

MATRIX METHOD FOR FINDING THE PATHS ON WEIGHTED ORIENTED GRAPHS IN THE TASKS OF PORT NET OPERATIONAL PLANNING**A. L. Kuznetsov**Admiral Makarov State University of Maritime and Inland Shipping,
St. Petersburg, Russian Federation

Network planning, or network analysis, is a class of application methods for project management that provides planning, analysis of deadlines (both early and late), risk of project failure or individual parts of the project. These methods allow you to link the performance of different works and processes over time, to make an operational schedule of the project, to get a forecast of the total duration of the project. In the modern practice of designing, building and managing seaports, network planning represents the most sought-after toolkit for decision makers. Network planning methods are conditionally divided into deterministic (Gant's diagrams, rigid and with additional time-lag, critical path method, etc.) and probabilistic. Latter, in turn, are divided into non-alternative (method of statistical tests or the Monte Carlo method, the method of evaluating and revising PERT plans) and alternative (GERT graphic assessment method).

In many applications, the basis of the used method is to find a path on the graph. Multiple repetition of experiments, characteristic of the most effective probabilistic methods, imposes high demands to reduce the computational laboriousness of the used algorithms. In addition, the different nature of cause-and-effect relations between objects of network models leads to the formation of such a structure depicting graph processes, which do not allow the use of most known algorithms. A matrix algorithm for finding paths on weighted oriented graphs, characterized by low computational laboriousness, simplicity and visibility, and allowing different types of cause-effect relations between events, is described in the paper. The proposed algorithm is effective in terms of the set tasks, and its implementation is almost no different from the pseudo-code used to describe it.

This makes it easy to implement, easy to debug and verify code, and easy to embed the algorithm in various network planning applications. One of these tasks is to find critical paths in the context of the time parameters of all works (operations) linking the tops of events.

Keywords: network planning, graph search, simulation techniques.

For citation:

Kuznetsov, Aleksandr L. "Matrix method for finding the paths on weighted oriented graphs in the tasks of port net operational planning." *Vestnik Gosudarstvennogo universiteta morskogo i rechnogo flota imeni admirala S. O. Makarova* 12.2 (2020): 230–238. DOI: 10.21821/2309-5180-2020-12-2-230-238.

УДК 656.615**МАТРИЧНЫЙ МЕТОД ПОИСКА ПУТЕЙ НА ВЗВЕШЕННЫХ ОРИЕНТИРОВАННЫХ ГРАФАХ В ЗАДАЧАХ СЕТЕВОГО ПЛАНИРОВАНИЯ ПРИ ПРОЕКТИРОВАНИИ И ЭКСПЛУАТАЦИИ МОРСКИХ ПОРТОВ****А. А. Кузнецов**ФГБОУ ВО «ГУМРФ имени адмирала С.О. Макарова»,
Санкт-Петербург, Российская Федерация

Отмечается, что сетевое планирование, или сетевой анализ, представляет собой класс прикладных методов управления проектами, обеспечивающих планирование, анализ сроков выполнения (как ранних, так и поздних), риска невыполнения проекта или его отдельных частей. Данные методы позволяют увязать выполнение различных работ и процессов во времени, составить операционный график выполнения проекта, получить прогноз общей продолжительности реализации всего проекта. В современной практике проектирования, строительства и управления морскими портами сетевое планирование представляет наиболее востребованный инструментальный лиц, принимающих решение. Методы сетевого планирования условно подразделяются на детерминированные (диаграммы Ганта, жесткие и с дополнительным временным люфтом, метод критического пути и др.) и вероятностные. Последние, в свою очередь, делят-

ся на неальтернативные (метод статистических испытаний или метод Монте-Карло, метод оценки и пересмотра планов PERT) и альтернативные (метод графической оценки и анализа GERT). Во многих приложениях основу используемого метода составляет поиск пути на графе. Многократное повторение экспериментов, характерное для наиболее эффективных вероятностных методов, предъявляет высокие требования к снижению вычислительной трудоемкости используемых алгоритмов. Кроме того, различный характер причинно-следственных связей между объектами сетевых моделей приводит к формированию такой структуры изображающего процессы графа, которые не позволяют применять большинство известных алгоритмов.

