1,3
9	31	37	46	598	1	1,3
10	25	31	29	525	1	1,3
11	34	40	38	768	1	1,3
12	14	20	18	84	1	1,3
13	16	22	21	74	1	1,3
14	13	19	17	65	1	1,3
15	6	12	10	46	1	1,3
16	16	22	17	210	1	1,3
17	11	17	15	140	1	1,3
18	14	20	18	140	1	1,3
19	22	28	26	288	1	1,3

Очистить

Подтвердить

Рис. 3. Входные данные для работы с недетерминированными ресурсами
[Fig. 3. Input data for working with non-deterministic resources]

Рассмотрим второй вариант алгоритма, в котором возобновляемые ресурсы находятся в едином центре и работы проекта обращаются в этот единый центр за этими ресурсами. После завершения работы ресурсы возвращаются в центр управления для последующего перераспределения на новые работы. Если в центре нет достаточного для работы количества ресурсов, то работа ждет поступления нужного количества ресурсов до тех пор, пока необходимые ресурсы не будут возвращены и доступны для повторного использования. Преимущества этого подхода включают строгий контроль и учет ресурсов, что позволяет точно планировать и отслеживать выполнение проекта. Однако данный метод может привести к задержкам из-за необходимости ожидания возврата ресурсов.

Для данного варианта алгоритма будем использовать сети Петри с дополнительными маркерами. На рис. 4 изображена структура распределения ресурсов из единого центра. Позиция p_i соответствует некоторой работе проекта.

Как только маркер активности приходит в позицию p_i , он переходит в позицию p_{i1} , которая активизирует работу с ресурсами для рассматриваемой работы. В каждую позицию $r_j, j=1, \dots, L$ работы с ресурсами приходит маркер активности от данной работы. Ресурсные позиции имеют специальные массивы $A=(a_{li})$ размерности $2 \times N$. Первая строка данного массива содержит специальные флажки активности, как только маркер активности пришел от позиции p_i , то $a_{li}=l$ (l — приоритет на обслуживание), а соответствующий элемент второй строки этого же столбца становится равным времени прихода маркера активности от позиции p_i . Среди активных позиций находится позиция с самым маленьким значением a_{li} , проверяется условие — достаточно ли ресурса для работы p_i , если да, то ресурсный маркер активности передается в позицию p_{ir} для соответствующего ресурса и соответствующей работы. Когда все ресурсные позиции для данной работы станут активны, то маркер активности передается в позицию p_{i2} , в этой позиции выпол-

