

УДК 004.05

DOI: [10.26102/2310-6018/2022.37.2.003](https://doi.org/10.26102/2310-6018/2022.37.2.003)

Анализ тестовых задач мультиверсионного формирования отказоустойчивых программных систем

И.В. Ковалев^{1,2,3}, Д.И. Ковалев^{1,2}, Н.Д. Амбросенко¹, Д.В. Боровинский⁴

¹Красноярский государственный аграрный университет,
Красноярск, Российская Федерация

²Красноярский краевой Дом науки и техники РосСНИО,
Красноярск, Российская Федерация

³Сибирский федеральный университет,
Красноярск, Российская Федерация

⁴Сибирская пожарно-спасательная академия ГПС МЧС России,
Железногорск, Российская Федерация
grimm7jow@gmail.com

Резюме: Актуальность исследования обусловлена современными требованиями к надежности функционирования программных систем критических приложений. Авторами развивается подход, базирующийся на современной информационной технологии мультиверсионного формирования высоконадежных программных систем. В статье проведен анализ тестовых задач мультиверсионного формирования отказоустойчивых программных систем, выполненный на базе алгоритмов муравьиной колонии, включая стандартный и модифицированный алгоритмы. В работе программная система определяется заданным набором программных модулей, соединенных определенным образом и образующих граф переходов с вероятностями перехода. Причем исполнение каждого модуля является мультиверсионным, то есть модуль включает несколько версий, каждая из которых характеризуется значением надежности и стоимости исполнения. В результате, совокупность выбранных для исполнения версий в модуле определяет его надежность и стоимость, а благодаря наличию графа программы мы имеем возможность вычислить надежность и стоимость всей программной системы. Условия задачи включают ограничения, накладываемые на надежность и стоимость итогового решения. При анализе использовалась заданная схема программной системы, учитывался долговременный режим реализации программных функций и возможность изменения структуры программы в процессе ее реализации. Показано, что использование модифицированного алгоритма дает нам преимущество не только в качестве значения целевой функции, но и в скорости улучшения этого решения, что особенно важно для практики при реализации программных систем в режиме реального времени.

Ключевые слова: программная система, отказоустойчивость, алгоритм муравьиной колонии, мультиверсионный метод, тестовая задача.

Благодарности: проведение исследований осуществляется при поддержке КГАУ «Красноярский краевой фонд поддержки научной и научно-технической деятельности» в рамках проекта «Контроль траектории полета воздушных судов в экстремальных условиях Арктики и Крайнего Севера» в соответствии с заявкой 2021110907918.

Для цитирования: Ковалев И.В., Ковалев Д.И., Амбросенко Н.Д., Боровинский Д.А. Анализ тестовых задач мультиверсионного формирования отказоустойчивых программных систем. *Моделирование, оптимизация и информационные технологии*. 2022;10(2). Доступно по: <https://moitvvt.ru/ru/journal/pdf?id=1167> DOI: 10.26102/2310-6018/2022.37.2.003

Test tasks analysis of fault-tolerant software system multiversion formation

I.V. Kovalev^{1,2,3}, D.I. Kovalev^{1,2}✉, N.D. Ambrosenko¹, D.V. Borovinsky⁴

¹*Krasnoyarsk State Agrarian University,
Krasnoyarsk, Russian Federation*

²*Krasnoyarsk Science & Technology City Hall,
Krasnoyarsk, Russian Federation*

³*Siberian Federal University,
Krasnoyarsk, Russian Federation*

⁴*Siberian Fire and Rescue Academy EMERCOM of Russia,
Zheleznogorsk, Russian Federation
grimm7jow@gmail.com✉*

Abstract: The relevance of the study is due to the modern requirements for the operational reliability of software systems for critical applications. The authors develop an approach based on modern information technology of highly reliable software system multiversion formation. The paper analyzes test tasks of fault-tolerant software system multiversion formation with the aid of ant colony algorithms including standard and modified algorithms. In this article, a software system is defined by a predefined set of software modules connected in a particular way and forming a transition graph with transition probabilities. Moreover, the execution of each module is multiversional, in other words, the module is comprised of several versions with each one characterized by the value of reliability and cost of execution. As a result, the set of versions, selected for execution in the module, determines its reliability and cost, and, owing to the presence of the program graph, we are able to calculate the reliability and cost of the entire software system. The conditions of the problem feature restrictions on the reliability and cost of the final solution. A predefined scheme of the software system was used in the analysis, taking into account the long-term mode of program functions implementation and the capacity to change program structure in the process of its implementation. It is shown that the employment of the modified algorithm provides an advantage not only in the quality of the objective function value, but also in the speed of improving this solution, which is especially important for practical purposes when implementing software systems in real time.

Keywords: software system, fault tolerance, ant colony algorithm, multiversion method, test task.

Acknowledgments: the research is carried out with the support of the Krasnoyarsk Regional Fund of Science and Technology Support under the project “Control of the flight path of aircraft in extreme conditions of the Arctic and the Far North” in accordance with proposal 2021110907918.

