

**МЕТОД РАСПРЕДЕЛЕНИЯ И ПЛАНИРОВАНИЯ ВЫПОЛНЕНИЯ ЗАДАЧ
АГЕНТАМИ РОЕВЫХ РОБОТОТЕХНИЧЕСКИХ СИСТЕМ
В УСЛОВИЯХ НЕДЕТЕРМИНИРОВАННОЙ СРЕДЫ¹**

Статья поступила в редакцию 10.08.2022, в окончательном варианте – 10.08.2022.

Петренко Вячеслав Иванович, Северо-Кавказский федеральный университет, 355017, Российская Федерация, г. Ставрополь, ул. Пушкина, 1,
кандидат технических наук, заведующий кафедрой, ORCID: 0000-0003-4293-7013, e-mail: vipetrenko@ncfu.ru

Тебужева Фариза Биляловна, Северо-Кавказский федеральный университет, 355017, Российская Федерация, г. Ставрополь, ул. Пушкина, 1,
доктор физико-математических наук, заведующий кафедрой, ORCID: 0000-0002-7373-4692, e-mail: ftebueva@ncfu.ru

Павлов Андрей Сергеевич, Северо-Кавказский федеральный университет, 355017, Российская Федерация, г. Ставрополь, ул. Пушкина, 1,
старший преподаватель, ORCID: 0000-0002-8413-8706, e-mail: losde5530@gmail.com

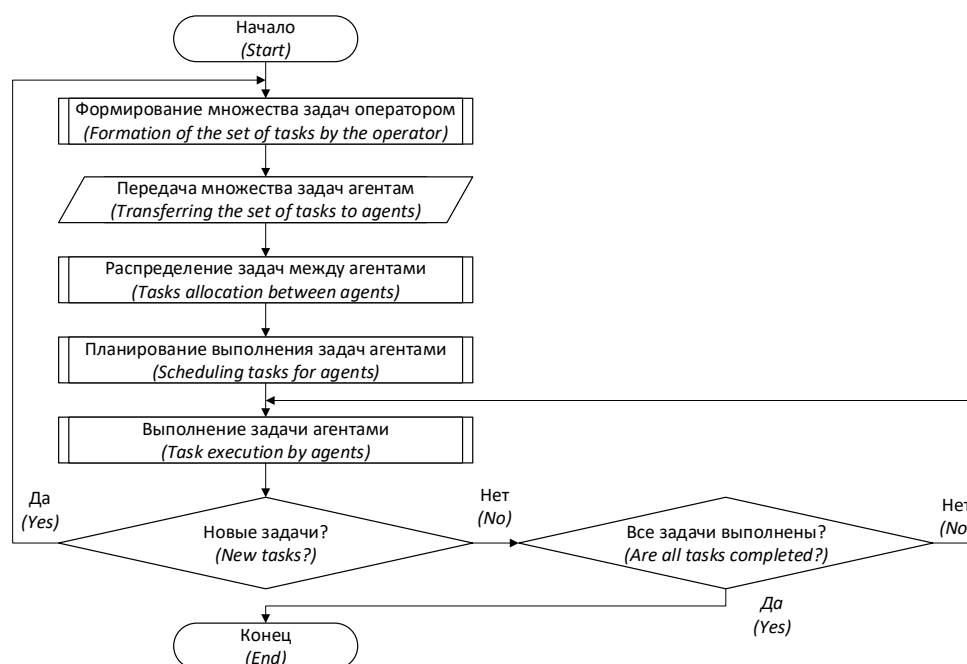
Гурчинский Михаил Михайлович, Северо-Кавказский федеральный университет, 355017, Российская Федерация, г. Ставрополь, ул. Пушкина, 1,
аспирант, ORCID: 0000-0002-1739-2624, e-mail: gurmikhail@gmail.com

