

УДК 519.85

DOI: [10.26102/2310-6018/2022.38.3.007](https://doi.org/10.26102/2310-6018/2022.38.3.007)

Модифицированный генетический алгоритм построения расписания проекта

В.В. Коротков 

*Воронежский государственный университет,
Воронеж, Российская Федерация
korotkov@cs.vsu.ru *

Резюме. В работе описывается модифицированный генетический алгоритм решения задачи составления расписания выполнения проекта с учетом ресурсных ограничений. Актуальность исследования обусловлена обширной распространенностью проектной формы организации деятельности и крайне высокой вычислительной сложностью рассматриваемой задачи, что требует дальнейшего улучшения существующих эвристических алгоритмов для возможности эффективного планирования крупных проектов. Имеющиеся генетические алгоритмы основаны на методах кодирования порядка работ и реализациях генетических операторов, которые слабо учитывают особенности задачи. В связи с этим предлагается использование альтернативного метода кодирования и соответствующего оператора кроссовера, которые, в отличие от классических подходов, позволяют вычленять в качестве наследуемых признаков относительные, а не абсолютные положения работ. В работе рассматриваются основные свойства подобного кодирования, представимого в виде квадратных булевых матриц. Представлен оператор отображения, позволяющий привести булевы матрицы к каноническому строчному виду. Было проведено сравнение полученного генетического алгоритма и классических реализаций на тестовой выборке задач. Предложенные подходы продемонстрировали потенциальную эффективность, особенно при планировании крупных проектов. Результаты работы могут представлять практическую ценность при разработке систем поддержки принятия решений в проектном менеджменте.

Ключевые слова: генетический алгоритм, оператор кроссовера, планирование проекта, комбинаторная оптимизация, теория расписаний.

Для цитирования: Коротков В.В. Модифицированный генетический алгоритм построения расписания проекта. *Моделирование, оптимизация и информационные технологии.* 2022;10(3). Доступно по: <https://moitvvt.ru/ru/journal/pdf?id=1214> DOI: 10.26102/2310-6018/2022.38.3.007

Modified genetic algorithm for project scheduling

V.V. Korotkov 

*Voronezh State University,
Voronezh, Russian Federation
korotkov@cs.vsu.ru *

Abstract. The paper describes a modified genetic algorithm for solving resource-constrained project scheduling problem. The relevance of the study is due to the widespread prevalence of project organization of activities and the extremely high computational complexity of the problem under consideration. Further improvement of existing heuristic algorithms is needed to enable efficient planning of large projects. The available genetic algorithms are based on activity order encoding methods and implementations of genetic operators, which does not fully take into account the specifics of the problem. Therefore, the paper proposes an alternative encoding method and the corresponding crossover operator, which, unlike classical approaches, highlights relative rather than absolute positions of activities as inherited features. The study regards the main properties of such encoding which can be

represented as square Boolean matrices. A mapping operator that helps to reduce Boolean matrices to a canonical row form is also introduced. The resulting genetic algorithm and classical implementations were compared using a test set of tasks. The suggested approach has shown potential efficiency, especially with large projects. The findings can be of practical importance in the development of decision support systems for project management.

Keywords: genetic algorithm, crossover operator, project planning, combinatorial optimization, scheduling theory.

For citation: Korotkov V.V. Modified genetic algorithm for project scheduling. *Modeling, Optimization and Information Technology*. 2022;10(3). Available from: <https://moitvvt.ru/journal/pdf?id=1214> DOI: 10.26102/2310-6018/2022.38.3.007 (In Russ.).