For citation: Kovalev I.V., Kovalev D.I., Ambrosenko N.D., Borovinsky D.V. Test tasks analysis of fault-tolerant software system multiversion formation. *Optimization and Information Technology*. 2022;10(2). Available from: <https://moitvvt.ru/ru/journal/pdf?id=1167> DOI: 10.26102/2310-6018/2022.37.2.003 (In Russ.).

Введение

В работе [1] авторами была рассмотрена модификация и программная реализация алгоритма муравьиной колонии для формирования состава мультиверсионного программного обеспечения с динамической архитектурой. Результаты моделирования [2-4] показали применимость разработанного подхода. Однако для того, чтобы оценить предложенные модификации базового алгоритма, необходимо провести исследования параметров его работы на ряде тестовых примеров, которые позволят проверить эффективность решений тестовых задач различными алгоритмами.

Как правило, постановка задачи [5-7] определяется заданным набором программных модулей, соединенных определенным образом, и образующих граф переходов с вероятностями перехода из одного модуля в другой. Исполнение каждого модуля является мультиверсионным. Модуль включает определенное количество версий, каждая из которых характеризуется значением надежности и стоимости исполнения. Совокупность выбранных для исполнения версий в модуле определяет его надежность и стоимость, а благодаря наличию графа программы мы имеем возможность вычислить надежность и стоимость программной системы (ПС). Условия задачи включают ограничения, накладываемые на надежность и стоимость итогового решения [8-10].

При анализе тестовых задач авторы основное внимание уделили задачам с долговременными этапами реализации программных функций (время работы программы от 1200 с и более) и задачам с изменением структуры программы в процессе реализации ее программных функций.

Постановка тестовой задачи

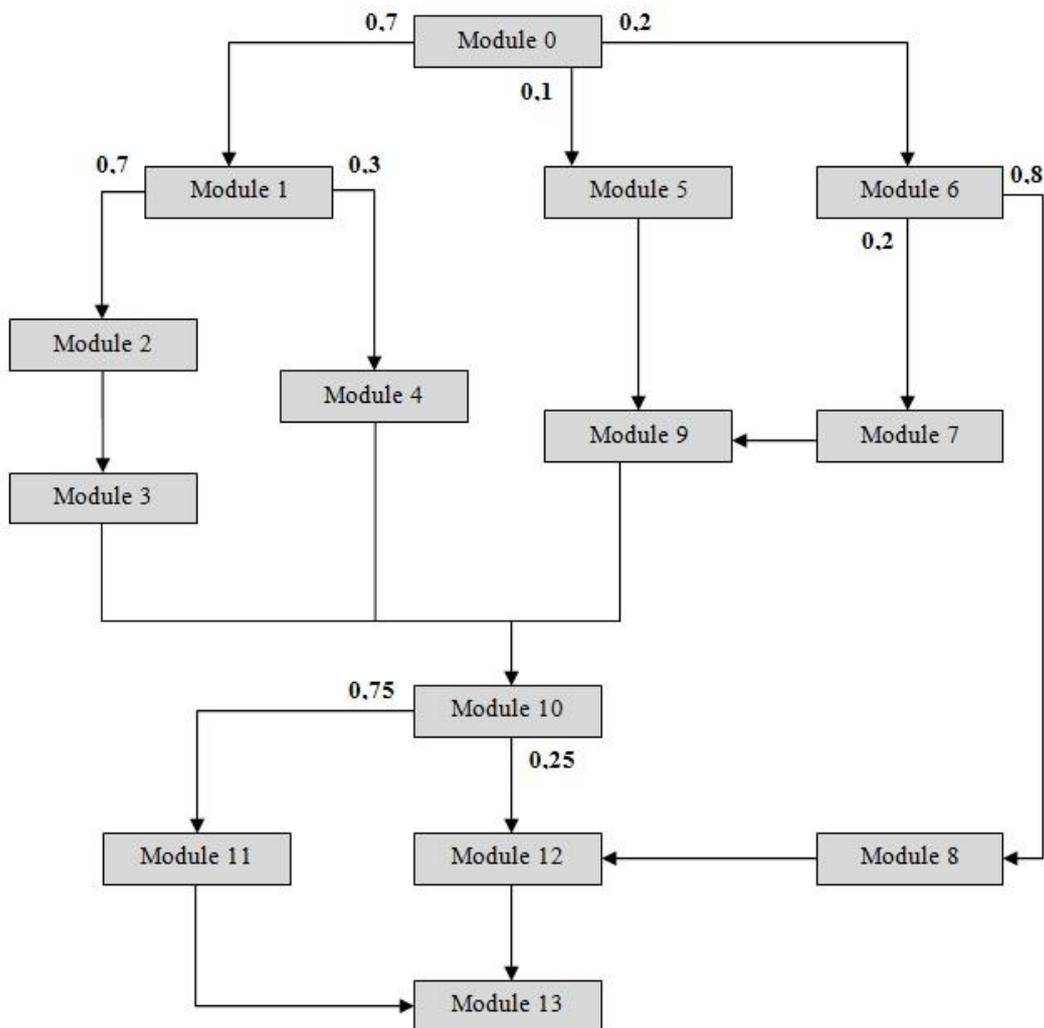


Рисунок 1 – Схема ПС в тестовой задаче
 Figure 1 – Scheme of a software system in a test task

Для того чтобы получить данные о сравнении результатов работы стандартного и модифицированного алгоритмов, была создана тестовая задача [4], на которой проводились серии экспериментов для получения и сравнения показателей эффективности. Программная система представляет собой набор из 140 версий, каждая из которых обладает показателем надежности R и стоимости ее использования C . Эти 140 версий распределены на группы по 10 версий, образующих программные модули. Схема ПС представлена на Рисунке 1, где у каждого модуля указаны вероятности перехода от одного модуля к другому.