Применение роевых робототехнических систем в условиях недетерминированной среды актуализирует вопросы разработки соответствующих методов и алгоритмов распределения и планирования выполнения задач. Под условиями недетерминированной среды в работе понимается такая ситуация, когда максимальное количество задач лимитировано, а сразу после выполнения первой из них появляется новая задача, то есть происходит динамическое изменение списка задач в процессе функционирования роевых робототехнических систем. При функционировании роевых робототехнических систем в условиях недетерминированной среды существующие методы и алгоритмы не позволяют оптимально распределить задачи между всеми роботами системы и запланировать последовательность выполнения задач, закрепленных за каждым из роботов. Помимо этого, известные методы распределения и планирования задач не учитывают ограничения сенсорных и вычислительных возможностей робототехнических устройств, используемых в составе роевых робототехнических систем (например, малый объем оперативной памяти, низкая тактовая частота процессора, малая емкость аккумуляторной батареи, низкая производительность бортовых датчиков и сенсоров и т.д.). Также стоит отметить, что не все известные методы, направленные на решение указанной задачи, учитывают специфику децентрализованного управления роевыми робототехническими системами, заключающуюся в ограниченной области видимости, в результате чего их применение в реальных сценариях использования роевых робототехнических систем сопряжено со значительными проблемами. Целью работы является повышение эффективности распределения и планирования выполнения задач в роевых робототехнических системах в условиях недетерминированной среды с учетом ограниченных возможностей элементов роевых робототехнических систем и специфики децентрализованного управления. Решение задачи выполнено с использованием методов системного анализа, аналитической геометрии и искусственных нейронных сетей. Элементом научной новизны являются предложенные алгоритмы сортировки задач и поиска транзитных задач, обеспечивающие повышение эффективности планирования и распределения задач в роевых робототехнических системах в условиях недетерминированной среды с учетом ограниченных возможностей элементов роевых робототехнических систем. Предложенный метод отличается от известных методов алгоритмом сортировки приоритета выполнения задач в виде связанного списка, что дает возможность осуществлять масштабирование численности агентов роевых робототехнических систем при динамическом изменении списка актуальных задач. Еще одним отличием является процедура распределения задач между агентами роевых робототехнических систем, позволяющая искать промежуточные задачи для выполнения, что обеспечивает уменьшение общего времени выполнения задач по сравнению с аналогичными решениями. На основе предложенного метода в работе представлена нейросетевая модификация этого метода, отличающаяся учетом специфики децентрализованного управления. Представленное решение программно реализовано на языке Python и может быть использовано при моделировании децентрализованных систем управления роевых робототехнических системами.

Ключевые слова: роевые робототехнические системы, распределение задач, планирование последовательности выполнения задач, искусственные нейронные сети

¹ Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 20-37-90026.

Графическая аннотация (Graphical annotation)



**METHOD FOR TASKS ALLOCATION AND PLANNING
THE SEQUENCE OF PERFORMING TASKS BY AGENTS
OF SWARM ROBOTIC SYSTEMS UNDER UNCERTAINTY**

The article was received by the editorial board on 10.08.2022, in the final version – 10.08.2022.

Petrenko Vyacheslav I., North Caucasian Federal University, 1 Pushkin St., Stavropol, 355017, Russian Federation,

Cand. Sci. (Engineering), Head of Department, ORCID: 0000-0003-4293-7013, e-mail: vipetrenko@ncfu.ru

Tebueva Fariza B., North Caucasian Federal University, 1 Pushkin St., Stavropol, 355017, Russian Federation,

Doct. Sci. (Physics and Mathematics), Head of Department, ORCID: 0000-0002-7373-4692, e-mail: fbtebueva@ncfu.ru

Pavlov Andrey S., North Caucasian Federal University, 1 Pushkin St., Stavropol, 355017, Russian Federation,