Введение

Задача планирования проекта в условиях ресурсных ограничений (Resource Constrained Project Scheduling Problem) заключается в построении наикратчайшего по суммарной длительности расписания его выполнения при учете ограниченности имеющихся ресурсов и наличия связей предшествования между работами. Пусть проект разбит на n отдельных работ. Обозначим A – сетевой график проекта, наличие ребра (i, j) в котором означает, что работа j не может выполняться до завершения работы i . Пусть также d_i – продолжительность работы i , a_k – имеющееся число единиц ресурса k , r_{ik} – необходимое для выполнения работы i число ресурса k . Необходимо определить такие времена начала работ s_i , чтобы минимизировать время выполнения проекта:

$$\max(s_i + d_i) \rightarrow \min, \quad (1)$$

при условиях

$$s_i + d_i \leq s_j \quad (i, j) \in A, \quad (2)$$

$$s_i \geq 0 \quad i = 1, \dots, n \quad (3)$$

$$\sum_{i \in S_t} r_{ik} \leq a_k \quad k = 1, \dots, m, t = 1, \dots, T. \quad (4)$$

Здесь неравенства (2) – ограничения предшествования, (4) – ограничение на число одновременно занятых ресурсов, S_t – множество работ, находящихся в состоянии выполнения в момент времени t .

Задача RCPSP относится к классу NP-трудных [1], в связи с чем построение оптимальных расписаний сколько-либо крупных проектов за приемлемое время может быть выполнено только различными приближенными алгоритмами. Среди них и генетические алгоритмы, доказавшие свою высокую эффективность, что отражено во множестве работ, например [2-3].

Одним из ключевых вопросов при проектировании любого генетического алгоритма является выбор подходящего метода кодирования хромосомы и реализации генетических операторов мутации и кроссовера. В случае использования кодирования и операторов, лучше учитывающих специфику конкретной задачи, стоит ожидать более качественных результатов. Особую важность для эффективности генетического алгоритма имеет оператор кроссовера, отличающий его от прочих метаэвристических подходов, таких как локальный поиск и роевой интеллект. Для решения задачи RCPSP ранее предложено несколько подходов к кодированию решений, включая кодирование списком работ (activity list representation), вектором приоритетов (random key

representation), списком эвристических правил (priority rule representation) [4]. В работах [2, 5] продемонстрирована наибольшая эффективность первого подхода по сравнению с альтернативами. Он подразумевает, что решение представлено линейным списком работ, упорядоченным с учетом ограничений предшествования. Соответствующее такой закодированной форме расписание получают путем применения алгоритма последовательной или параллельной схемы генерации, который постепенно размещает работы на самое раннее возможное время, добавляя их в заданном порядке и учитывая ресурсные ограничения [6].

При кодировании упорядоченным списком работ для получения новых решений и организации процедуры эвристического поиска применяют классические операторы. Реализация унарного оператора может заключаться в обмене двух произвольных или соседних работ местами или сдвиге произвольной работы [4]. В генетических алгоритмах унарный оператор реализует мутацию хромосом. В качестве же бинарного оператора кроссовера традиционно применяются одноточечный, двухточечный и равномерный (Рисунок 1) [5]. Они стремятся перенести в дочерние особи отдельные подпоследовательности смежных работ из родительских списков или сохранить абсолютные позиции некоторого подмножества работ. Данные операторы традиционны для комбинаторных задач, кодируемых в виде перестановок элементов, и не учитывают специфику задачи проектного планирования. В частности, из-за наличия ресурсных ограничений и прочих неявных связей между работами не абсолютное положение элементов в закодированном порядке, а их расположение относительно друг друга может являться характерным признаком, отличающим более оптимальное решение от всех других.

Таким образом, целью данной работы является разработка эффективного генетического алгоритма, решающего указанные недостатки классических подходов. Для этого необходимо решить ряд задач: ввести подходящий метод кодирования, позволяющий легко вычленять информацию об относительном расположении работ; определить оператор кроссовера, который позволил бы дочерним решениям наследовать подобные признаки, оставляя свободу вариации абсолютного расположения работ в закодированном порядке, провести сравнительное тестирование полученного алгоритма с классическими реализациями.