В данной статье описывается матричный алгоритм поиска путей на взвешенных ориентированных графах, отличающийся низкой вычислительной трудоемкостью, простотой и наглядностью, а также допускающий различные виды причинно-следственных связей между составными событиями. Предложенный алгоритм является результативным в отношении поставленных задач, а его реализация практически не отличается от псевдокода, использованного для его описания, что обеспечивает легкость реализации, простоту отладки и верификации кода, легкость встраивания алгоритма в различные прикладные задачи сетевого планирования. Одной из таких задач является нахождение критических путей в условиях разброса временных параметров всех работ (операций), связывающих между собой вершины-события.

Ключевые слова: сетевое планирование, поиск путей на графе, методы имитационного моделирования.

Для цитирования:

Кузнецов А. Л. Матричный метод поиска путей на взвешенных ориентированных графах в задачах сетевого планирования при проектировании и эксплуатации морских портов / А. Л. Кузнецов // Вестник Государственного университета морского и речного флота имени адмирала С. О. Макарова. — 2020. — Т. 12. — № 2. — С. 230–238. DOI: 10.21821/2309-5180-2020-12-2-230-238.

Введение (Introduction)

Ключевой целью сетевого планирования в управлении является сокращение продолжительности проекта при условии сохранения параметров качества и объема продукции. Сложность, комплексность и капиталоемкость задач проектирования, строительства и управления развитием морских портов делает данный научный инструмент одним из основных средств при создании этих важнейших элементов транспортной инфраструктуры.

Методики сетевого планирования появились впервые в 50-е гг. в США [1]. В 1956 г. поиски наиболее эффективного использования вычислительной машины Univac привели к привлечению ее для составления планов-графиков крупных комплексов работ по модернизации заводов фирмы «Дюпон». Результатом исследований явился рациональный и простой метод описания проекта, который получил название *метода критического пути* — МКП или CPM (*Critical Path Method*) [2], [3].

Параллельно в научных лабораториях ВМС США появился *метод анализа и оценки программ* — PERT (*Program Evaluation and Review Technique*), использованный при реализации проекта разработки ракетной системы «Поларис» [4], [5]. Руководство программой с использованием нового средства оказалось настолько успешным, что проект был завершен на два года раньше запланированного срока. Как следствие, этот метод управления начал использоваться для планирования всех проектов, выполняемых вооруженными силами США [6], [7]. Вскоре подобную методику начали применять и крупные гражданские промышленные корпорации [8], [9]. По существу, значительный выигрыш по времени реализации проектов явился результатом применения математических методов управления сложными комплексами работ, что стало возможным лишь благодаря развитию вычислительной техники.

Первоначально крупные корпорации осуществляли самостоятельную разработку программного обеспечения, но вскоре первые системы управления проектами появились и на рынке публичного программного обеспечения. Бурное развитие систем управления проектами началось с появлением персональных компьютеров, быстро ставших рабочим инструментом широкого круга руководителей. Резкое расширение круга применения обусловило потребность в создании систем управления проектами нового типа, в которых важнейшим показателем являлись

простота использования. Управленческие системы нового поколения стали восприниматься как средство управления проектом, понятное любому менеджеру, не требующее специальной подготовки, обеспечивающее простое «включение» в работу. С расширением круга пользователей систем проектного менеджмента происходит развитие методических приемов их использования. Отечественные и зарубежные отраслевые журналы регулярно публикуют статьи, посвященные системам управления проектами, содержащие советы пользователям таких систем и анализ использования методики сетевого планирования для решения задач в различных сферах управления.

Применение методологии сетевого планирования в управлении требует структуризации операций по отрезкам времени и группировки последовательно-параллельных операций для оптимизации хода выполнения всего проекта. На основе этого осуществляется построение схемы, в которую сводятся все значимые данные по каждой операции в соответствии с их формализованной структурой и группами параллельных операций. Результатом является построение сетевой диаграммы, дающей возможность системного представления всех операций и создающих условия для управления процессом реализации проекта.

