МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ, ЧИСЛЕННЫЕ МЕТОДЫ И КОМПЛЕКСЫ ПРОГРАММ / MATHEMATICAL MODELING, NUMERICAL METHODS AND PROGRAM COMPLEXES

DOI: https://doi.org/10.60797/IRJ.2024.150.47

РАСПАРАЛЛЕЛИВАНИЕ МЕТОДА МУРАВЬЕВ-ОПЫЛИТЕЛЕЙ ДЛЯ ЗАДАЧИ ПОСТРОЕНИЯ МОДЕЛИ АНАЛИЗА ВЫЖИВАЕМОСТИ

Научная статья

Микулик И.И.[1, *], Благовещенская Е.А.[2], Ходаковский В.А.[3]
[1] ORCID : 0000-0002-0542-7914;
[2] ORCID : 0000-0002-2425-5556;
[3] ORCID : 0000-0002-2060-4560;
[1, 2, 3] Петербургский государственный университет путей сообщения Императора Александра I, Санкт-Петербург, Российская Федерация

* Корреспондирующий автор (guess_97[at]mail.ru)

Аннотация

Прогностические модели играют ключевую роль в принятии решений в различных научных и прикладных областях, позволяя предсказать исход событий на основе входных параметров исследуемого объекта. Одним из таких направлений является анализ выживаемости – набор статистических методов, предназначенных для оценки вероятности наступления терминальных событий. В статье рассматривается задача построения прогностических моделей на основе гибридного метаэвристического алгоритма муравьев-опылителей. Исследуется эффективность распараллеливания данного алгоритма, что позволяет сократить время вычислений при обучении моделей. В результате работы продемонстрировано, что предложенный метод построения моделей анализа выживаемости может быть эффективно распараллелен. Распараллеливание ускоряет процесс обучения модели, что особенно важно в условиях многократного обучения промежуточных моделей. При этом, как показали эксперименты, зависимость эффективности распараллеливания от числа вычислительных узлов имеет нелинейный характер: увеличение числа узлов повышает скорость работы алгоритма, но с определённым пределом. Результаты показывают, что распараллеливание на небольшом числе вычислительных узлов является наиболее эффективным с точки зрения сокращения времени вычислений.

Ключевые слова: распараллеливание, оптимизация, анализ выживаемости, модель Кокса, муравьиный алгоритм.

PARALLELIZATION OF THE ANT-POLLINATOR METHOD FOR THE SURVIVAL ANALYSIS MODEL BUILDING PROBLEM

Research article

Mikulik I.I.[1, *], Blagoveshchenskaya Y.A.[2], Khodakovskii V.A.[3]
[1] ORCID : 0000-0002-0542-7914;
[2] ORCID : 0000-0002-2425-5556;
[3] ORCID : 0000-0002-2060-4560;
[1, 2, 3] Emperor Alexander I St. Petersburg State Transport University, Saint-Petersburg, Russian Federation

* Corresponding author (guess_97[at]mail.ru)

Abstract

Prognostic models play a key role in decision-making in various scientific and applied areas, allowing to predict the outcome of events based on the input parameters of the studied object. One of such areas is survival analysis – a set of statistical methods designed to estimate the probability of occurrence of terminal events. The paper addresses the problem of building predictive models based on a hybrid metaheuristic ant-pollination algorithm. The efficiency of parallelization of this algorithm is examined, which allows to reduce the computational time in training the models. The work demonstrates that the proposed survival analysis model building method can be effectively parallelized. Parallelization speeds up the model training process, which is especially important when intermediate models are repeatedly trained. Experiments show that the dependence of parallelization efficiency on the number of computational nodes is non-linear: increasing the number of nodes increases the speed of the algorithm, but with a certain limit. The results show that parallelization on a few computational nodes is the most effective in terms of reducing computation time.

Keywords: parallelization, optimization, survival analysis, Cox model, ant algorithm.

Введение

Прогностические модели играют важную роль в принятии решений в различных прикладных научных областях. Прогностические модели позволяют предсказать исход некоторого события по входным параметрам исследуемого объекта. Одним из ключевых направлений среди таких моделей является анализ выживаемости. Это набор статистических методов, которые позволяют оценить вероятность наступления терминального события, после которого объект перестает подлежать наблюдению. Анализ выживаемости применяется для моделирования и исследования распределения времени до наступления терминальных событий [1]. Модели и методы анализа выживаемости используются в медицине [2], биотехнологии [3], а также в прикладных задачах экономики и социологии [4].