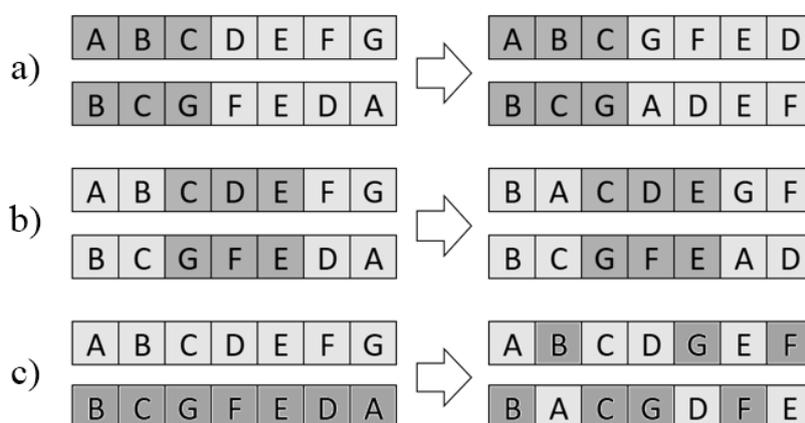


Рисунок 1 – Одноточечный (a), двухточечный (b) и равномерный (c) операторы
Figure 1 – One-point (a), two-point (b) and uniform (c) operators

Материалы и методы

Относительное расположение всех попарно взятых работ удобно представлять в виде квадратной булевой матрицы M размера $n \times n$. Значение 1 в произвольной ячейке

$M[i, j]$ означает, что работа i предшествует работе j . Тогда всех прямых и не прямых последователей работы можно определить по единицам в соответствующей строке, а предшественников – по единицам в соответствующем столбце. Пример подобной матрицы изображен на Рисунке 2.

	A	B	C	D	E	F
A	0	1	1	1	1	1
B	0	0	1	1	1	1
C	0	0	0	0	0	1
D	0	0	1	0	1	1
E	0	0	1	0	0	1
F	0	0	0	0	0	0

Рисунок 2 – Пример булевой матрицы предшествования
Figure 2 – Example of a Boolean precedence matrix

Приведем некоторые значимые свойства подобной матрицы:

- 1) элементы главной диагонали всегда равны нулю;
- 2) если $M[i, j] = 1$, то $M[j, i] = 0$ и наоборот;
- 3) суммы элементов строк и столбцов образуют ряд неотрицательных целых чисел от 0 до $n-1$;
- 4) для получения закодированного линейного порядка достаточно расположить работы по возрастанию сумм соответствующих столбцов или по убыванию сумм соответствующих строк.

С помощью подобного вида матрицы можно представить не только линейный порядок элементов, но и связи предшествования. В этом случае матрица заполняется лишь частично, а всякий допустимый порядок может быть получен произвольным заполнением оставшихся ячеек. Пример подобного представления ограничений приведен на Рисунке 3. Символом «*» отмечены незаполненные ячейки.

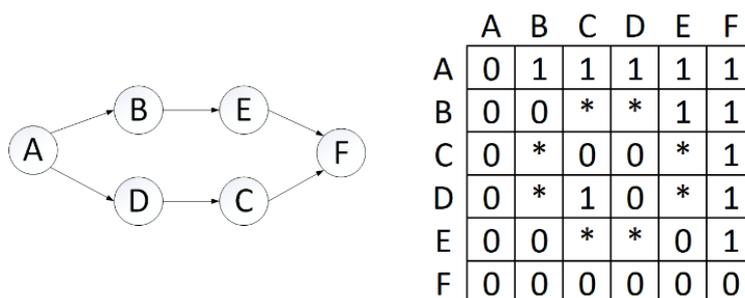


Рисунок 3 – Граф связей предшествования и соответствующая бинарная матрица
Figure 3 – Precedence link graph and corresponding binary matrix

Для порядка работ, закодированного булевой матрицей, могут применяться классические генетические операторы мутации. В частности, оператор обмена двух элементов реализуется за счет обмена содержимого соответствующих им строк и столбцов.