Сетевое планирование ориентировано в первую очередь на минимизацию продолжительности проекта с применением двух основных методов: *метода критического пути* и *метода пересмотра планов*. Критическим называется максимальный по продолжительности полный путь в сети; работы, лежащие на этом пути, также называются *критическими*. Длительность критического пути определяет наименьшую общую продолжительность работ по всему проекту [10]. Метод оценки и пересмотра планов заключается в обеспечении соблюдения графиков выполнения работ, составляющих проект. Для этого осуществляется его декомпозиция на ряд подзадач, для каждой из которых назначается время и приоритет ее выполнения. В зависимости от приоритета задач и их влияния на весь проект предпринимаются необходимые действия для снижения продолжительности проекта. Во всех случаях основу большинства методов и их вариантов составляют базовые алгоритмы нахождения кратчайших путей во взвешенных графах. К их числу относятся алгоритмы Флойда – Уоршелла [11], Форда – Беллмана [12], классический алгоритм Дейкстры [13], алгоритм Дейкстры для разреженных графов [14], [15].

В то же время указанные алгоритмы отличаются достаточно высокой вычислительной трудоемкостью, сложностью программной реализации и невозможностью работы с различными типами причинно-следственных связей. В данной публикации предлагается описание общей идеи, реализация и примеры использования матричного алгоритма поиска пути на взвешенных графах произвольной природы, который представляется свободным от части этих недостатков.

Методы и материалы (Methods and Materials)

Сетевая диаграмма проектных событий и работ, называемых в портовых проектах *операциями*, обычно представляется в виде ориентированного графа, вершинами которого являются события, а дугами — элементарные операции, выполнение которых связывает предшествующее событие с последующим. У каждого события может быть несколько «предшественников», которые могут приводить к наступлению данного события по разным сценариям: И / ИЛИ. Событие, связанное с предшествующими по ИЛИ, случается тогда, когда выполняется одна из предшествующих операций. Событие, связанное с предшествующими по И, случается тогда, когда выполняются все предшествующие операции. Примером условия И может быть процедура подготовки складской бригады к загрузке контейнера: бригада должна осмотреть контейнер и сверить данные о грузе с документами (рис. 1).

Примером условия ИЛИ может служить процедура погрузки контейнера на автомобиль: контейнер будет погружен на автомобиль RTG или RS в зависимости от того, какая из машин окажется ближе к соответствующей секции (рис. 2).