Senior Lecturer, ORCID: 0000-0002-8413-8706, e-mail: losde5530@gmail.com

Gurchinskiy Mikhail M., North Caucasian Federal University, 1 Pushkin St., Stavropol, 355017, Russian Federation,

postgraduate student, ORCID: 0000-0002-1739-2624, e-mail: gurmikhail@gmail.com

The use of swarm robotic systems in a non-deterministic environment actualizes the issues of developing appropriate methods and algorithms for distributing and scheduling tasks. Under the conditions of a non-deterministic environment, we mean such a situation when the maximum number of tasks is limited, and immediately after the first task is completed, a new task appears, that is, the list of tasks changes dynamically during the operation of the swarm robotic systems. When swarm robotic systems operate in a non-deterministic environment, the existing methods and algorithms do not allow optimal distribution of tasks between all robots of the system and plan the sequence of tasks assigned to each of the robots. In addition, the known methods of task distribution and scheduling do not take into account the limitations of the sensory and computing capabilities of robotic devices used in the swarm robotic systems (for example, a small amount of RAM, low processor clock frequency, low battery capacity, low performance of onboard sensors and sensors, etc.). It is also worth noting that not all known methods aimed at solving this problem take into account the specifics of the decentralized control of swarm robotic systems, which consists in a limited scope, because of which their use in real scenarios of using swarm robotic systems is associated with significant problems. The aim of the work is to increase the efficiency of distribution and planning of tasks in the swarm robotic systems in a non-deterministic environment, taking into account the limited capabilities of the elements of the swarm robotic systems and the specifics of decentralized control. The problem was solved using the methods of system analysis, analytical geometry and artificial neural networks. An element of scientific novelty is the proposed algorithms for sorting tasks and searching for transit tasks, which provide an increase in the efficiency of planning and distribution of tasks in swarm robotic systems

in a non-deterministic environment, taking into account the limited capabilities of swarm robotic systems elements. The proposed method differs from the known methods by the sorting algorithm of task execution priority in the form of a linked list, which makes it possible to scale the number of swarm robotic systems agents with a dynamic change in the list of urgent tasks. Another difference is the procedure for distributing tasks between swarm robotic systems agents, which makes it possible to search for intermediate tasks to be performed, which reduces the total task execution time compared to similar solutions. Based on the proposed method, the paper presents a neural network modification of this method, which differs by taking into account the specifics of decentralized control. The presented solution is programmatically implemented in Python and can be used in modeling decentralized control systems of swarm robotic systems.

Keywords: swarm robotic systems, task allocation, task scheduling, artificial neural networks

Введение. В настоящее время в групповой робототехнике доминируют системы управления, построенные по принципу централизованной стратегии управления [1]. Данная стратегия предполагает, что каждому роботу (далее «агенту») центральное вычислительное устройство назначает задачи, а также в каждый дискретный момент времени формирует управляющее воздействие для исполнительных устройств агента. Типичным примером приложения групповой робототехники, в том числе роевых робототехнических систем (РРТС), является автоматизированный склад. Схематическое представление централизованной системы управления РРТС представлено на рисунке 1.

Достоинством централизованных систем управления является высокая эффективность выполнения задач агентами при их ограниченном количестве, а также при функционировании в строго детерминированной среде. Однако увеличение численности используемых агентов РРТС требует увеличения ресурсов каналов связи, что уменьшает эффективность выполнения задач. Также стоит отметить, что масштабирование существующей системы управления требует привлечения дополнительных вычислительных ресурсов, способных обеспечить требуемый объем расчетов для поддержания возможности формирования управляющих инструкций для каждого агента в режиме реального времени.



Рисунок 1 – Схема централизованной стратегии управления

Представленные недостатки централизованных систем управления группой роботов привели к переходу исследователей и разработчиков к частичному или полному автономному управлению. Ярким примером такого перехода является автоматизированный склад компании Alibaba, Китай [2]. В основу системы управления агентами положена смешанная стратегия управления, включающая в себя элементы децентрализации. Так, часть функционала системы управления выполняется на центральном вычислительном устройстве, например, формирование списка актуальных задач, мониторинг выполнения задач и т.д. Другая часть функционала системы управления переносится на вычислительные устройства агентов, например, распределение задач, планирование пути и траектории движения и т.д. Представленный подход существенно уменьшает нагрузку на каналы связи, а также позволяет распределить вычисления между всеми узлами системы. Схематическое представление смешанной системы управления представлено на рисунке 2.