Применение оператора скрещивания в случае кодирования булевыми матрицами должно сохранять относительный порядок некоторых работ. Основной трудностью при этом является сохранение ранее перечисленных свойств. Учитывая это, может быть сформулирован следующий алгоритм.

1. Разделить работы на два непересекающихся множества. Для этого работы рассматриваются в порядке, заданном одним из родителей, и добавляются в одно из подмножеств. При этом на каждом шаге с некоторой вероятностью может произойти смена текущего заполняемого подмножества на другое. Процедура выполняется, пока все работы не окажутся распределены.

2. Перенести значения ячеек, отражающих относительное расположение работ первого подмножества, из первого родителя в дочернее решение.

3. Выполнить операцию дизъюнкции с матрицей, кодирующей ограничения предшествования, для их удовлетворения.

4. Заполнить оставшиеся пустыми ячейки дочерней матрицы. Для этого последовательно на каждом i -ом шаге рассматриваются столбцы, сумма элементов которых равна i . Из них отбирается столбец k , не имеющий незаполненных клеток, или чья соответствующая работа расположена ранее во втором родителе. Работа k располагается на i -ую позицию. Для этого во всех незаполненных ячейках k -ой строки, соответствующих предшественникам, записывается значение 1.

5. Вторая дочерняя матрица получается аналогичным образом, повторив шаги 2-4 с заменой первого и второго родителя местами.

Указанный алгоритм после переноса информации о подмножестве работ и связей предшествования стремится заполнить пустые ячейки так, чтобы все прочие работы расположились в том же относительном порядке, что и во втором родителе. При этом обеспечивается сохранение всех свойств подобного матричного представления. На Рисунке 4 изображены две родительские матрицы, а также их дочерние решения вместе с промежуточным состоянием, получаемым после выполнения шагов 1-3 алгоритма. Были взяты ограничения предшествования с Рисунка 3.

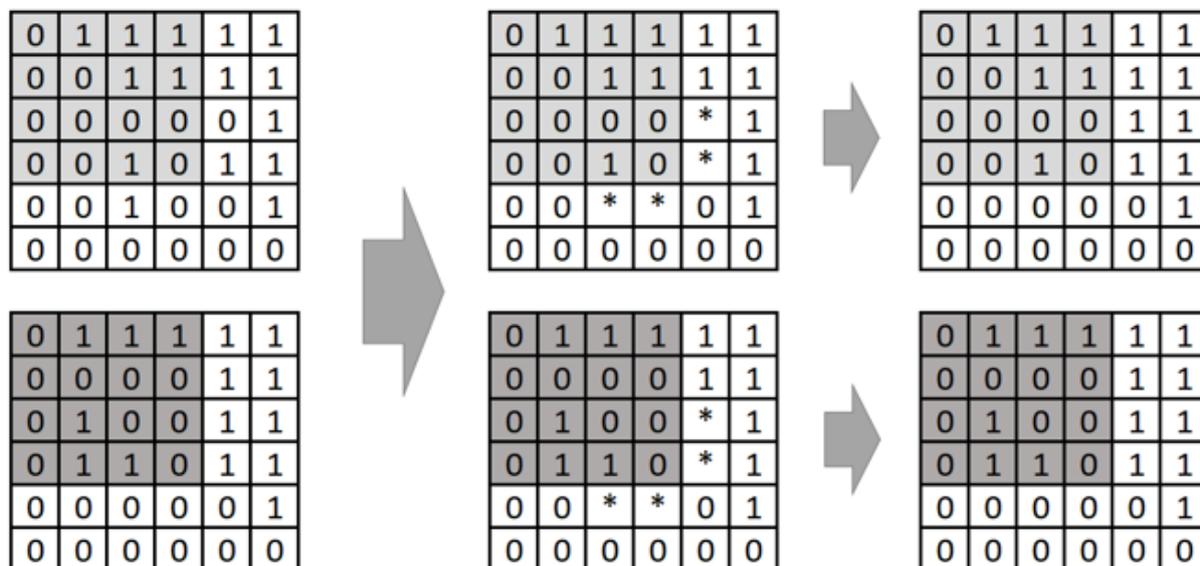


Рисунок 4 – Оператор скрещивания булевых матриц
Figure 4 – Boolean matrix crossover operator