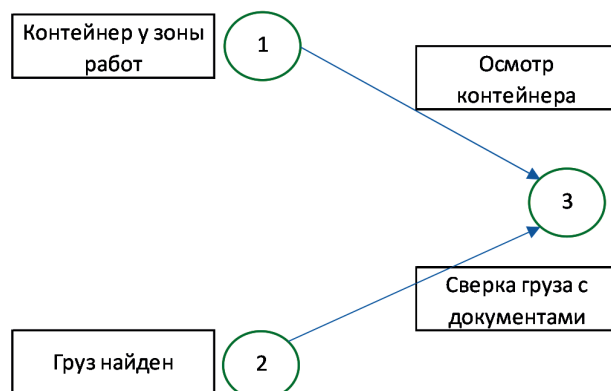


Рис. 1. Пример условия И:
подготовка бригады к началу операции

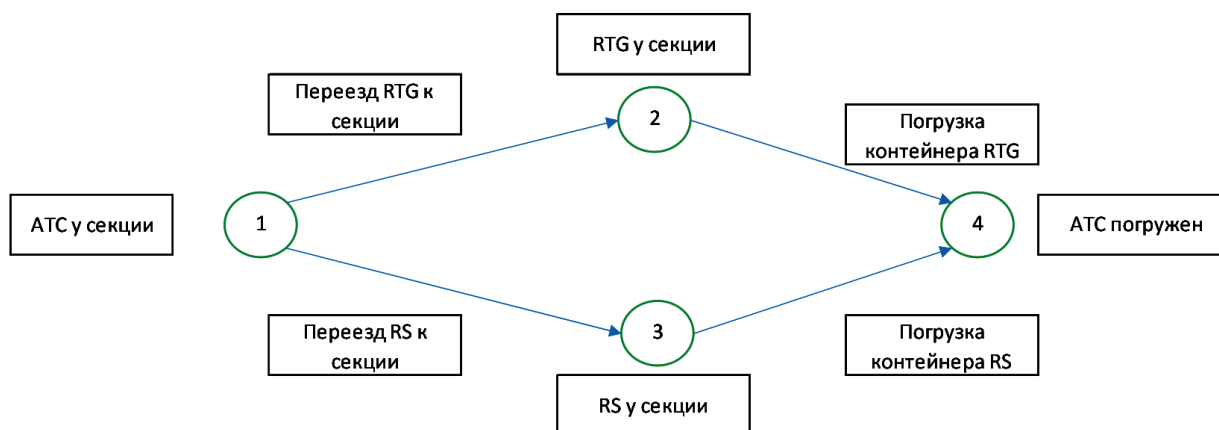


Рис. 2. Пример условия ИЛИ:
погрузка автомобиля

Каждое свершившееся событие характеризуется временем наступления. Каждая операция, связывающая событие-предшественник с событием-последователем, характеризуется длительностью. Событие-последователь в случае ИЛИИ наступает в самый ранний момент времени, определяемый любым событием-предшественником. Событие-последователь в случае И наступает в самый поздний момент времени, определяемый всеми событиями-предшественниками. До начала отдельного эксперимента проводится генерация случайных значений длительности всех операций, связывающих между собой все события. В начальный момент времени свершившимися считаются одно или несколько событий. Далее для всех остальных событий проверяется выполнение связей по схеме И или ИЛИИ и результирующим событиям присваивается статус свершившихся в момент времени, определяемый наступлением событий-предшественников с учетом длительности операции и схем И или ИЛИИ.

Самое раннее событие по схеме ИЛИИ или самое позднее по схеме И фиксируется для каждого нового свершившегося события. Если достигнута конечная вершина, то процесс прямого распространения считается законченным, в противном случае этот процесс повторяется. После завершения процесса прямого распространения по сформированным ссылкам выполняется обратный ход для нахождения критического пути при данном наборе значений длительности операций.

Результаты (Results)

Пусть ориентированный граф последовательности операций представлен K событиями. Наличие направленного ребра от вершины i к j обозначим как r_{ij} . Вершина e_i в этом случае называется *предшествующей для вершины e_j* , а вершина e_j — *последующей для вершины e_i* .

Граф может быть полностью определен несколькими способами: графически, матрично, списком. Матрицей смежности ориентированного графа называется квадратная таблица с ненулевым элементом r_{ij} в случае, если имеется направленное ребро от вершины i к j , т. е. $R = r_{ij}$, и нулевым элементом в противоположном случае. Например, для показанного на рис. 3 направленного графа операций матрица смежности будет выглядеть в виде, приведенном на рис. 4.