Рисунок 2 – Схема смешанной стратегии управления

Одной из сложнейших задач в смешанных и децентрализованных системах управления является оптимальное распределение задач между всеми агентами системы [3–6]. Большинство известных методов и алгоритмов распределения задач предназначены для использования в детерминированной среде с ограниченным количеством задач. В рассмотренном примере автоматизированного склада Alibaba среда функционирования агентов является недетерминированной, так как одновременно функционирует множество агентов и людей, а список задач изменяется динамически в процессе функционирования РРТС.

В данных условиях интенсификация исследований в области решения задачи распределения задач позволит увеличить эффективность выполнения задач по транспортировке груза агентами в недетерминированной среде с учетом таких показателей эффективности, как время выполнения задачи, длина пути, энергоэффективность, функциональная безопасность и т.д. Дополнительно стоит отметить необходимость разработки таких решений, которые позволят не только оптимальным образом распределить актуальные задачи между агентами, но и оптимизировать последовательность их выполнения.

Таким образом, наблюдается противоречие в науке и практике, которое заключается в необходимости комплексного подхода для повышения эффективности функционирования робототехнической системы с одной стороны и недостаточным уровнем развития научно-методического аппарата распределения и планирования задач с другой стороны, чем и определяется актуальность исследований в данной области.

Социально-экономический эффект от решения представленной задачи состоит в том, что оптимальное распределение и планирование выполнения задач группой роботов позволит увеличить количество транспортировок посылок за один цикл работы от аккумуляторной батареи робота путем уменьшения суммарного пройденного расстояния каждым роботом системы.

С точки зрения автоматизированного склада решение рассмотренной задачи позволит снизить экономические издержки за счет следующих аспектов:

- отсутствие необходимости в покупке и обслуживании высокопроизводительных средств связи и центральных вычислительных устройств для управления робототехнической системой;
- увеличение объема выполненной работы при сохранении уровня издержек на обслуживание и ремонт робототехнических устройств и их компонентов.

1. Актуальное состояние проблемы. Анализ работ в области распределения задач позволяет говорить о большом многообразии теоретических методов распределения задач, особенно при равном количестве агентов и подзадач. Среди множества методов можно выделить эвристические алгоритмы [7], аналитические алгоритмы [8], алгоритмы на основе моделей рыночной экономики [9, 10], методы на основе потенциальных полей [11, 12], вероятностные алгоритмы [13, 14], методы на основе машинного обучения и искусственных нейронных сетей [15, 16], методы нечеткой логики [17], муравьиные алгоритмы [18, 19], методы динамического и целочисленного программирования [20, 21], генетические алгоритмы [22], смешанные алгоритмы [23].

В работе [24] представлен централизованный алгоритм распределения задач между агентами РРТС с целью автоматизации логистических работ на автоматизированном складе. Основной упор авторы сделали на обеспечении возможности масштабирования количества задач, исходя из чего, в работе предложен эвристический обобщенный алгоритм, включающий в себя алгоритмы кластеризации и маршрутизации (планирования пути). Аналогичный подход применили авторы работы [25], но для децентрализованного распределения задач. В работе [26] рассматривается задача распределения задач при транспортировке объектов в автоматизированном складе. Авторы предложили централизованный эвристический алгоритм, позволяющий оптимизировать как длину траектории движения агентов, так и время их перемещения с учетом возможных коллизий (столкновений) с другими агентами. Недостатком данного алгоритма является необходимость наличия центрального устройства, на котором осуществляются все расчеты по распределению задач, а также устойчивый канал связи между всеми агентами и центральным вычислительным устройством, что не всегда можно обеспечить в реальных условиях. В работе [27] представлен модифицированный алгоритм распределения задач на основе алгоритма пчелиной колонии [28], предназначенный для минимизации времени выполнения задания в условиях отсутствия информации о задачах. Основным недостатком данного алгоритма является допущение авторов о стационарности агентов, из-за чего его применение без дальнейшей модификации существенно ограничено.