Характеристики модулей, которые используются при расчете, следующие:

1. время работы программы 1200 с;
2. квант времени 200 мс;
3. время реакции на изменение 100 мс;
4. $C \rightarrow \min, R_{\min} = 0,995, 0 \leq t < 600$;
5. $R \rightarrow \max, C_{\max} = 2300, 600 \leq t < 1200$.

Результаты и анализ решения тестовых задач

Для того чтобы оценить эффективность вследствие модификации алгоритма, произведем сначала расчет стандартным алгоритмом [11, 12] при значениях вероятности выбора версии по формуле (1).

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta}, \text{ если } j \in N_i^k. \quad (1)$$

Здесь N_i^k является набором вершин, в которые возможен переход для k -го муравья, когда он находится в вершине i . В начале поиска устанавливается начальное значение феромона, равное определенной константе (например, $\tau_{ij} = 1, \forall (i, j) \in A$), для всех граней. При нахождении в i -м узле k -й муравей использует значение феромона для вычисления вероятности p_{ij}^k перехода в вершину j . Коэффициент α и β получены из эксперимента в процессе подбора [1, 4].

Значение эвристической привлекательности рассчитывается в соответствии с формулой (2).

$$\begin{cases} \eta_{ij} = 1/c_{ij}, C \rightarrow \min \\ \eta_{ij} = r_{ij}, R \rightarrow \max \end{cases} \quad (2)$$

В результате сбора статистики работы программы по данному алгоритму, были получены данные, на основании которых построены следующие графики, представленные на Рисунках 2 и 3.

Из графика на Рисунке 2 видим, что на первом этапе при минимизации стоимости решения были получены следующие данные: начальная стоимость найденного решения $C = 2941,9$. Оптимальная стоимость найденного решения $C = 2250,8$. За начальные 82 с были достигнуты 90 % улучшений начального решения и было получено значение $C = 2316,2$.

На втором этапе (график на Рисунке 3) при максимизации надежности решения были получены следующие данные: начальная надежность найденного решения $R = 0,983805$. Оптимальная надежность найденного решения $R = 0,995989$. За начальные 54 с были достигнуты 90 % улучшений начального решения и было получено значение $R = 0,994812$.

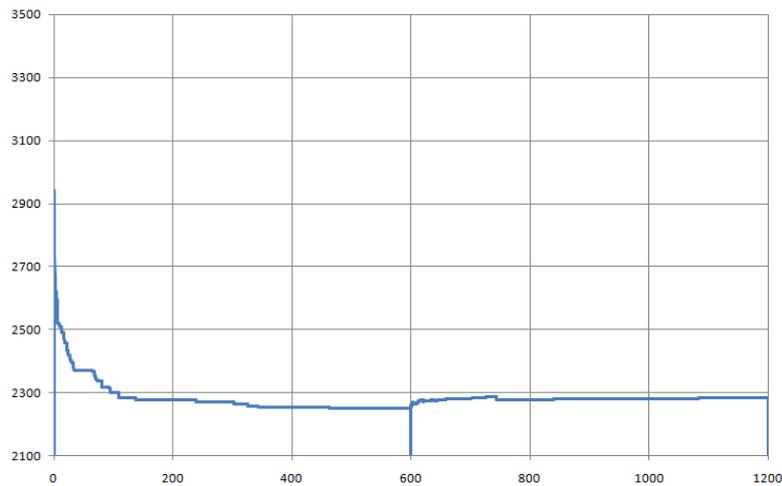


Рисунок 2 – Изменение стоимости решения (стандартный алгоритм)
 Figure 2 – Changing the cost of a solution (standard algorithm)

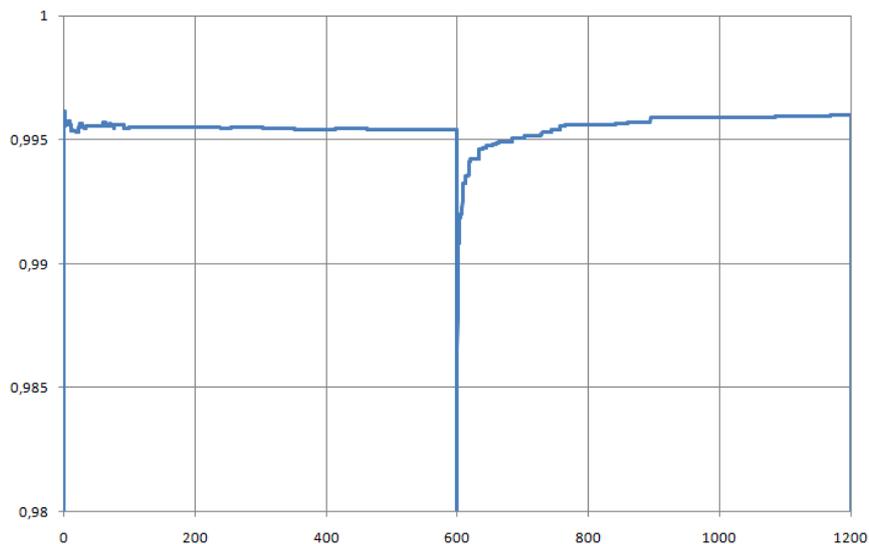


Рисунок 3 – Изменение надежности решения (стандартный алгоритм)
 Figure 3 – Changing the reliability of a solution (standard algorithm)