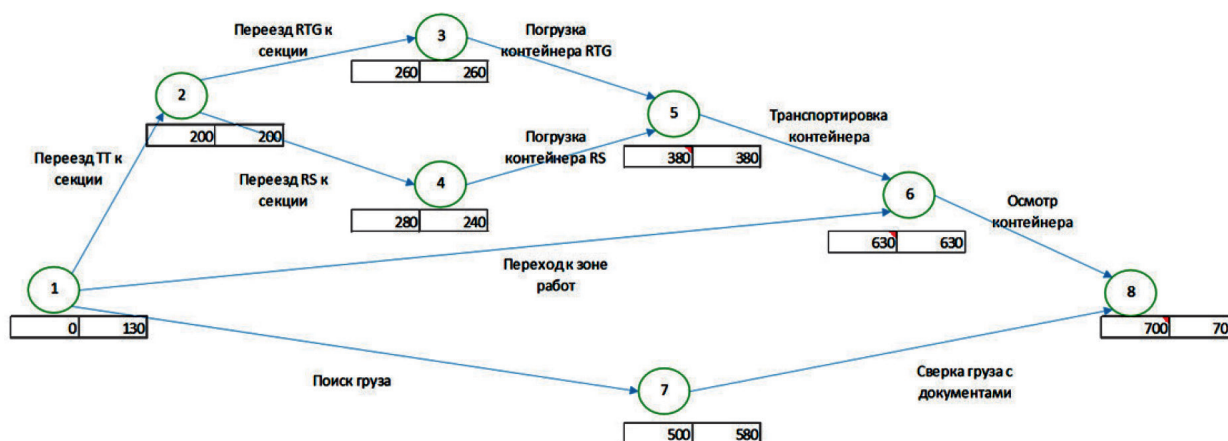


Рис. 3. Граф операций

	j	1	2	3	4	5	6	7	8
i	1		1	0	0	0	1	1	0
	2	0	0	1	1	0	0	0	0
	3	0	0	0	0	1	0	0	0
	4	0	0	0	0	1	0	0	0
	5	0	0	0	0	0	1	0	0
	6	0	0	0	0	0	0	0	1
	7	0	0	0	0	0	0	0	1
	8	0	0	0	0	0	0	0	0

Рис. 4. Матрица смежности графа операций

Причинно-следственные связи между элементарными событиями представлены двумя квадратными матрицами связи: по И и по ИЛИ. По сути, они являются матрицами смежности: каждая из них показывает, что для наступления события e_j необходимо наступление тех или иных предшествующих событий r_{ij} .

В случае матрицы смежности типа ИЛИ для наступления события e_j достаточно наступления хотя бы одного предшествующего события. Обозначим эту матрицу как $O = o_{ij}$. В случае матрицы смежности типа И для наступления события e_j необходимо наступление всех предшествующих событий. Обозначим эту матрицу как $A = a_{ij}$. Очевидно, что указанные матрицы являются разными интерпретациями одного и того же представления матрицы связности графа $R = r_{ij}$.

Пусть в начальный момент времени состояние событий описывается вектором-строкой e_{j0} , $j = \overline{1, K}$; $e_j \in \{0, 1\}$. Если $e_j = 1$, то соответствующее событие считается случившимся. В следующую итерацию логического вывода для матрицы ИЛИ состояние событий будет определяться новым вектором-строкой e_{j1} , в котором фиксируемое наступление событий будет определяться состояни-

ем событий предшествующей итерации e_{j0} и их связями с текущими событиями. Для того, чтобы получить эту связь, введем операцию логического произведения матриц и векторов в виде их частного случая. Логическим произведением двух матриц: $X = x_{ij}$ и $Y = y_{ij}$, будем называть новую матрицу $Z = z_{ij}$, элементы которой определяются как $z_{ij} = \bigcup_{m=1}^K x_{im} \cap y_{mj}$.

Пусть теперь в начальный момент времени состояние событий описывается вектором-строкой e_{j0} : $j = \overline{1, K}$; $e_j \in 0, 1$ и если $e_j = 1$, то соответствующее событие считается случившимся. Рассмотрим произведение слева от исходного вектора-строки e_{j0} на матрицу смежности, или $e_j = e_{j0} \times o_{ij}$. По введенному ранее определению операции логического произведения матриц и векторов запишем $\|e_j\| = \left\| \bigcup_{m=1}^K e_m \cap r_{mj} \right\|$. Отдельный член $e_m \cap r_{mj}$ в случае истинности показывает, что событию e_j в данной итерации предшествует свершившееся событие e_m , связанное с событием e_j причинно-следственной связью r_{mj} , т. е. являющееся его свершившимся предшественником. Полученный вектор показывает события, состоявшиеся на этой итерации. Для регистрации текущего состояния их надо объединить с ранее наступившими событиями, т. е. $\|e_j\|_1 = \left\| \bigcup_{m=1}^K e_m \cap r_{mj} \right\|_1 \cup \|e_j\|_0$. Аналогично каждый следующий шаг итерации будет описываться как $e_{jn} = e_{jn-1} \times o_{ij} \cup e_{jn-1}$. Поскольку максимальный по длине путь на графе не превышает полного числа вершин K , путь в конечную вершину будет найден как число итераций, не превышающих этого значения.