Методом-аналогом можно считать работу [29], в которой предложен эвристический метод планирования выполнения задач по принципу минимизации штрафа за просроченное выполнение задач. В основе принципа минимизации штрафа лежит идея разделения множества задач на два подмножества: первое подмножество включает те задачи, которые могут быть назначены одному из агентов РРТС и выполнены в срок. Во второе подмножество задач входят все те задачи, которые не удовлетворяют данному условию, то есть результат их выполнения неизбежно приведет к получению штрафа. Данный метод состоит из двух этапов. На первом этапе выполняется распределение первого подмножества задач между всеми агентами системы по критерию минимального остатка времени до дедлайна (метка времени, до которой задача должна быть выполнена без получения штрафа). Вторым этапом является оценка штрафа для каждой задачи второго подмножества (формула (8) оригинальной работы) и назначение приоритета этой задачи. Далее на основе рассчитанных приоритетов задач они назначаются агентам системы для выполнения. Таким образом, в результате

распределения и планирования задач каждый агент имеет список закрепленных за ним задач в виде очереди, то есть первая закрепленная задача будет выполнена первой.

Необходимо отметить, что данный метод предназначен для централизованных систем, то есть все вычисления и назначение списка задач агентам выполняются на центральном вычислительном устройстве. Данный факт делает неприменимым этот метод для РРТС в классическом его понимании [30], либо требует модификации метода путем добавления элементов децентрализации. С другой стороны, авторами оригинальной работы упоминается, что в результате выполнения второго этапа метода один или несколько агентов могут быть «перегружены», то есть время выполнения списка задач закрепленных за такими агентами задач существенно больше среднего времени выполнения списка задач другими агентами. Описанная ситуация требует дополнительного перераспределения задач между агентами, в результате чего может быть получена неоптимальная последовательность выполнения задач по одному из критериев оценки эффективности функционирования РРТС.

Исходя из проведенного обзора, повышение эффективности распределения задач в РРТС в условиях недетерминированной среды является актуальной и своевременной задачей.

2. Постановка задачи. Дано: РРТС численностью n агентов r_i , $R = r_1, r_2, \dots, r_n$; O – множество, содержащее m элементарных задач m (далее просто «задач»), $O = o_1, o_2, \dots, o_m$; Y – множество выходных параметров (количество выполненных задач агентами РРТС); Z – множество внутренних параметров РРТС (текущая позиция, заряд батареи, скорость, ускорение и т.д.); E – множество параметров среды; Q – множество показателей качества функционирования РРТС, например, время выполнения множества задач O , длина пути, пройденная агентами, суммарный расход аккумуляторной батареи агентов и т.д.

Целью данной работы является повышение эффективности планирования задач в РРТС в условиях неполноты информации о задании с учетом ограниченных возможностей элементов РРТС и специфики децентрализованного управления. Содержательная (вербальная) постановка научной задачи: разработать метод M повышения качества/эффективности функционирования РРТС R по показателям $q_1, \dots, q_k$ в диапазоне значений входных и выходных параметров (T, Y) системы за счет варьирования значений ее внутренних параметров Z и при ограничениях на значения параметров среды $E \in E_{\text{доп}}$. Формальная постановка научной задачи: найти метод M такой(-ую), что

$$M : (R, T, Y, Z, E, Q) \rightarrow \{\Delta q_1, \dots, \Delta q_k\} \mid \forall \Delta q_i > 0, q_i \in Q, i = \overline{1, k}, \quad (1)$$

при этом $\Delta q_i = q_i^{\text{п}} - q_i^{\text{д}}$, $i = \overline{1, k}$, где индекс «д» значит «до использования метода», индекс «п» – «после использования метода».