В рамках модифицированного алгоритма для того, чтобы задействовать информацию о всех оптимизируемых параметрах, было принято решение внести изменение в способ расчета значения эвристической привлекательности. Использовались формулы (3), (4), по которым осуществлялся расчет. Вероятность выбора версии равна

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta}, \text{ если } j \in N_i^k. \quad (3)$$

Значение эвристической привлекательности рассчитывается по формуле (4).

$$\eta_{ij} = r_{ij}/c_{ij}. \quad (4)$$

В результате сбора статистики работы программы на базе модифицированного алгоритма были получены данные, на основании которых построены графики на Рисунках 4 и 5. На них также приведен график изменения параметров при применении стандартного алгоритма для визуализации сравниваемых результатов.

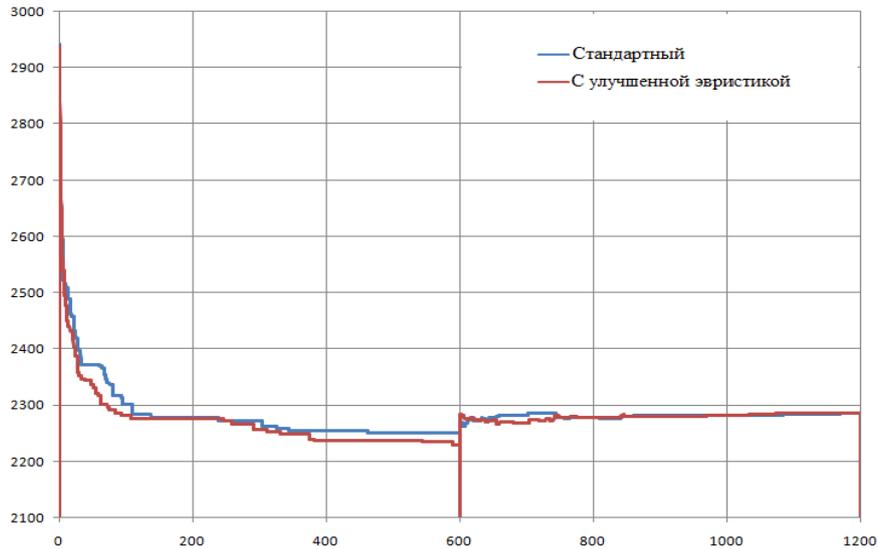


Рисунок 4 – Изменение стоимости решения (стандартный алгоритм и алгоритм с улучшенной эвристикой)

Figure 4 – Changing the cost of a solution (standard algorithm and algorithm with improved heuristics)

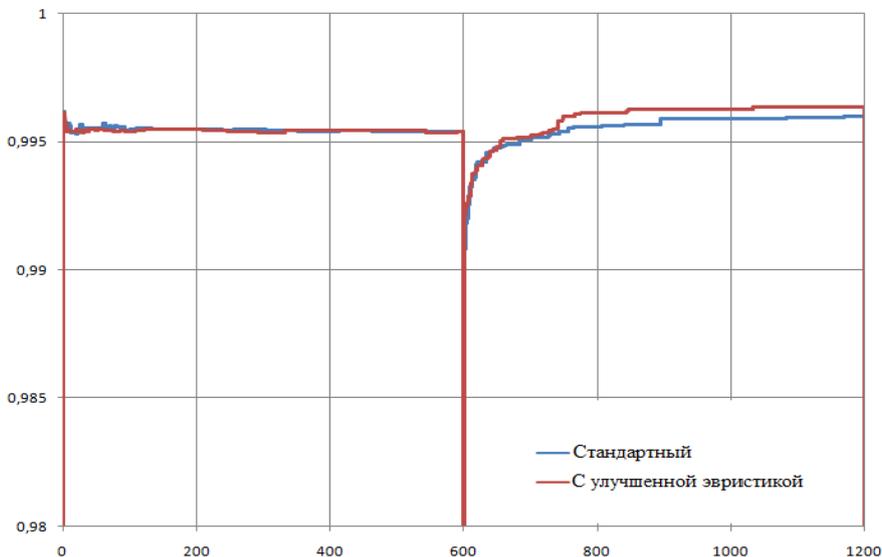


Рисунок 5 – Изменение надежности решения (стандартный алгоритм и алгоритм с улучшенной эвристикой)

Figure 5 – Changing the reliability of a solution (standard algorithm and algorithm with improved heuristics)

На первом этапе (Рисунок 4) при минимизации стоимости решения были получены следующие результаты: начальная стоимость найденного решения $C = 2938,5$. Оптимальная стоимость найденного решения $C = 2228,4$. За начальные 73 с

были достигнуты 90 % улучшений начального решения и было получено значение $C = 2295,5$.