На основе предложенного метода кодирования хромосомы и оператора скрещивания может быть сформулирован модифицированный генетический алгоритм. В данной реализации использовались компоненты генетических алгоритмов из [7-8]. Ее основные элементы:

1) В качестве функции приспособленности выступает длительность расписания. Для декодирования расписания используется последовательная схема генерации.

2) Начальная популяция генерируется случайным образом. Для этого каждой особи ставится в соответствие пустая булева матрица и пустое множество работ J . Затем последовательно выбирается очередная случайная работа j , чьи предшественницы уже добавлены в J . Все ячейки столбца j , соответствующие элементам из J , заполняются единицами, а j добавляется в J .

3) Мутация хромосомы осуществляется следующим образом. Для каждой работы, расположенной на i -ой позиции, берется случайное значение $p_i \in [0,1]$. Если $p_i \leq P_{PM}$, где P_{PM} – заранее заданная вероятность мутации, то работы на позициях i и $i+1$ меняются местами, если это не нарушает ограничений предшествования.

4) Перед скрещиванием вся популяция разбивается на пары. Оператор скрещивания применяется с заранее заданной вероятностью P_{PC} .

5) Применяется турнирный метод селекции с размером группы 3. Из трех случайно отбираемых особей одна наиболее приспособленная отправляется в новое поколение. Процедура выполняется до достижения необходимой фиксированной численности популяции.

Булевы матрицы предшествования представимы в каноническом для генетических алгоритмов виде булевых строк. Для этого необходимо ввести следующее преобразование:

$$\sum_{i=1}^{n-1} (n-i)! \times Z_i, \quad (5)$$

где Z_i – число нулей в i -ой строке верхнетреугольной матрицы. К примеру, для матрицы с Рисунка 4 получим следующий результат: $(6-1)! \times 0 + (6-2)! \times 0 + (6-3)! \times 2 + (6-4)! \times 0 + (6-5)! \times 0 = 12 \Leftrightarrow (1\ 1\ 0\ 0)$.

Таким образом, для обоснования сходимости генетического алгоритма может применяться теорема схем Холланда, сформулированная для канонического генетического алгоритма с кодированием бинарными строками [9].

Результаты и обсуждение

Для сравнения модифицированного генетического алгоритма составления расписания проекта с классическими подходами из базы PSBLIB [10] были отобраны экземпляры задач размером 60, 90 и 120 работ. Все задачи были решены пятью вариациями генетического алгоритма, различающиеся только методами кодирования и реализациями генетического оператора скрещивания:

1) *One-point* – кодирование списком работ и однотоочный оператор скрещивания;

2) *Two-point* – кодирование списком работ и двухточечный оператор скрещивания;

3) *Uniform 1* – кодирование списком работ и равномерный оператор скрещивания;

4) *Uniform 2* – кодирование списком работ и равномерный оператор скрещивания с принципом разбиения работ на два подмножества, аналогичным применяемому в предлагаемом для булевых матриц;

5) *Matrix* – кодирование булевыми матрицами и предложенный оператор скрещивания.

Для устранения влияния случайных факторов и проверки эффективности предложенных модификаций вероятность мутации была равна нулю, а начальная популяция была одинакова для всех алгоритмов. Размер популяции равнялся 80, число

популяций – 40, вероятность смены направления на каждом шаге алгоритма разбиения множества работ – 0.2.

Всего было случайно отобрано 144 экземпляра задачи из библиотеки PSBLIB, по 48 штук каждого из трех рассмотренных размеров. Результаты представлены в Таблице 1. Для каждого размера и алгоритма указано абсолютное число и процент экземпляров, для которых данным алгоритмом получено наилучшее, по сравнению с прочими, решение.