3. Методы и материалы. Каждая задача $o_j \in O, j = \overline{1, m}$ представляет собой вектор $[x_j, y_j, \hat{x}_j, \hat{y}_j, t_j, f_j]$, где x_j, y_j – координаты задачи o_j ; $\hat{x}_j, \hat{y}_j$ – координаты места, куда необходимо транспортировать груз; t_j – метка времени, к которой задача должна быть выполнена (дедлайн); f_j – коэффициент для расчета штрафа за выполнение задачи после дедлайна. Значение штрафа F_j за невыполнение задачи в срок может быть вычислено следующим образом:

$$F_j = (c_j - t_j) f_j, \quad (2)$$

где c_j – время завершения выполнения задачи o_j . Соответственно, агенты РРТС получают штраф только в том случае, если задача o_j будет выполнена после дедлайна, то есть $c_j > t_j$.

В качестве среды функционирования РРТС рассматривается автоматизированный склад. На рисунке 3 представлен пример среды для проведения последующих экспериментов с 5 задачами. Цветные круги определяют исходные позиции задач, а крестики – позиции, в которые необходимо переместить товар в рамках выполнения задачи. Вся карта разделена на ряд областей:

- область получения товаров (ПТ). Предполагается, что здесь осуществляется разгрузка и прием товаров, а также их первоначальное размещение на стеллажах;
- область сортировки товаров (СТ). Эта область содержит ряды со стеллажами, на которых размещены товары;
- область комплектации заказов (КЗ). В данной области оператор собирает заказ, поочередно собирая нужные товары со стеллажей;
- область отгрузки заказов (ОЗ). В эту область осуществляется перемещение уже собранных заказов для дальнейшей их передачи курьерской службе.

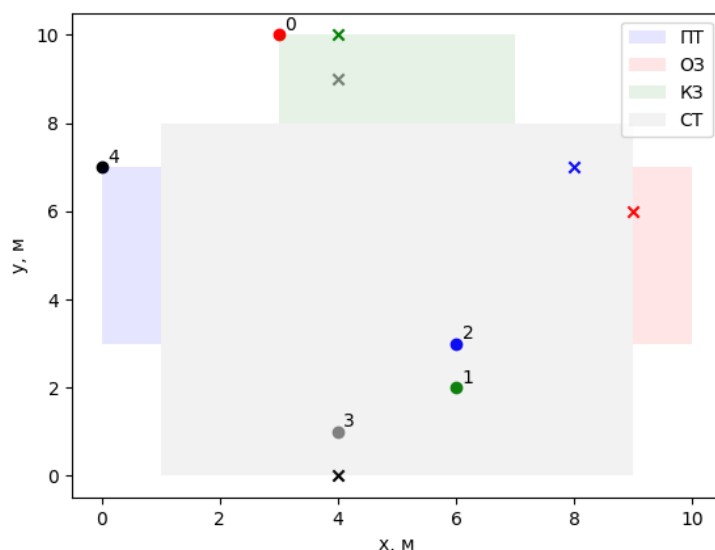


Рисунок 3 – Пример исходных данных для проведения эксперимента

В каждой из представленных областей положения стеллажей являются фиксированными, так как в противном случае проезд агента со стеллажом может быть перекрыт. Исходя из этого, начальные и конечные координаты задач также являются детерминированными (пример разметки автоматизированного склада представлен на рисунке 4).