На втором этапе (Рисунок 5) при максимизации надежности решения были получены следующие результаты: начальная надежность найденного решения $R = 0,982397$. Оптимальная надежность найденного решения $R = 0,996371$. За начальные 54 с были достигнуты 90 % улучшений начального решения и было получено значение $R = 0,995024$.

Как видно из графиков (Рисунки 4 и 5), алгоритм с модифицированным значением эвристической привлекательности демонстрирует либо сравнимые, либо лучшие результаты на протяжении всей своей работы. В результате сравнения стоимости найденных решений на первом этапе работы алгоритма выявлено, что модифицированный алгоритм демонстрирует на 0,9952 % лучший результат, чем стандартный алгоритм. На втором этапе результат, достигнутый при использовании модифицированного алгоритма на 0,038 % превосходит результат в случае использования стандартного алгоритма.

Для того чтобы задействовать информацию о структуре оптимизируемой ПС [2], было принято решение добавить в формулу расчета вероятности информации о вероятности использования модуля g_i , а также количестве уже выбранных версий данного модуля n_i^k . В результате получена формула (5), по которой и осуществлялся дальнейший расчет. Итак, вероятность выбора версии определяется по формуле (5).

$$p_{ij}^k = \frac{\frac{g_i}{n_i^k} [\tau_{ij}]^\alpha [n_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}]^\alpha [n_{il}]^\beta}, \text{ если } j \in N_i^k. \quad (5)$$

Значение эвристической привлекательности, как и ранее, рассчитывается по формуле (4).

В результате обработки тестовых данных при работе программы с данным алгоритмом, были получены следующие графики, представленные на Рисунках 6 и 7.

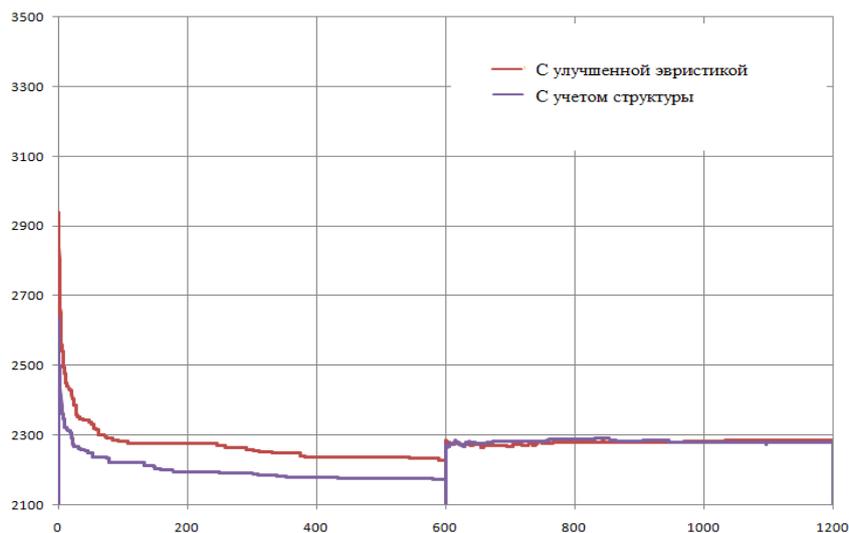


Рисунок 6 – Изменение стоимости решения с учетом структуры ПС
Figure 6 – Changing the cost of a solution, taking into account the structure of a software system

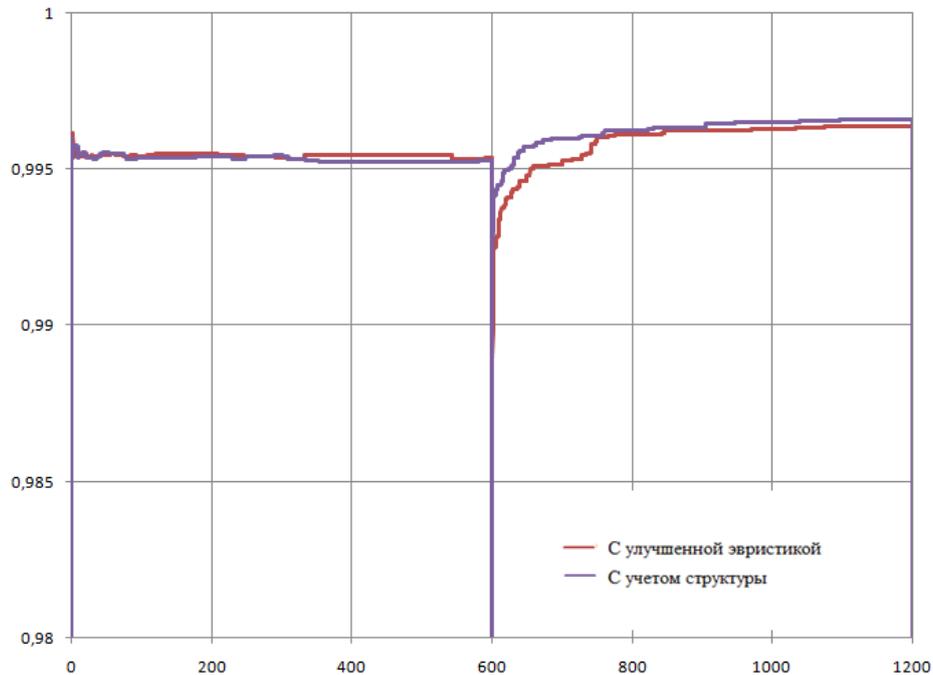


Рисунок 7 – Изменение надежности решения с учетом структуры ПС

Figure 7 – Changing the reliability of a solution, taking into account the structure of a software system