На рис. 5, а показаны соответствующие вектора-строки состояний для приведенного примера и интерпретации графа по схеме ИЛИ, откуда видно, что конечное событие 8 наступает после выполнения второй итерации, после чего состояния событий перестают меняться. На рис. 5, б показаны соответствующие вектора-строки состояний для приведенного примера, но входящие ребра интерпретируются по схеме И, т. е. требуют для наступления выполнения не одного, а всех событий-предшественников. В этом случае конечное событие 8 наступает после выполнения пятой итерации.

а)

ИЛИ	1	2	3	4	5	6	7	8
0	1	0	0	0	0	0	0	0
1	1	1	0	0	0	1	1	0
2	1	1	1	1	0	1	1	1
3	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1

б)

И	1	2	3	4	5	6	7	8
0	1	0	0	0	0	0	0	0
1	1	1	0	0	0	0	1	0
2	1	1	1	1	0	0	1	0
3	1	1	1	1	1	0	1	0
4	1	1	1	1	1	1	1	0
5	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1

Рис. 5. Векторы состояний:

а — для итераций по схеме ИЛИ; б — для итераций по схеме И

Вообще, умножение единичного вектора-строки на матрицу смежности не по логическим правилам, а по правилам обычных матричных операций дает вектор-строку количества предшествующих событий (см. рис. 6, а). Умножение полученного на каждой итерации вектора-строки текущего состояния на матрицу смежности по обычным правилам умножения матриц дает количество событий, наступивших к этому моменту (рис. 6, б). Как видно из данного рисунка, текущее состояние задачи на пятой итерации совпадает с исходным вектором числа предшествующих событий, что свидетельствует о выполнении всех условий наступления событий как по схеме ИЛИ, так и по схеме И.

а)

Вектор числа предшественников							
0	1	1	1	2	2	1	2

б)

	Вектор числа предшественников							
	0	1	1	1	2	2	1	2
	1	2	3	4	5	6	7	8
1	0	1	0	0	0	1	1	0
2	0	1	1	1	0	1	1	1
3	0	1	1	1	2	1	1	1
4	0	1	1	1	2	2	1	1
5	0	1	1	1	2	2	1	2
6	0	1	1	1	2	2	1	2

Рис. 6. Вектор количества предшествующих событий (а);
вектор количества событий, наступивших к определенному моменту (б)

Наступление нового события e_j на каждой итерации позволяет фиксировать момент его наступления T_j : при использовании матрицы смежности ИЛИ таким моментом является самый ранний момент появления предпосылки к его наступлению. Этим моментом является минимальная сумма момента наступления T_i некоторого события-предшественника e_j и длительности реализации причинно-следственной связи r_{ij} , которую будем обозначать t_{ij} . Иными словами, имеем $T_j = \min_i (T_i + t_{ij})$. Моменты наступления событий фиксируются в матрице T (рис. 7, а).

а)

ИЛИ	1	2	3	4	5	6	7	8
0	100	0	0	0	0	0	0	0
1	100	120	0	0	0	155	144	0
2	100	120	150	160	0	155	144	166
3	100	120	150	160	167	155	144	166
4	100	120	150	160	167	155	144	166

б)

И	1	2	3	4	5	6	7	8
0	100	0	0	0	0	0	0	0
1	100	120	0	0	0	0	144	0
2	100	120	150	160	0	0	144	0
3	100	120	150	160	187	0	144	0
4	100	120	150	160	187	231	144	0
5	100	120	150	160	187	231	144	264
6	100	120	150	160	187	231	144	264

Рис. 7. Матрица моментов наступления событий:
а — по ИЛИ; б — по И

При использовании матрицы смежности типа И моментом наступления события будет являться не самый ранний, а самый поздний момент наступления события из всей комбинации требуемых для наступления события предшественников, т. е. $T_j = \max_i (T_i + t_{ij})$. Моменты наступления событий фиксируются в матрице T (рис. 7, б).

Путь по ИЛИ	Путь по И
1-7-8	1-2-4-5-6-8

Рис. 8. Результаты поиска пути по алгоритмам для матриц ИЛИ и И

В результате на каждом шаге итерационного процесса текущее состояние событий e_{kn} дополняется вектором-моментом их наступления T_{kn} . Кроме того, послужившее «реальным активатором» событие — самое раннее для матрицы ИЛИ и самое позднее для матрицы И — записывается в строку следа s_{kn} для последующего построения критического пути из конечной в начальную точку. На рис. 8 показаны найденные для матриц смежности разного типа критические пути.