Важной задачей является выбор и построение моделей по набору данных. Построение таких моделей возможно с помощью гибридных метаэвристических алгоритмов [5], одним из которых является алгоритм муравьев-опылителей. Задача построения модели является затратной по времени и ресурсам, так как алгоритм требует обучения модели при каждом найденном решении. Цель работы – исследование эффективности распараллеливания алгоритма муравьев-опылителей. Параллельная реализация позволяет сократить время построения модели, что является актуальным с точки зрения практического приложения алгоритма.

Методы и принципы исследования

2.1. Задача

Метод муравьев-опылителей позволяет строить функцию риска для модели анализа выживаемости по исходному набору данных.

Определим S – как набор входных данных, используемых для обучения модели. Зададим множество признаков $F = f_1, f_2, \ldots, f_n$ · $\hat{F}$ – подмножество признаков: $\hat{F} \subset F$ ·

Зададим произвольную функцию риска для модели $M_g(S, F)$ с функциональным ядром g, обученной на наборе данных S и с множеством признаков F в виде:

$$\lambda(t|\mathrm{X_i}) = \lambda_0(t)exp(g(\beta, \mathrm{X_i}))$$

Для оценки качества модели используется с-индекс [6].

Задача – распараллелить алгоритм A, входными данными которого являются набор данных S и множество признаков F, а результатом работы – модель M_g:

$$A : (S, F) \to M_g(S, \hat{F})|c \to \max \wedge |\hat{F}| \to \min$$

Функция риска является обобщением функции риска модели Кокса. При $g(\beta, X_i) = \beta \cdot X_i$ функция риска становится идентичной функции риска регрессионной модели Кокса.

2.2. Метод муравьев-опылителей

Метод муравьев-опылителей основан на гибридном методе муравьиной колонии и способен решать задачу построения функции выживаемости моделей анализа выживаемости [5]. Особенность алгоритма заключается в преобразовании множества вершин графа, которые представляют признаки или их комбинации в модели. Он имитирует процесс опыления цветковых растений с помощью насекомых-опылителей. Алгоритм состоит из трех компонентов: муравьиного алгоритма, который используется для построения модели; генетического алгоритма, предназначенного для оптимизации работы муравьиного алгоритма; и алгоритма опыления, который применяется для выбора признаков или их сочетаний.

Результатом работы алгоритма является расширенная модель Кокса, функцией риска которой является построенный полином $P_q(\hat{F})$:

$$P_q(\hat{F}) = \sum_{i=1}^{|\xi|} \varphi_i \prod_{j=1}^{|\hat{F}|} f_j^{\xi_{ij}} \ ,$$

где $\xi_{ij} \in \{0; 1\}$ – множество всех битовых векторов, каждый элемент которого – индикатор вхождения j-го признака в i-е слагаемое полинома, $\varphi_{ij} \in \{0; 1\}$ – маркер, указывающий вхождение i-го монома в P_q.

Ядро функции риска связано с полиномом $P_q(\hat{F})$ следующим образом:

$$g(\hat{\beta}, \mathrm{X_j}) = \sum_{\mathrm{j=1}}^{|\xi|} \hat{\beta}_\mathrm{j}\varphi_\mathrm{j} \prod_{\mathrm{k=1}}^{|\hat{\mathrm{F}}|} \mathrm{X}_{\mathrm{il}}^{\xi_{\mathrm{jk}}}|\mathrm{l} : \hat{\mathrm{F}}_\mathrm{k} = \mathrm{F_l}$$

Идея алгоритма заключается в следующем. Каждый моном из $P_q(\hat{F})$ представлен в алгоритме цветком. Множество цветов образует граф, путь по которому строят муравьи-опылители. Каждый муравей составляет множество цветов, сумма соответствующих мономов которых образует полином $P_q(\hat{F})$ ·

Муравьиный этап алгоритма использует идею простого муравьиного алгоритма [7], в котором переопределены правила откладывания феромона и выбора вершин в соответствии со спецификой задачи [5]. Вторым этапом гибридного алгоритма является приложение генетического алгоритма, используемого для оптимизации параметров муравьиного этапа.