На Рисунках 6 и 7 также приведен график изменения параметров при применении алгоритма с модифицированной эвристикой для визуального сравнения полученных данных. Как видно из этих графиков, алгоритм, учитывающий структуру ПС с новой формулой (5) расчета эвристической привлекательности, демонстрирует лучшие результаты на протяжении всей своей работы.

На первом этапе при минимизации стоимости решения получены следующие данные: начальная стоимость найденного решения $C = 2636,4$. Оптимальная стоимость найденного решения $C = 2172,7$. За начальные 133 с были достигнуты 90 % улучшений начального решения и было получено значение $C = 2213,9$.

На втором этапе при максимизации надежности решения получена начальная надежность найденного решения $R = 0,990544$. Оптимальная надежность найденного решения $R = 0,996597$. За начальные 124 с были достигнуты 90 % улучшений начального решения и было получено значение $R = 0,996042$.

На Рисунках 8 и 9 приведены графики сравнения алгоритма, учитывающего структуру ПС и стандартного алгоритма. В результате сравнения стоимости найденных решений на первом этапе работы было получено, что достигнутый результат модифицированного алгоритма на 3,47 % превосходит результат стандартного алгоритма. На втором этапе результат, достигнутый при использовании модифицированного алгоритма, на 0,061 % превосходит результат стандартного алгоритма.

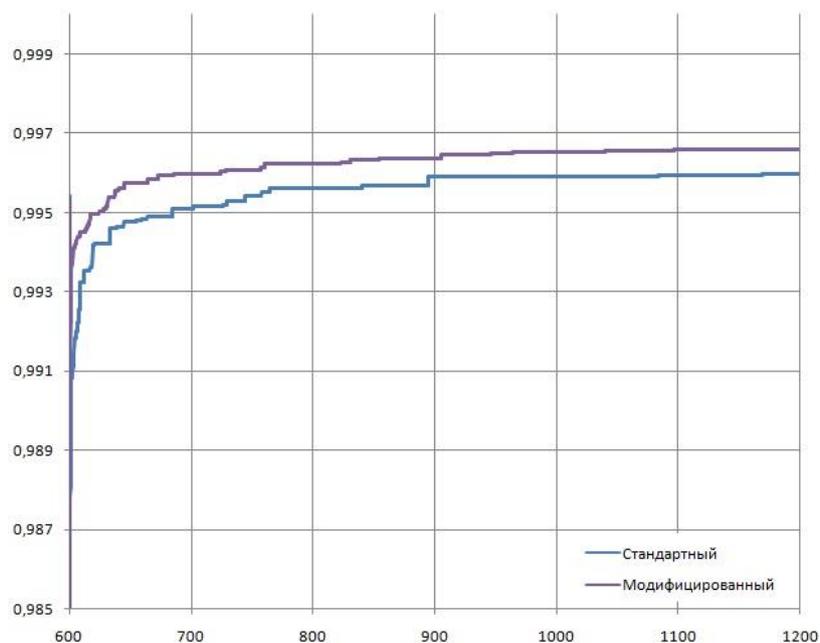


Рисунок 8 – Изменение надежности решения (стандартный алгоритм и алгоритм, учитывающий структуру ПС)

Figure 8 – Changing the reliability of a solution (standard algorithm and algorithm that takes into account the structure of a software system)

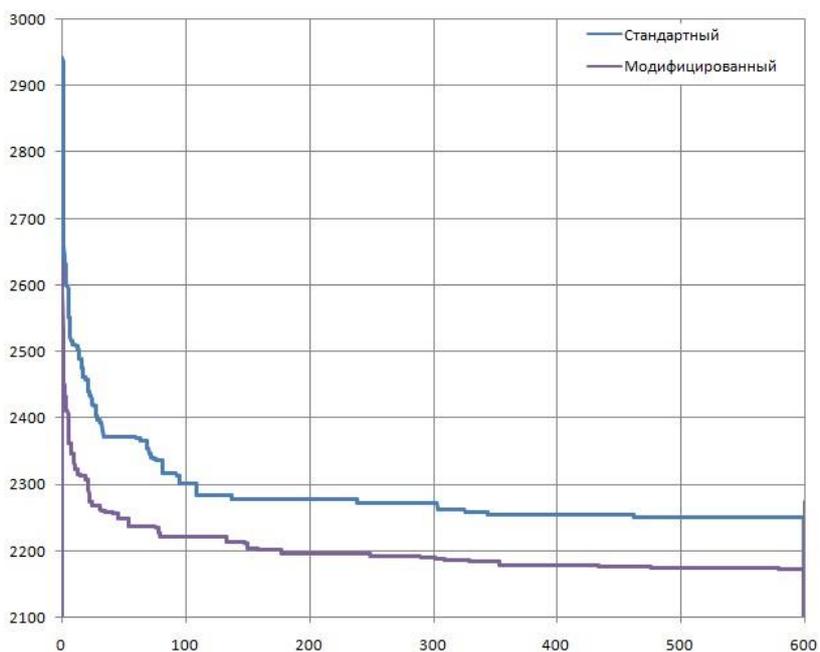


Рисунок 9 – Изменение стоимости решения (стандартный алгоритм и алгоритм, учитывающий структуру ПС)