Заключение (Conclusion)

Предложенный алгоритм действительно является результативным в отношении поставленных задач, а его реализация практически не отличается от псевдокода, использованного для его описания. Это обстоятельство обеспечивает легкость реализации, простоту отладки и верификации кода, легкость встраивания алгоритма в различные прикладные задачи сетевого планирования. Одной из таких задач является нахождение критических путей в условиях разброса временных параметров всех работ (операций), связывающих между собой вершины-события. Для получения статистически достоверных данных о критических путях (возможно, различных при различных сочетаниях длительностей выполнения работ) и формирования надежных вероятностных оценок их параметров требуется проведение обширных объемов испытаний, в которых низкая вычислительная трудоемкость предложенного метода может оказаться полезной.

СПИСОК ЛИТЕРАТУРЫ

1. Кудрявцев Е. М. Project 2003. Сетевое планирование и управление проектами / Е. М. Кудрявцев. — М.: ДМК Пресс, 2014. — 240 с.
2. Woolf M. B. CPM Mechanics: The Critical Path Method of Modeling Project Execution Strategy / M.B. Woolf. — ICS-Publications, 2012. — 480 p.
3. Kelley Jr J. E. Critical-path planning and scheduling / J. E. Kelley Jr, M. R. Walker // IRE-AIEE-ACM '59 (Eastern). — New York, NY, United States: Association for Computing Machinery, 1959. — С. 160–173. DOI: 10.1145/1460299.1460318.
4. Burgelman J. Computing project makespan distributions: Markovian PERT networks revisited / J. Burgelman, M. Vanhoucke // Computers & Operations Research. — 2019. — Vol. 103. — Pp. 123–133. DOI: 10.1016/j.cor.2018.10.017.
5. Newell M. The Project Management Question and Answer Book / M. Newell, M. Grashina. — American Management Association, 2003. — 98 p.
6. Washburn A. Military operations research / A. Washburn // Handbooks in Operations Research and Management Science. — 1994. — Vol. 6. — Pp. 67–106. DOI: 10.1016/S0927-0507(05)80085-4.
7. Thayer H. Management of the Hanford Engineer Works in World War II: How the Corps, DuPont and the Metallurgical Laboratory fast tracked the original plutonium works / H. Thayer. — ASCE Press, 1996. — 224 p. DOI: 10.1061/9780784401606.
8. Ali I. Isolating critical flow path and algorithmic partitioning of the AND/OR mobile workflow graph / I. Ali, S. Bagchi // Future Generation Computer Systems. — 2020. — Vol. 103. — Pp. 28–43. DOI: 10.1016/j.future.2019.09.059.
9. Burgelman J. Computing project makespan distributions: Markovian PERT networks revisited / J. Burgelman, M. Vanhoucke // Computers & Operations Research. — 2019. — Vol. 103. — Pp. 123–133. DOI: 10.1016/j.cor.2018.10.017.

10. Baker S. L. Critical Path Method (CPM) / S. L. Baker. — University of South Carolina, Dept. of Health Services Policy and Management Courses and Curricula, HSPM J716, 2004.
11. Bellman R. On a routing problem / R. Bellman // Quarterly of applied mathematics. — 1958. — Vol. 16. — Is. 1. — Pp. 87–90. DOI: 10.1090/qam/102435.
12. Ford Jr. L. R. Flows in Networks / L. R. Ford Jr., D. R. Fulkerson. — Princeton University Press, 1962. — 212 p.
13. Dijkstra E. W. A note on two problems in connexion with graphs / E. W. Dijkstra // Numerische mathematik. — 1959. — Vol. 1. — Pp. 269–271. DOI: 10.1007/BF01386390.
14. Кормен Т. Х. Алгоритмы: построение и анализ / Т. Х. Кормен, Ч. И. Лейзерсон, Р. Л. Ривест, К. Штайн. — 2-е изд. — М.: «Вильямс», 2006. — 1296 с.
15. Левитин А. В. Жадные методы: Алгоритм Дейкстры / А. В. Левитин // Алгоритмы. Введение в разработку и анализ. — М.: Вильямс, 2006. — С. 189–195.