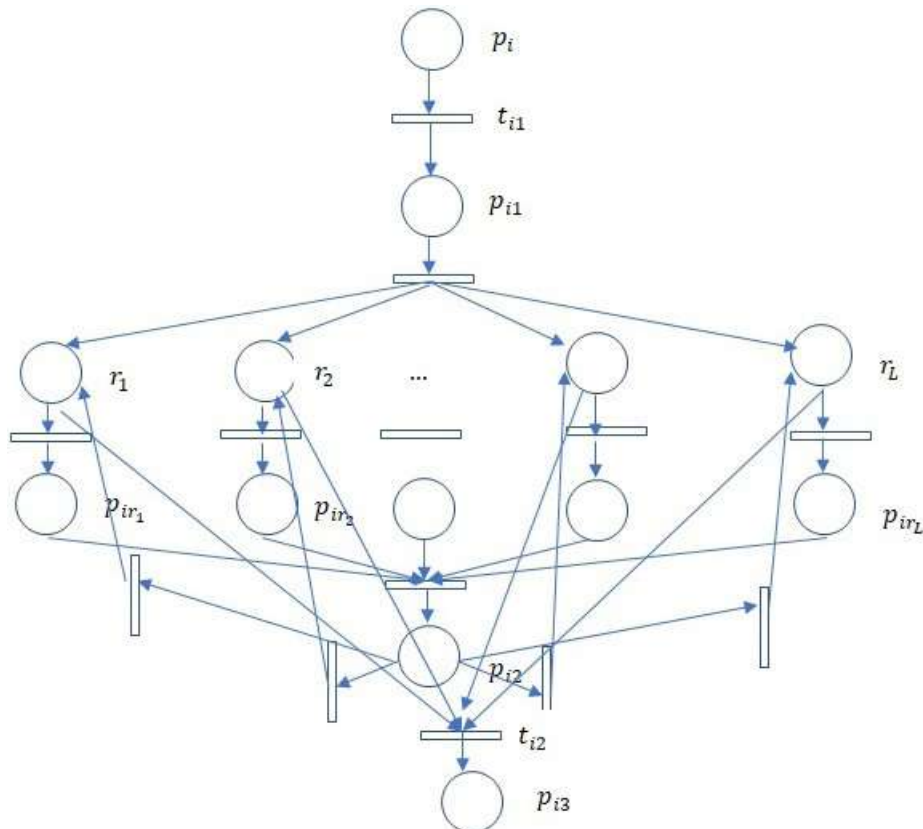


Рис. 4. Структура подсети распределения ресурсов
[Fig. 4. Structure of the resource distribution subnetwork]

няется работа, затем ресурсы возвращаются в соответствующие ресурсные центры. Подробнее весь процесс работы опишем в следующем алгоритме.

1. Для каждой работы (позиции сети Петри) p_i вводится требуемое значение ресурсов $\tau_k^i \forall k=1, \dots, L$.

2. Для всех ресурсных позиций вводится количество ресурсов каждого вида, хранящихся в ресурсном центре (управляющем центре) $RZ_k \forall k=1, \dots, L$. Также для ресурсных позиций вводятся двумерные массивы $A^k = (a_{li}^k)$, для начального момента времени $a_{li}^k = 0, \forall i=1, \dots, N, a_{2i}^k = \infty$ (очень большая константа) $\forall i=1, \dots, N$.

3. Вводятся временные характеристики проекта, вычисленные на основании алгоритма нахождения критического пути при длительности выполнения работ проекта $T(\tau_1^i, \tau_2^i, \dots, \tau_L^i) = T(p_i)$.

4. Для начальной позиции p_1 определяется маркер начала работы $S(p_1) = 0$, маркер активности идет к переходу t_{11} , переход срабатывает и маркер активности переходит в позицию p_{11} , через соответствующий данной позиции переход маркер активности приходят во все ресурсные позиции и заполняются значения $a_{11}^k = 1, a_{21}^k = S(p_1)$.

5. Для каждой ресурсной позиции проверяется условие $\tau_k^1 \leq RZ_k$, если оно выполнено, то ресурсный маркер активности с номером $a_{11}^k = 1$ передается в позицию p_{r_k1} , $RZ_k = RZ_k - \tau_k^1$. В качестве маркера времени перехода данной позиции в активное состояние фиксируется время $Tr_{r_k1} = S(p_1)$, элемент $a_{11}^k = 0$ (процедура с выделением ресурса для данной работы завершена). Если $\tau_k^1 > RZ_k$, то делается вывод, что проект не может быть реализован.

6. Когда во всех ресурсных позициях появляется ресурсные маркеры активности с номером $a_{11}^k = l, l=1$, срабатывает переход и ресурсный маркер активности с номером 1 попадает в позицию p_{12} , начинается процесс выполнения работы 1, время задержки в данной позиции равно $T(p_1)$. После истечения времени задержки срабатывают переходы возврата ресурса. Ресурсные маркеры активности с номером 1 возвращаются в ресурсные позиции. Если элементы $a_{i1}^k \neq 0, \forall i=1, \dots, N$,

то $a_{2i}^k = S(p_1) + T(p_1)$. Время завершения работы $D(p_1) = S(p_1) + T(p_1)$.