Таблица 1 – Результаты решения тестовых задач
Table 1 – Results of solving test problems

	<i>One-point</i>	<i>Two-point</i>	<i>Uniform 1</i>	<i>Uniform 2</i>	<i>Matrix</i>
j60.sm (48 задач)	31 (64.6%)	34 (70.1%)	30 (62.5%)	31 (64.6%)	46 (95.8%)
j90.sm (48 задач)	29 (60.4%)	33 (68.8%)	25 (52.1%)	26 (54.2%)	45 (93.8%)
j120.sm (48 задач)	7 (14.6%)	22 (45.8%)	3 (6.25%)	7 (14.6%)	34 (70.8%)
Итог (144 задачи)	67 (46.5%)	89 (61.8%)	58 (40.3%)	64 (44.4%)	125 (86.8%)

Как видно из результатов тестов, модифицированный генетический алгоритм оказался эффективнее на всех размерах задачи. Второй по эффективности оказалась реализация с двухточечным кроссовером. Наиболее явно разница проявилась на самом большом размере задачи, где остальные три варианта резко деградировали в своей относительной эффективности. Таким образом, можно сделать вывод, что предложенный модифицированный генетический алгоритм обладает потенциалом для решения крупных задач с большим числом работ. Такой случай часто встречается в реальной практике проектного планирования.

Заключение

Предложенный в работе модифицированный генетический алгоритм составления расписания проекта показал лучшие результаты на тестовой выборке задач по сравнению с классическими подходами. Этого удалось достичь благодаря применению кодирования в виде булевых матриц и соответствующего оператора кроссовера, которые позволили вычленять в качестве наследуемых признаков относительные расположения некоторых работ. Предложенный алгоритм может быть реализован в прикладных программных системах, ориентированных на календарное планирование массивных проектов.

СПИСОК ИСТОЧНИКОВ

1. Artigues C. The Resource-Constrained Project Scheduling Problem. In: Artigues C., Demasse S., Neron E. (eds.). *Resource-Constrained Project Scheduling: Models, Algorithms, Extensions and Applications*. Hoboken: ISTE; 2008:21–35. DOI:10.1002/9780470611227.ch1.
2. Kolisch R., Hartmann S. Experimental investigation of heuristics for resource-constrained project scheduling. *European Journal of Operational Research*. 2006;174(1):23–37. DOI:10.1016/j.ejor.2005.01.065.
3. Pellerin R. A survey of hybrid metaheuristics for the resource-constrained project scheduling problem. *European Journal of Operational Research*. 2020;280(2):395–416. DOI:10.1016/j.ejor.2019.01.063.

4. Kolisch R., Hartmann S. Heuristic Algorithms for the Resource-Constrained Project Scheduling Problem: Classification and Computational Analysis. In: Węglarz J. (eds.). *Project Scheduling*. Boston, MA: Springer; 1999:147–178. DOI:10.1007/978-1-4615-5533-9_7.
5. Hartmann S. A competitive genetic algorithm for resource-constrained project scheduling. *Naval Research Logistics*. 1998;45:733–750. DOI:10.1109/ICMLC.2005.1527446.
6. Kim J., Ellis R. Comparing Schedule Generation Schemes in Resource-Constrained Project Scheduling Using Elitist Genetic Algorithm. *Journal of Construction Engineering and Management*. 2010;136(2):160–169. DOI:10.1061/(ASCE)0733-9364(2010)136:2(160).
7. Зацаринный А.А., Коротков В.В., Матвеев М.Г. Моделирование процессов сетевого планирования портфеля проектов с неоднородными ресурсами. *Информатика и её применения*. 2019;13(2):92–99. DOI:10.14357/19922264190213.
8. Korotkov V.V., Matveev M.G. Individual Scheduling for the Multi-Mode Resource-Constrained Multi-Project Scheduling Problem. In: Becker J., Matveev M., Taratukhin V. (eds.). *Proceedings of the 1st International Conference of Information Systems and Design*. 2019. Available by: <http://ceur-ws.org/Vol-2570/paper5.pdf>.
9. Altenberg L. The Schema Theorem and Price's Theorem. *Foundations of Genetic Algorithms*. 1995;3:23–49. DOI:10.1016/B978-1-55860-356-1.50006-6.
10. Kolisch R., Sprecher A. PSPLIB – A project scheduling library. *European Journal of Operational Research*. 1996;96(1):205–216. DOI:10.1016/S0377-2217(96)00170-1.