REFERENCES

1. Kudryavtsev, E. M. *Project 2003. Setevoe planirovanie i upravlenie proektami*. M.: DMK Press, 2014.
2. Woolf, Murray B. *CPM Mechanics: The Critical Path Method of Modeling Project Execution Strategy*. ICS-Publications, 2012.
3. Kelley Jr, James E., and Morgan R. Walker. “Critical-path planning and scheduling.” *IRE-AIEE-ACM '59 (Eastern)*. New York, NY, United States: Association for Computing Machinery, 1959. 160–173. DOI: 10.1145/1460299.1460318.
4. Burgelman, Jeroen, and Mario Vanhoucke. “Computing project makespan distributions: Markovian PERT networks revisited.” *Computers & Operations Research* 103 (2019): 123–133. DOI: 10.1016/j.cor.2018.10.017.
5. Newell, Michael, and Marina Grashina. *The Project Management Question and Answer Book*. American Management Association, 2003.
6. Washburn, Alan. “Military operations research.” *Handbooks in Operations Research and Management Science* 6 (1994): 67–106. DOI: 10.1016/S0927-0507(05)80085-4
7. Thayer, Harry. *Management of the Hanford Engineer Works in World War II: How the Corps, Dupont and the Metallurgical Laboratory fast tracked the original plutonium works*. ASCE Press, 1996. DOI: 10.1061/9780784401606.
8. Ali, Ihtisham, and Susmit Bagchi. “Isolating critical flow path and algorithmic partitioning of the AND/OR mobile workflow graph.” *Future Generation Computer Systems* 103 (2020): 28–43. DOI: 10.1016/j.future.2019.09.059.
9. Burgelman, Jeroen, and Mario Vanhoucke. “Computing project makespan distributions: Markovian PERT networks revisited.” *Computers & Operations Research* 103 (2019): 123–133. DOI: 10.1016/j.cor.2018.10.017.
10. Baker, Samuel L. *Critical Path Method (CPM)*. University of South Carolina, Dept. of Health Services Policy and Management Courses and Curricula, HSPM J716, 2004.
11. Bellman, Richard. “On a routing problem.” *Quarterly of applied mathematics* 16.1 (1958): 87–90. DOI: 10.1090/qam/102435.
12. Ford Jr., Lester Randolph, and D. R. Fulkerson. *Flows in Networks*. Princeton University Press, 1962.
13. Dijkstra, Edsger W. “A note on two problems in connexion with graphs.” *Numerische mathematik* 1 (1959): 269–271. DOI: 10.1007/BF01386390.
14. Kormen, Tomas Kh., Charl’z I. Leizeron, Ronal’d L. Rivest, and Klifford Shtain. *Algoritmy: postroenie i analiz*. 2nd ed. M.: «Vil’yams», 2006.
15. Levitin, A. V. “Zhadnye metody: Algoritm Deikstry.” *Algoritmy. Vvedenie v razrabotku i analiz*. M.: Vil’yams, 2006. 189–195.

ИНФОРМАЦИЯ ОБ АВТОРЕ

Кузнецов Александр Львович —
доктор технических наук, профессор
ФГБОУ ВО «ГУМРФ имени
адмирала С. О. Макарова»
198035, Российская Федерация, Санкт-Петербург,
ул. Двинская 5/7
e-mail: thunder1950@yandex.ru, kaf_pgt@gumrf.ru

INFORMATION ABOUT THE AUTHOR

Kuznetsov, Aleksandr L. —
Dr. of Technical Sciences, professor
Admiral Makarov State University of Maritime
and Inland Shipping
5/7 Dvinskaya Str., St. Petersburg, 198035,
Russian Federation
e-mail: thunder1950@yandex.ru, kaf_pgt@gumrf.ru

Статья поступила в редакцию 06 февраля 2020 г.
Received: February 6, 2020.