7. Для перехода t_1 проверяется правило срабатывания:

$$D(p_1) \leq t_n(1). \quad (26)$$

8. Если условие не выполнено, то переход не срабатывает и проект не будет выполнен, выдается номер перехода, который не сработал. Алгоритм заканчивает работу.

9. Если все условия выполнены, то в каждой позиции $I^1(t_1)$ появляются маркеры активности. Время срабатывания перехода: $Ts(t_1) = D(p_1)$.

10. Для позиций $I^1(t_j)$, в которых появились маркеры активности:

$$S(p_i) = Ts(t_j). \quad (27)$$

11. Маркеры активности идут к переходам t_i , переходы срабатывают и маркеры активности переходят в позиции p_{i1} , через соответствующие данным позициям переходы маркеры активности приходят во все ресурсные позиции, $l := l + 1$, заполняются значения $a_{i1}^k = l, a_{21}^k = S(p_i)$.

12. В первой строке массива A^k каждой ресурсной позиции находится минимальный элемент, запоминается номер столбца i'_k . Для данной позиции проверяется условие $\tau_k^{i'_k} \leq RZ_k$, если оно выполнено, то ресурсный маркер активности с номером i'_k передается в позицию $p_{r_k i'_k}$, $RZ_k := RZ_k - \tau_k^{i'_k}$. В качестве маркера времени перехода данной позиции в активное состояние фиксируется время $Tr_{r_k i'_k} = a_{i'_k 1}^k$. Элемент $a_{i'_k 1}^k = 0$ (процедура с выделением ресурса для данной работы завершена). Если $\tau_k^{i'_k} > RZ_k$, то начинается процесс ожидания возврата ресурса.

13. Когда во всех ресурсных позициях появляются ресурсные маркеры активности с номером i , срабатывает переход, время срабатывания перехода $Tr_{i2} = \max_k Tr_{r_k i}$ и ресурсный маркер активности с номером i попадают в позицию p_{i2} , начинается процесс выполнения работы i , время задержки в данной позиции равно $T(p_i)$. После истечения времени задержки срабатывают переходы возврата ресурса. Ресурсные маркеры активности с номером i возвращаются в ресурсные позиции, если элементы $a_{i1}^k \neq 0, \forall i=1, \dots, N$, то

$a_{2i}^k = T(p_{i2}) + T(p_i)$. Время завершения работы $D(p_i) = a_{2i}^k$.

14. Для перехода t_{i2} проверяется правило срабатывания:

$$D(p_i) \leq t_n(i). \quad (28)$$

15. Если условие не выполнено, то переход не срабатывает, проект не будет выполнен, выдается номер перехода, который не сработал. Алгоритм заканчивает работу.

16. Если условие выполнено, то в позиции p_{i3} появляются маркеры активности и $D(p_{i3}) = D(p_i)$.

17. Для каждого внешнего перехода t_j проверяется наличие маркеров активности в каждой позиции из множества $I^0(t_j)$, состоящего из позиций p_{i3} . Если маркеры активности присутствуют во всех анализируемых позициях, то переход срабатывает, время срабатывания перехода $Ts(t_j) := \max_{p_{i3} \in I^0(t_j)} D(p_{i3})$.

17. Если $I^1(t_j) \neq \emptyset$, в каждой позиции $I^1(t_j)$ появляются маркеры активности и осуществляется переход на шаг 12. Если $I^1(t_j) = \emptyset$, то делается вывод, что проект успешно выполнен.

На рассматриваемом примере алгоритм был запущен в начале с количеством ресурсов в ресурсном центре, равным 54 единицы. По результатам расчетов проект не был выполнен, не сработал внутренний переход на работе 7. При увеличении количества ресурсов в ресурсном центре проект был успешно выполнен. Пошаговое выполнение алгоритма с приращением разных видов ресурса позволит найти минимальное количество ресурсов, при котором проект будет выполнен.