Этап опыления основан на популяционной концепции и реализуется через применение четырех операторов к популяции цветков: селекции, кроссбридинга, лайнбридинга и старения. Каждый цветок имеет параметр – возраст. Оператор селекции выбирает цветы с наибольшей концентрацией феромонов. Оператор кроссбридинга с определенной вероятностью создает новые цветы, чьи мономы составлены из мономов цветов-предков. Оператор лайбридинга добавляет случайным образом в популяцию моном с единичным признаком, а оператора старения повышает возраст каждого из цветов. Если возраст цветка переступит некоторый порог, цветок погибает.

Алгоритм можно представить в виде шагов:

Начало

1. Определить параметры $n, \tau_0, \rho, o_max, \alpha_0, \beta_0, Q_0$
2. Положить $c = 0, P = \emptyset$
3. Положить множество цветов:

$$V = \{v_i = (e_i = f_i, \tau_i = rand(0, \tau_0), \eta_i = c(S, P_i \equiv f_i), o_i = o_{max})|\forall f_i \in F\}$$

4. Положить множество муравьев $A = \{a_k = (\alpha_k = \alpha_0, \beta_k = \beta_0, Q_k = Q_0)\}$
5. Пока не достигнут критерий остановки:

5.1. Для каждого муравья $a_k \in A$:

5.1.1. $E_k(t) = \{v_{random}\}$
5.1.2. $c_k(t-1) = 0$
5.1.3. $c_k(t) = \eta_i$
5.1.4. Пока $c_k(t) > c_k(t-1)$:
5.1.4.1. Выбрать v в соответствии с правилом выбора вершины
5.1.4.2. $E_k(t) = E_k(t) \cup \{v\}$
5.1.4.3. $c_k(t-1) = c_k(t)$
5.1.4.4. $P_k = \sum_{i,v_i \in E_k(t)} e_i$
5.1.4.5. $c_k(t) = f(S, P_k)$
5.1.5. Если $c_k(t) > c$:
5.1.5.1. $c = c_k(t)$
5.1.5.2. $P = P_k$
5.1.6. Для каждого $v_i \in E_k(t)$ вычислить $\Delta\tau_v(t)$
5.2. Применить оператор выбора
5.3. Применить оператор кроссинговера
5.4. Применить оператор мутации
5.5. Применить оператор селекции цветов
5.6. Применить оператор кроссбридинга
5.7. Применить оператор лайнбридинга
5.8. Применить оператор старения
6. Вернуть значения c, P

Критерием остановки алгоритма может являться количество итераций или сходимость решений к одному значению.

2.3. Распараллеливание метода

Одним из преимуществ многих метаэвристических алгоритмов является возможность их распараллеливания. Существует несколько способов оценивания распараллеливания.

Параллельное ускорение – это отношение скорости выполнения программы на нескольких вычислительных узлах к скорости выполнения программы на одном вычислительном узле (1):

$$S(p) = \frac{V(p)}{V(1)} \quad (1)$$

где $V(p)$ – средняя скорость выполнения программы на p кластерах.

Также используется другой показатель, который позволяет получить оценку эффективности распараллеливания с учётом количества вычислительных узлов – параллельная эффективность (2):

$$E(p) = \frac{S(p)}{p} \quad (2)$$

Чем ближе $E(p)$ к единице, тем эффективнее является распараллеливание [8].

Известно, что муравьиные алгоритмы имеют параллельную реализацию [9], так как поиск решения отдельного муравья одной итерации не зависит от результатов поиска других муравьев. Поэтому муравьиные алгоритмы эффективно распараллеливаются по данным. Однако рассматриваемый метод является гибридным, а значит, существует возможность появления новых зависимостей, не поддающихся распараллеливанию. Так как в рассматриваемом алгоритме наиболее затратным действием является обучение модели, оно положено за единицу вычисления для определения скорости выполнения программы в $S(p)$ в формуле 2.

Схема распараллеливания алгоритма муравьев-опылителей представлена на рисунке 1.