REFERENCES

1. Artigues C. The Resource-Constrained Project Scheduling Problem. In: Artigues C., Demassey S., Neron E. (eds.). *Resource-Constrained Project Scheduling: Models, Algorithms, Extensions and Applications*. Hoboken: ISTE; 2008:21–35. DOI:10.1002/9780470611227.ch1.
2. Kolisch R., Hartmann S. Experimental investigation of heuristics for resource-constrained project scheduling. *European Journal of Operational Research*. 2006;174(1):23–37. DOI:10.1016/j.ejor.2005.01.065.
3. Pellerin R. A survey of hybrid metaheuristics for the resource-constrained project scheduling problem. *European Journal of Operational Research*. 2020;280(2):395–416. DOI:10.1016/j.ejor.2019.01.063.
4. Kolisch R., Hartmann S. Heuristic Algorithms for the Resource-Constrained Project Scheduling Problem: Classification and Computational Analysis. In: Węglarz J. (eds.). *Project Scheduling*. Boston, MA: Springer; 1999:147–178. DOI:10.1007/978-1-4615-5533-9_7.
5. Hartmann S. A competitive genetic algorithm for resource-constrained project scheduling. *Naval Research Logistics*. 1998;45:733–750. DOI:10.1109/ICMLC.2005.1527446.
6. Kim J., Ellis R. Comparing Schedule Generation Schemes in Resource-Constrained Project Scheduling Using Elitist Genetic Algorithm. *Journal of Construction Engineering and Management*. 2010;136(2):160–169. DOI:10.1061/(ASCE)0733-9364(2010)136:2(160).
7. Zatsarinnyi A.A., Korotkov V.V., Matveev M.G. Modeling the process of network planning of a portfolio of projects with heterogeneous resources under fuzziness. *Информатика и её применения = Informatics and its Applications*. 2019;13(2):92–99. DOI:10.14357/19922264190213. (In Russ.)
8. Korotkov V.V., Matveev M.G. Individual Scheduling for the Multi-Mode Resource-Constrained Multi-Project Scheduling Problem. In: Becker J., Matveev M., Taratukhin V. (eds.). *Proceedings of the 1st International Conference of Information Systems and Design*. 2019. Available by: <http://ceur-ws.org/Vol-2570/paper5.pdf>.

9. Altenberg L. The Schema Theorem and Price's Theorem. Foundations of Genetic Algorithms. 1995;3:23–49. DOI:10.1016/B978-1-55860-356-1.50006-6.
10. Kolisch R., Sprecher A. PSPLIB – A project scheduling library. European Journal of Operational Research. 1996;96(1):205–216. DOI:10.1016/S0377-2217(96)00170-1.

ИНФОРМАЦИЯ ОБ АВТОРАХ / INFORMATION ABOUT THE AUTHORS

Коротков Владислав Владимирович, Vladislav Vladimirovich Korotkov, Assistant ассистент Воронежского государственного университета, Воронеж, Российская Федерация. Lecturer at Voronezh State University, Voronezh, Russian Federation.

e-mail: korotkov@cs.vsu.ru

ORCID: [0000-0002-4045-7721](https://orcid.org/0000-0002-4045-7721)

Статья поступила в редакцию 19.07.2022; одобрена после рецензирования 23.08.2022; принята к публикации 07.09.2022.

The article was submitted 19.07.2022; approved after reviewing 23.08.2022; accepted for publication 07.09.2022.