ЗАКЛЮЧЕНИЕ

Применение аппарата сетей Петри к исследованию ресурсного обеспечения проектов позволяет проводить имитационные расчеты при разных стратегиях распределения ресурсов. Подобные инструменты хорошо моделируют неопределенность, связанную с наличием нечеткости и случайности при распределении ресурсов между работами проекта и позволяют воспроизводить процесс движения ресурсов при наличии единого ресурсного

центра. Изложенные в статье алгоритмы доведены программной реализации, что позволило провести экспериментальные расчеты и полноценный анализ результатов. Предложенное алгоритмическое и программное обеспечение может служить эффективным средством поддержки принятия решений при управлении проектами в IT-сфере и в производстве.

КОНФЛИКТ ИНТЕРЕСОВ

Авторы декларируют отсутствие явных и потенциальных конфликтов интересов, связанных с публикацией настоящей статьи.

СПИСОК ЛИТЕРАТУРЫ

1. Margherita E. G. Managing industry 4.0 automation for fair ethical business development: A single case study / E. G. Margherita, A. M. Braccini // *Technological Forecasting and Social Change*. – 2021. – 172. – P. 1–20. doi: 10.1016/j.techfore.2021.121048.
2. Oanea O. A Tool for workflow modeling and analysis / O. Oanea, K. Van Hee, R. Post, L. Somers, J. M. Van der Werf // *In Proceedings of the 6th International Conference on Application of Concurrency to System Design*, 28–30 June 2006, Turku, Finland. IEEE. – P. 279–282.
3. Bashir M. Global Supervisory Structure for Decentralized Systems of Flexible Manufacturing Systems Using Petri Nets / M. Bashir, L. Hong // *Transactions of the Institute of Measurement and Control*. – 2019. – 42, 595. – P. 5–21.
4. Gradisar D. Automated Petri-net modelling based on production management data / D. Gradisar, G. Music // *Mathematical and Computer Modelling of Dynamical Systems*. – 2007. – 13. – P. 267–290.
5. Chang-Pin Lin. Applying Petri Nets on Project Management / Lin Chang-Pin, Dai Hung-Lin // *Universal Journal of Mechanical Engineering*. – 2014. – 2, 8. – P. 249–255.
6. Grolleau E. Optimal off-line exhaustive scheduling using Petri nets in uniprocessor and multiprocessor architectures. LISIENSMA, Université de Poitiers. – Режим доступа: <https://theses.hal.science/tel-04357588v1> (Accessed 29th October 2024).