Figure 9 – Changing the cost of a solution (standard algorithm and algorithm that takes into account the structure of a software system)

Заключение

Таким образом, на основании обработки статистических данных было установлено, что на первом этапе при использовании стандартного алгоритма

оптимальный результат при значении $C = 2250,8$ был достигнут на 463 секунде. Аналогичный результат достигается при использовании модифицированного алгоритма на 46 секунде, что дает нам 90-процентный выигрыш по времени достижения данного результата. На втором этапе при использовании стандартного алгоритма оптимальный результат в $P = 0,995989$ был достигнут за 568,7 секунды. Аналогичный результат достигается при использовании модифицированного алгоритма на 124 секунде, что дает нам 78 % выигрыша по времени достижения данного результата. Следовательно, использование модифицированного алгоритма дает нам преимущество не только в качестве значения целевой функции, но и в скорости улучшения этого решения, что особенно важно при реализации ПС в режиме реального времени.

СПИСОК ИСТОЧНИКОВ

1. Ковалев И.В., Ковалев Д.И., Брит А.А., Сарамуд М.В. Информационная технология для мультиверсионного формирования отказоустойчивых программных систем. *Системы управления и информационные технологии*. 2021;2(84):56–68. DOI: 10.36622/VSTU.2021.84.2.013.
2. Буховцев Д.Д. Применение модифицированного алгоритма муравьиной колонии для решения задачи календарного планирования распределенных предприятий. *Современные инновации, системы и технологии*. 2021;1(1):29–42. Доступно по: <https://doi.org/10.47813/2782-2818-2021-1-1-29-42>.
3. Сарамуд М.В. К вопросу прогнозирования времени наработки на отказ отказоустойчивых программных комплексов. *Материалы XXIV Международной научно-практической конференции, посвященной памяти генерального конструктора ракетно-космических систем академика М. Ф. Решетнева*. Красноярск, 2020:453–454.
4. Сарамуд М.В., Ковалев Д.И. Средство автоматизированного проектирования мультиверсионного программного комплекса. *Свидетельство о регистрации программы для ЭВМ 2021610550*, 15.01.2021. Заявка № 2020667734, 28.12.2020.
5. Ning J., Zhang C., Sun P., Feng Y. Comparative study of ant colony algorithms for multi-objective optimization. *Information*. 2019;10(1):11.
6. Jeona Young-Jae Kimb, Jae-Chul, Yunc Sang-Yun, Leed Kwang Y. Application of ant colony algorithm for network reconfiguration in distribution systems. *IFAC Proceedings Volumes*. 2003;36(20):773–778.
7. Борисенков Д.С. Преобразование структуры программы на этапе проектирования. *Перспективы развития информационных технологий*. 2014;22:15–20.
8. Поначугин А.В. Определение надёжности программного обеспечения в структуре современной информационной системы. *Кибернетика и программирование*. 2019;2:65–72.
9. Зенюткин Н.В., Ковалев Д.И., Туев Е.В, Туева Е.В. О способах формирования информационных структур для моделирования объектов, сред и процессов. *Современные инновации, системы и технологии*. 2021;1(1):10–22. Доступно по: <https://doi.org/10.47813/2782-2818-2021-1-1-10-22>.
10. Гимаров В.В., Глушко С.И., Дли М.И. Применение алгоритмов муравьиной колонии при управлении сложными проектами. *Транспортное дело России*. 2012;4:107–109.
11. Dorigo M., Stützle T. Ant colony optimization: Overview and recent advances. *Handbook of Metaheuristics. International Series in Operations Research & Management Science*. 2010;146:5–8. Available by: https://doi.org/10.1007/978-1-4419-1665-5_8.
12. Colormi A., Dorigo M., Maniezzo V. Distributed optimization by ant colonies. *Proceedings of ECAL91 - European Conference on Artificial Life*. 1991:134–142.