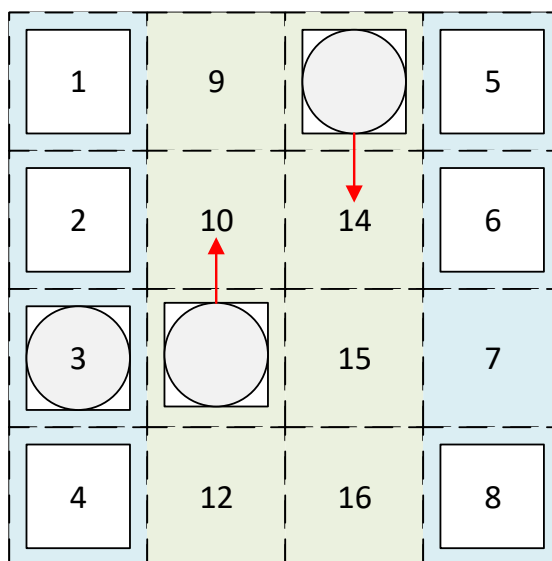


Рисунок 4 – Пример разметки автоматизированного склада

На рисунке 4 представлен пример организации расположения стеллажей (клетки 1–8 голубого цвета) для хранения продукции и дорожек для перемещения агентов (клетки 9–16 зеленого цвета). Подобные обозначения позиций для размещения стеллажей могут использоваться при оборудовании штрих-кодами, RFID-метками и т.д. для идентификации нужной позиции стеллажа. Движение агентов может быть организовано двухполосным (красными стрелками показано направление движения агентов), для того чтобы не создавать заторы, а агенты имели достаточно места для маневра при захвате и отгрузке стеллажей. Также стоит отметить, что если некоторый агент не функционирует, например, на момент его запуска, то его позиция также является строго детерминированной (специально выделенные позиции для зарядных станций, пункта обслуживания и ремонта и т.д.). Таким образом, всю карту автоматизированного склада можно представить в виде поля, разбитого на клетки, где совокупность этих клеток представляет собой множество P , содержащее как позиции задач $p_j = x_j, y_j, j = \overline{1, m}$, так и m' позиций перемещения агентов РПТС $p_r = x_r, y_r, r = \overline{m+1, m'}$ (приведенные координаты x_j, y_j и x_r, y_r являются координатами центров клеток).

3.1. Аналитический метод планирования последовательности выполнения задач агентами РРТС. В работе [29] предложен централизованный метод планирования выполнения задач на основе итерационного эвристического алгоритма. Однако применение этого метода невозможно в случае гибридной архитектуры системы управления автоматизированного склада, что требует модификации этого решения путем внедрения элементов децентрализации. Помимо этого, как отмечалось ранее, один или несколько агентов РРТС в результате распределения задач могут быть «перегружены». Такая ситуация, с одной стороны, требует непрерывного мониторинга результатов распределения задач на центральном вычислительном устройстве, а с другой стороны, вызывает необходимость повторения процедуры распределения и планирования выполнения задач. Аналогичная ситуация возникает и в том случае, если происходит увеличение численности агентов – для каждого нового агента системы потребуется перераспределение задач, что может занимать достаточно времени, особенно при значительном количестве новых агентов.

Исходя из этого, в данной работе предлагается идея разработки итерационного алгоритма для планирования ограниченного количества задач (не более двух для каждого агента). Аналогично методу-аналогу предлагаемый метод может быть разделен на два этапа: первый предполагает предобработку исходного множества задач, а второй – распределение задач между агентами.

3.1.1. Алгоритм предобработки исходного множества задач. Согласно вышеописанной идее работы, последовательность задач, закрепленных за агентом r_i , должна состоять не более чем из 2 задач. Процедуру попарного планирования задач предлагается реализовать по аналогии со связным списком, т.е. текущий элемент списка явно указывает на следующий. Для этого сначала предлагается осуществить на стороне центрального вычислительного устройства предобработку списка задач путем их сортировки по возрастанию по значению дедлайна t_j . Т.е. необходимо выполнить перестановку элементов множества O в таком порядке, при котором для заданной функции упорядочения S справедливо соотношение:

$$S(t_1) \leq S(t_2) \leq \dots \leq S(t_m). \quad (3)$$

В результате выполнения сортировки будет сформирован список L , в самом начале которого расположен ряд задач, которые должны быть выполнены в первую очередь. С учетом того, что количество агентов РРТС n ограничено, то первые n задач списка L могут быть распределены между агентами РРТС таким образом, чтобы минимизировать значение штрафа за просроченное выполнение задач или избежать его совсем. Назовем первые n задач списка L «горящими».