7. Зенкин А. А. Методы и задачи сетевого планирования: учебное пособие / А. А. Зенкин. – Москва : КНОРУС, 2020. – 206 с.
 8. Lin Y. N. Review on Petri Net Modeling and Analysis of a Smartphone Manufacturing System / Y. N. Lin, T. Y. Hsieh, C. Y. Yang, V. R. L. Shen, T. T. Y. Juang, T. J. Huang, A. Cagiano // *Cogent Engineering*. – 2020. – 7, 1.
 9. Hao M. A workflow modeling optimization and analysis algorithm based on Petri-net / M. Hao // In 2008 International Conference on Management Science and Engineering 15th Annual Conference Proceedings, 10–12 September 2008, Long Beach, USA. IEEE. – P. 188–192.
 10. Murata T. Petri nets: Properties, analysis and applications / T. Murata // In Proceedings of the IEEE. – 1989. – 77, 4. – P. 541–580.
 11. Иванченко А. И. Оценка качества контроля в задачах управления организационными системами / А. И. Иванченко, И. Б. Руссман // *Стандарты и качество*. – 2003. – № 9. – С. 88–90.
 12. Руссман И. Б. Непрерывный контроль процесса достижения цели / А. А. Гайдай, И. Б. Руссман // Институт проблем управления им. В. А. Трапезникова РАН. Управление большими системами. – 2004. – №7. – С. 106–113.
 13. Бондаренко Ю. В. Математические методы поддержки сетевого анализа проекта и оценки риска планирования при нечеткой информации о продолжительностях работ / Ю. В. Бондаренко, Е. В. Васильчикова // *Вестник ВГУ. Серия: Системный анализ и информационные технологии*. – 2023. – № 2. – С. 100–111.
 14. Леденева Т. М. Нечеткая модель проекта с продолжительностями работ в форме обобщенных гауссовых чисел / Т. М. Леденева, Д. А. Черменев // *Вестник ВГУ. Серия: Системный анализ и информационные технологии*. – 2015. – № 2. – С. 72–81.
- Азарнова Татьяна Васильевна** — д-р техн. наук, проф., заведующий кафедрой математических методов исследования операций Воронежского государственного университета.
E-mail: ivdas92@mail.ru
ORCID iD: <https://orcid.org/0000-0001-6342-9355>
- Бондаренко Юлия Валентиновна** — д-р техн. наук, проф., профессор кафедры математических методов исследования операций Воронежского государственного университета.
E-mail: bond.julia@mail.ru
ORCID iD: <https://orcid.org/0000-0002-5442-9123>
- Поляков Николай Аркадьевич** — аспирант 2-го года обучения кафедры математических методов исследования операций Воронежского государственного университета.
E-mail: nickpolyakov123@gmail.com
ORCID iD: <https://orcid.org/0009-0003-9041-1005>
- Медведева Ольга Александровна** — канд. физ.-мат. наук, доц., доцент кафедры вычислительной математики и прикладных информационных технологий Воронежского государственного университета.
E-mail: o_a_medvedeva@mail.ru
ORCID iD: <https://orcid.org/0009-0004-3503-0813>

APPLICATION OF TEMPORARY PETRI NETS TOOLS TO MANAGE PROJECT RESOURCE PARAMETERS

© 2024 T. V. Azarnova✉, Y. V. Bondarenko, N. A. Polyakov, O. A. Medvedeva

Voronezh State University

1, Universitetskaya Square, 394018 Voronezh, Russian Federation

Annotation. The efficiency of the project largely depends on the strategy used to manage the resource parameters of the project tasks, which can be performed sequentially and in parallel. Resource allocation can be deterministic and stochastic, but in practice, projects are usually implemented under conditions of risk and uncertainty, so in the process of planning the resource provision of the project, it is necessary to take into account the risk and uncertainty factors, which can be endogenous (internal organizational) and exogenous (external) in nature. To model the impact of risk and uncertainty factors on the project and to evaluate the efficiency of the resource provision strategy used, it is possible to use the simulation modeling toolkit that meets certain requirements for adaptability and the possibility of implementing feedback. Such a simulation modeling toolkit can be built on the basis of the Petri net apparatus. The purpose of the study is to develop an algorithm for simulation modeling of the project implementation process with a certain resource provision strategy, based on the construction of a model of this process in the form of a temporary Petri net. At the initial stage, a network graph of the project is built and, based on average or fixed values according to some principle, the critical path, critical time of the entire project, and time parameters for each event and each work are calculated. A colored temporary Petri net is built according to the network graph with calculated values of time parameters for the implementation of works, the transitions of markers in which are carried out when a certain block of time and resource constraints is fulfilled. Modeling the process of performing works based on colored temporary Petri nets will allow assessing the efficiency of project resource management.

Keywords: Petri nets, calendar planning, network graph, assessment of the efficiency of project resource allocation.

CONFLICT OF INTEREST

The authors declare the absence of obvious and potential conflicts of interest related to the publication of this article.

REFERENCES

1. Margherita E. G. and Braccini A. M. (2021) Managing industry 4.0 automation for fair ethical business development: A single case study. *Technological Forecasting and Social Change*. No 172. P. 1–20.
2. Oanea O., van Hee K., Post R., Somers L. and van der Werf J. M. (2006) *Yasper: A Tool for*

workflow modeling and analysis. In *Proceedings of the 6th International Conference on Application of Concurrency to System Design*, 28–30 June 2006, Turku, Finland. IEEE. P. 279–282.