REFERENCES

1. Kovalev I.V., Kovalev D.I., Brit A.A., Saramud M.V. Informacionnaja tehnologija dlja mul'tiversionnogo formirovanija otkazoustojchivyh programmnyh sistem. *Sistemy upravleniya i informatsionnyye tekhnologii = Modern Innovations, Systems and Technologies*. 2021;2(84):56–68. DOI: 10.36622/VSTU.2021.84.2.013 (In Russ.)
2. Buhovcev D.D. Primenenie modifitsirovannogo algoritma murav'inoj kolonii dlja reshenija zadachi kalendarnogo planirovanija raspredelennyh predpriyatij. *Sovremennyye innovatsii, sistemy i tekhnologii = Modern Innovations, Systems and Technologies*. 2021;1(1):29–42. Available by: <https://doi.org/10.47813/2782-2818-2021-1-1-29-42> (In Russ.)
3. Saramud M.V. K voprosu prognozirovaniya vremeni narabotki na otkaz otkazoustojchivyh programmnyh kompleksov. *Materialy XXIV Mezhdunarodnoj nauchno-prakticheskoy konferencii, posvjashhennoj pamjati general'nogo konstruktora raketno-kosmicheskikh sistem akademika M. F. Reshetneva*. Krasnoyarsk, 2020:453–454. (In Russ.)
4. Saramud M. V., Kovalev D. I. Sredstvo avtomatizirovannogo proektirovanija mul'tiversionnogo programmno kompleksa. *Svidetel'stvo o registracii programmy dlja JeVM* 2021610550, 15.01.2021. *Zajavka № 2020667734*, 28.12.2020. (In Russ.)
5. Ning J., Zhang C., Sun P., Feng Y. Comparative study of ant colony algorithms for multi-objective optimization. *Information*. 2019;10(1):11.
6. Jeona Young-Jae Kimb, Jae-Chul, Yunc Sang-Yun, Leed Kwang Y. Application of ant colony algorithm for network reconfiguration in distribution systems. *IFAC Proceedings Volumes*. 2003;36(20):773–778.
7. Borisenkov D.S. Preobrazovanie struktury programmy na jetape proektirovanija. *Perspektivy razvitija informacionnyh tehnologij*. 2014;22:15–20. (In Russ.)
8. Ponachugin A.V. Opredelenie nadjozhnosti programmno obespechenija v strukture sovremennoj informacionnoj sistemy. *Kibernetika i programmirovanie*. 2019;2:65–72. (In Russ.)
9. Zenjutkin N.V., Kovalev D.I., Tuev E.V, Tueva E.V. O sposobah formirovanija informacionnyh struktur dlja modelirovanija objektov, sred i processov. *Modern Innovations, Systems and Technologies*. 2021;1(1):10–22. Available by: <https://doi.org/10.47813/2782-2818-2021-1-1-10-22> (In Russ.)
10. Gimarov V.V., Glushko S.I., Dli M.I. Primenenie algoritmov murav'inoj kolonii pri upravlenii slozhnymi proektami. *Transportnoe delo Rossii = Transport business of Russia*. 2012;4:107–109. (In Russ.)
11. Dorigo M., Stützle T. Ant colony optimization: Overview and recent advances. *Handbook of Metaheuristics. International Series in Operations Research & Management Science*. 2010;146:5–8. Available by: https://doi.org/10.1007/978-1-4419-1665-5_8.
12. Colomi A., Dorigo M., Maniezzo V. Distributed optimization by ant colonies. *Proceedings of ECAL91 - European Conference on Artificial Life*. 1991:134–142.

ИНФОРМАЦИЯ ОБ АВТОРАХ / INFORMATION ABOUT THE AUTHORS

Ковалев Игорь Владимирович, доктор технических наук, профессор, профессор кафедры информатики Сибирского федерального университета, Красноярск, Российская Федерация.

e-mail: kovalev.fsu@mail.ru

ORCID: [0000-0003-2128-6661](https://orcid.org/0000-0003-2128-6661)

Kovalev Igor Vladimirovitch, Doctor of Technical Sciences, Professor, Professor of Informatics Department of the Siberian Federal University, Krasnoyarsk, Russian Federation.

Ковалев Дмитрий Игоревич, аспирант кафедры информационных технологий и математического обеспечения информационных систем Красноярского государственного аграрного университета, Красноярск, Российская Федерация.
e-mail: grimm7jow@gmail.com
ORCID: [0000-0001-5308-308X](https://orcid.org/0000-0001-5308-308X)

Kovalev Dmitry Igorevitch, Post-graduate Student of the Department of Information Technologies and Mathematical Support of Information Systems of the Krasnoyarsk State Agrarian University, Krasnoyarsk, Russian Federation.

Амбросенко Николай Дмитриевич, кандидат технических наук, доцент кафедры информационных технологий и математического обеспечения информационных систем Красноярского государственного аграрного университета, Красноярск, Российская Федерация.
e-mail: nikolai.ambrosenko@yandex.ru

Ambrosenko Nikolai Dmitrievich, Candidate of Technical Sciences, Associate Professor of the Department of Information Technologies and Mathematical Support of Information Systems of the Krasnoyarsk State Agrarian University, Krasnoyarsk, Russian Federation.

Боровинский Дмитрий Владимирович, кандидат экономических наук, заведующий кафедрой государственного и муниципального управления Сибирской пожарно-спасательной академии ГПС МЧС России, Железнодорожск, Российская Федерация.
e-mail: b-dim@mail.ru

Borovinsky Dmitry Vladimirovich, Candidate of Economic Sciences, Head of the Department of State and Municipal Administration of the Siberian Fire and Rescue Academy EMERCOM of Russia, Zheleznogorsk, Russian Federation.

Статья поступила в редакцию 04.04.2022; одобрена после рецензирования 08.04.2022; принята к публикации 15.04.2022.

The article was submitted 04.04.2022; approved after reviewing 08.04.2022; accepted for publication 15.04.2022.