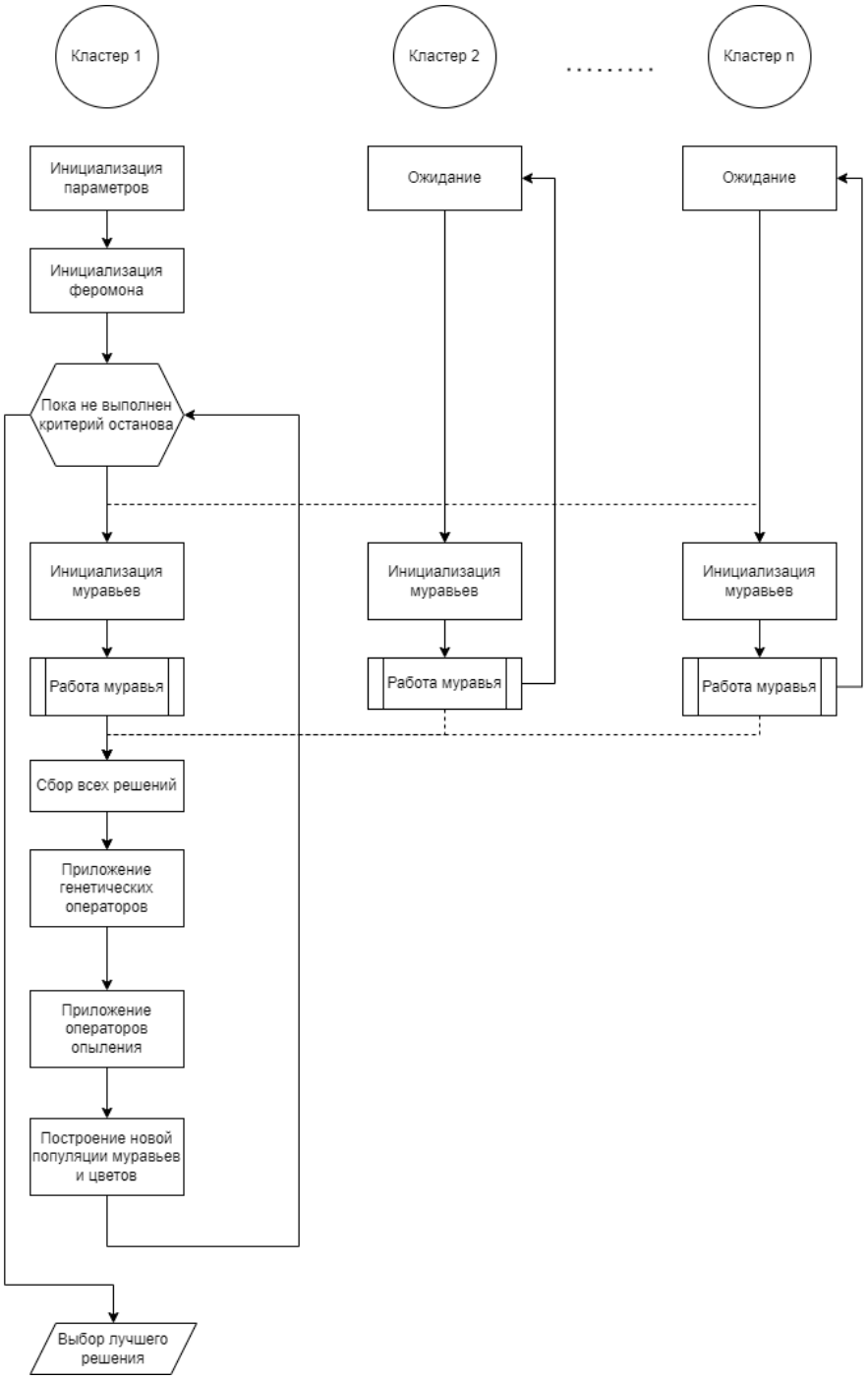

Рисунок 1 - Схема распараллеливания алгоритма муравьев-опылителей

DOI: https://doi.org/10.60797/IRJ.2024.150.47.1

В рисунке 1 шаг «работа муравья» может быть уточнен. Блок-схема работы, которую выполняет муравей, указана на рисунке 2.