3. Bashir M. and Hong L. (2019) Global Supervisory Structure for Decentralized Systems of Flexible Manufacturing Systems Using Petri Nets. *Transactions of the Institute of Measurement and Control*. Vol. 42, No 595. P. 5–21.

4. Gradisar D. and Music G. (2007) Automated Petri-net modeling based on production management data. *Mathematical and Computer Modeling of Dynamical Systems*. No. 13. P. 267–290.

5. Chang-Pin Lin and Hung-Lin Dai. (2014) Applying Petri Nets on Project Management. *Universal Journal of Mechanical Engineering*. Vol. 2, No. 8. P. 249–255.

✉ Azarnova Tatyana V.
e-mail: ivdas92@mail.ru

6. Grolleau E. (2007) Optimal off-line exhaustive scheduling using Petri nets in uniprocessor and multiprocessor architectures. LISIENSMA, Universite de Poitiers. <https://theses.hal.science/tel-04357588v1> (Accessed 29th October 2024).
7. Zenkin A. A. (2020) Methods and tasks of network planning: a tutorial. Moscow : KNORUS, 206 p.
8. Lin Y. N., Hsieh T. Y., Yang C. Y., Shen V. R. L., Juang T. T. Y., Huang T. J. and Caggiano A. (2020) Review on Petri Net Modeling and Analysis of a Smartphone Manufacturing System. *Cogent Engineering*. Vol. 7, No.1.
9. Hao M. (2008) A workflow modeling optimization and analysis algorithm based on Petri-net. In *2008 International Conference on Management Science and Engineering 15th Annual Conference Proceedings*, 10–12 September 2008, Long Beach, USA. IEEE. pp. 188–192.
10. Murata T. (1989) Petri nets: Properties, analysis and applications. In *Proceedings of the IEEE*. Vol. 77, No 4. P. 541–580.
11. Ivanchenko A. I. and Russman I. B. (2003) Evaluation of the quality of control in the tasks of managing organizational systems. *Standards and quality*. No. 9. P. 88–90.
12. Russman I. B. (2004) Continuous control of the goal achievement process. V. A. Trapeznikov Institute of Control Sciences of the Russian Academy of Sciences. *Control of large systems*. No 7. P. 106–113.
13. Bondarenko Yu. V. and Vasilchikova E. V. (2023) Mathematical methods for supporting project network analysis and assessing planning risks with fuzzy information on work durations. *VSU Bulletin. Series: System Analysis and Information Technologies*. 2023. No 2. P. 100–111.
14. Ledeneva T. M. and Chermenev D. A. (2015) Fuzzy project model with work durations in the form of generalized Gaussian numbers *VSU Bulletin. Series: System Analysis and Information Technologies*. No 2. P. 72–81.

Azarnova Tatyana V. — D.Sc. (Eng.), Prof., Head of the Department of Mathematical Methods of Operations Research, Voronezh State University.
E-mail: ivdas92@mail.ru
ORCID iD: <https://orcid.org/0000-0001-6342-9355>

Bondarenko Yulia V. — D.Sc. (Eng.), Prof., Professor of the Department of Mathematical Methods of Operations Research, Voronezh State University.
E-mail: bond.julia@mail.ru
ORCID iD: <https://orcid.org/0000-0002-5442-9123>

Polyakov Nikolay A. — 2nd year postgraduate student of the Department of Mathematical Methods of Operations Research, Voronezh State University.
E-mail: nickpolyakov123@gmail.com
ORCID iD: <https://orcid.org/0009-0003-9041-1005>

Medvedeva Olga A. — Ph.D. of Physico-mathematical Sciences, Docent, Associate Professor at the Department of computational mathematics and applied information technology, Voronezh State University.
E-mail: o_a_medvedeva@mail.ru
ORCID iD: <https://orcid.org/0009-0004-3503-0813>