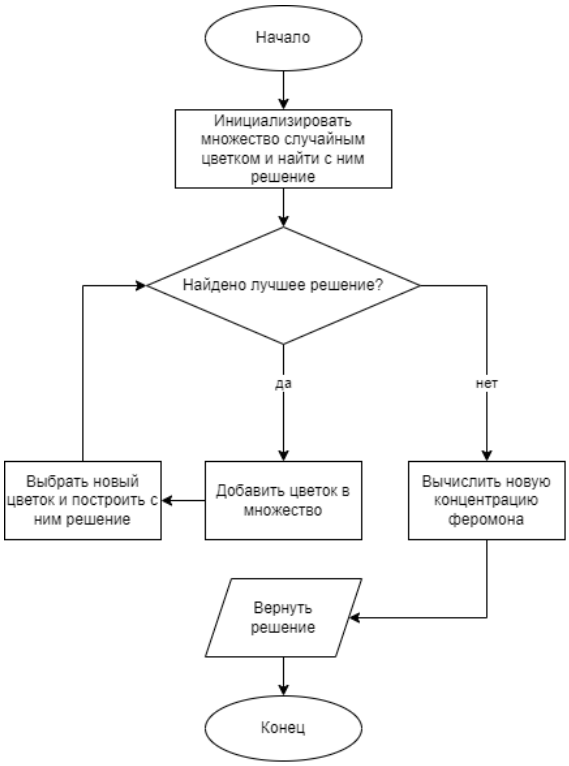

Рисунок 2 - Блок-схема операций, которые совершает муравей
DOI: https://doi.org/10.60797/IRJ.2024.150.47.2

Основные результаты

Распараллеливание осуществлялось с помощью MPI, стандартного интерфейса программирования для написания параллельных программ на кластерных компьютерах. Программа была запущена 50 раз при различных конфигурациях с разным количеством потоков.

В таблице 1 представлены результаты распараллеливания алгоритма муравьев-опылителей, в которой отражены время, скорость выполнения программы, а также оценочные характеристики.

Таблица 1 - Результаты распараллеливания алгоритма муравьев-опылителей на базе данных больных раком предстательной железы

DOI: https://doi.org/10.60797/IRJ.2024.150.47.3

Количество потоков	Среднее время выполнения алгоритма, с	Скорость выполнения программы	Параллельное ускорение	Параллельная эффективность
1	38,254	3,673	1	1
2	28,560	5,225	1,423	0,711
3	26,219	6,149	1,674	0,558
4	21,300	6,970	1,898	0,474
6	21,018	7,273	1,980	0,330
12	19,279	8,099	2,205	0,184

Тестирование проводилось на базе данных больных раком предстательной железы, наблюдавшихся, либо получавших лечение в ФГБУ «РНЦРХТ им. ак. А.М. Гранова» Минздрава России в период с января 1996 по декабрь 2016 года [10]. В исследование включены обезличенные данные 5073 пациентов, у которых была доступна информация о степени распространенности опухолевого процесса.

Обсуждение

Параллельное ускорение и параллельная эффективность в таблице 4 вычислены по формулам (2) и (3) соответственно. Из результатов распараллеливания видно, что дополнительные кластеры ускоряют работу выполнения программы, однако зависимость ускорения нелинейная. С увеличением количества кластеров уменьшается приток ускорения и снижается параллельная эффективность. Следовательно, алгоритм может быть распараллелен, однако вследствие того, что алгоритм имеет последовательную часть, рост параллельного ускорения с возрастанием количества кластеров снижается.

Заключение

Таким образом продемонстрировано, что предложенный метод построения моделей анализа выживаемости с выбором признаков может быть распараллелен. Распараллеливание является важной задачей, так как время обучения модели является затратным, а обучение промежуточных моделей происходит неоднократно.

Распараллеливание программы представлено нелинейной зависимостью как времени, так и скорости работы, от числа кластеров. С увеличением числа вычислительных узлов эффективность распараллеливания снижается. Запуск работы алгоритма на двух вычислительных узлах увеличил скорость работы алгоритма в 1,423 раза; на шести вычислительных узлах – в 1,98 раза; на двенадцати – в 2,205. Это показывает целесообразность распараллеливания алгоритма на сравнительно небольшом количестве вычислительных узлов.

Конфликт интересов

Не указан.

Рецензия

Все статьи проходят рецензирование. Но рецензент или автор статьи предпочли не публиковать рецензию к этой статье в открытом доступе. Рецензия может быть предоставлена компетентным органам по запросу.

Conflict of Interest

None declared.

Review

All articles are peer-reviewed. But the reviewer or the author of the article chose not to publish a review of this article in the public domain. The review can be provided to the competent authorities upon request.

Список литературы / References

1. George B. Survival analysis and regression models / B. George, S. Seals, I. Aban // J. Nucl. Cardiol. — 2014. — № 21 (4). — P. 686–694. — DOI: 10.1007/s12350-014-9908-2.

2. Purnami S. Cox Model Survival Analysis to Evaluate Treatment of Electro-Capacitive Cancer Therapy (ECCT) For Cancer Patients / S. Purnami, R. Putra, A. Edina [et al.] // Journal of Physics: Conference Series. — 2021. — 1863. — 012036. — DOI: 10.1088/1742-6596/1863/1/012036.

3. Dokoumetzidis A. Modelling and simulation in drug absorption processes / A. Dokoumetzidis, G. Valsami, P. Macheras // Xenobiotica. — 2007. — Vol. 37. — №. 10-11. — P. 1052–1065.

4. Зозуленко К.С. Математические модели определения цены опционов / К.С. Зозуленко // Современные проблемы и тенденции развития экономики и управления. — Казань: АЭТЕРНА, 2018.

5. Микулик И.И. Методика для решения задачи выбора признаков в регрессионной модели Кокса / И.И. Микулик // Вестник астраханского государственного технического университета. Серия: управление, вычислительная техника и информатика. — 2024. — № 3. — С. 85–94.

6. Harrell F.E. Evaluating the yield of medical tests / F.E. Harrell, R.M. Califf, D.B. Pryor [et al.] // JAMA. — 1982. — № 247 (18). — P. 2543–2546.

7. Liu G. A sequential excitation and simplified ant colony optimization based global extreme seeking control method for performance improvement / G. Liu, Y. Bai, L. Zhu [et al.] // Swarm and Evolutionary Computation. — 2024. — № 86.

8. Соснин В.В. Введение в параллельные вычисления / В.В. Соснин, П.В. Балакшин, Д.С. Шилко [и др.]. — Санкт-Петербург: Университет ИТМО, 2023. — 128 с.

9. Stützle T. Parallelization strategies for ant colony optimization / T. Stützle. — Berlin: Springer Berlin Heidelberg, 1998.

10. Жаринов Г.М. База данных больных раком предстательной железы / Г.М. Жаринов // База данных РФ № 2016620331. — 2016. — https://elibrary.ru/download/elibrary_39345961_80608804.pdf (дата обращения: 13.09.2024)

Список литературы на английском языке / References in English

1. George B. Survival analysis and regression models / B. George, S. Seals, I. Aban // J. Nucl. Cardiol. — 2014. — № 21 (4). — P. 686–694. — DOI: 10.1007/s12350-014-9908-2.

2. Purnami S. Cox Model Survival Analysis to Evaluate Treatment of Electro-Capacitive Cancer Therapy (ECCT) For Cancer Patients / S. Purnami, R. Putra, A. Edina [et al.] // Journal of Physics: Conference Series. — 2021. — 1863. — 012036. — DOI: 10.1088/1742-6596/1863/1/012036.

3. Dokoumetzidis A. Modelling and simulation in drug absorption processes / A. Dokoumetzidis, G. Valsami, P. Macheras // Xenobiotica. — 2007. — Vol. 37. — №. 10-11. — P. 1052–1065.

4. Zozulenko K.S. Matematicheskie modeli opredelenija tseny optsionov [Mathematical models for option pricing] / K.S. Zozulenko // Modern problems and trends in the development of economics and management. — Kazan': AETERNA, 2018. [in Russian]

5. Mikulik I.I. Metodika dlja reshenija zadachi vybora priznakov v regressionnoj modeli Koksa [Methodology of solving the feature selection problem for the cox regression model] / I.I. Mikulik // Bulletin of Astrakhan State Technical University. Series: Management, Computer Science and Informatics. — 2024. — № 3. — P. 85–94. [in Russian]

6. Harrell F.E. Evaluating the yield of medical tests / F.E. Harrell, R.M. Califf, D.B. Pryor [et al.] // JAMA. — 1982. — № 247 (18). — P. 2543–2546.

7. Liu G. A sequential excitation and simplified ant colony optimization based global extreme seeking control method for performance improvement / G. Liu, Y. Bai, L. Zhu [et al.] // Swarm and Evolutionary Computation. — 2024. — № 86.

8. Sosnin V.V. Vvedenie v parallel'nye vychislenija [Introduction to Parallel Computing] / V.V. Sosnin, P.V. Balakshin, D.S. Shilko [et al.]. — Sankt-Peterburg: ITMO University, 2023. — 128 p. [in Russian]

9. Stützle T. Parallelization strategies for ant colony optimization / T. Stützle. — Berlin: Springer Berlin Heidelberg, 1998.

10. Zharinov G.M. Baza dannyh bol'nyh rakom predstatel'noj zhelezy [Prostate cancer patient database] / G.M. Zharinov // Baza dannyh RF № 2016620331 [Database of the Russian Federation No. 2016620331]. — 2016. — https://elibrary.ru/download/elibrary_39345961_80608804.pdf (accessed: 13.09.2024) [in Russian]