Следующим шагом предобработки списка задач является поиск для каждой задачи o_j списка L ближайшей задачи в геометрическом пространстве и добавление ее в качестве «указателя» o_j^{ptr} к задаче o_j . При этом в качестве указателя o_j^{ptr} могут рассматриваться только те задачи, которые располагаются в списке L после задачи o_j , для того, чтобы соблюдать порядок выполнения задач в соответствии с их дедлайнами:

$$\forall o_j: \min(D(o_j, o_l)) \Rightarrow o_j^{ptr} = o_l; l = \overline{j+1, m}; j = \overline{1, m}, \quad (4)$$

где $D(o_j, o_l)$ – функция вычисления расстояния между двумя задачами.

Таким образом, список задач L преобразуется в связный список, в результате чего каждый агент r_i при распределении задач будет выбирать только одну горящую задачу. А после ее выполнения – перейдет к выполнению задачи-указателя. Согласно выражению (4), последняя задача o_m списка L не будет иметь указателя, что должно быть предусмотрено на следующем этапе предлагаемого метода. После того как связный список L сформирован, оператор передает его по каналу связи агентам РРТС для дальнейшего распределения задач.

Блок-схема этапа предобработки списка задач представлена на рисунке 5.

Таким образом, отличием первого этапа предложенного метода от метода-аналога является то, что:

- выполняется обработка только исходного множества задач без их распределения между агентами со стороны центрального вычислительного устройства;
- обработка исходного множества задач предполагает не только их сортировку в порядке возрастания по дедлайну, но и формирование связного списка по принципу геометрической близости задач;
- при необходимости предложенная реализация вычисления близости задач может включать в себя множество критериев, которые позволят достигнуть требуемых значений показателей эффективности функционирования РРТС.



Рисунок 5 – Блок-схема этапа предобработки исходного множества задач

3.1.2. Алгоритм распределения и планирования последовательности выполнения задач агентами РРТС. Распределение задач между агентами предлагается реализовать на основе жадного алгоритма [31]. Каждый агент r_i выбирает для выполнения такую горящую задачу o_j , которая находится ближе всего к агенту. Необходимо отметить, что этап распределения задач выполняется децентрализованно, и соответственно, после выбора той или иной задачи каждый агент должен оповестить соседних агентов о том, что эта задача уже занята. Информацию о занятой задаче агенты заносят в собственную память для того, чтобы в последующем не возникало коллизий (выбор одной и той же задачи несколькими агентами).

После выполнения горячей задачи каждый агент оповещает других агентов о завершении текущей задачи. Соответственно, после этого агент готов приступить к выполнению задачи-указателя. Однако необходимо отметить следующую особенность. На этапе предобработки списка задач предполагалось, что задачей-указателем может быть только самая ближайшая задача. При этом дедлайн задачи-указателя t_j^{ptr} может быть настолько велик, что целесообразно отложить выполнение этой задачи, если поблизости есть невыполненные задачи с меньшим дедлайном, чем у задачи-указателя. Таким образом, в работе предлагается использование термина «транзитная» задача для любых задач, которые могут быть выполнены «по пути» к выполнению задачи-указателя без получения штрафа агентом. Выбор транзитной задачи может быть представлен следующим выражением:

$$\min \left((c_i + c_j^{ptr}) \leq t_j^{ptr} \right), l = \overline{1, m}, \quad (5)$$

где $(c_i + c_j^{ptr})$ – время завершения выполнения задачи-указателя после выполнения транзитной задачи.

Аналогично процедуре распределения горящих задач, после выбора транзитной задачи агент оповещает остальных агентов о своем выборе и приступает к выполнению задачи.

Возможна ситуация, когда после выполнения транзитной задачи агентом r_i задача-указатель уже будет закреплена за другим агентом. Тогда выбор следующей задачи осуществляется непосредственно по основным узлам связного списка.

Преимуществом предлагаемого решения является простота реализации и низкая вычислительная сложность. Также необходимо отметить, что численность агентов РРТС может быть уменьшена и это не потребует перепланирования последовательности выполнения задач (увеличится только продолжительность выполнения текущего списка задач). Аналогично при увеличении количества

агентов любой новый агент выбирает свободную задачу согласно связного списка L . Блок-схема этапа распределения задач представлена на рисунке 6а, а блок-схема процедуры поиска транзитной задачи показана на рисунке 6б.

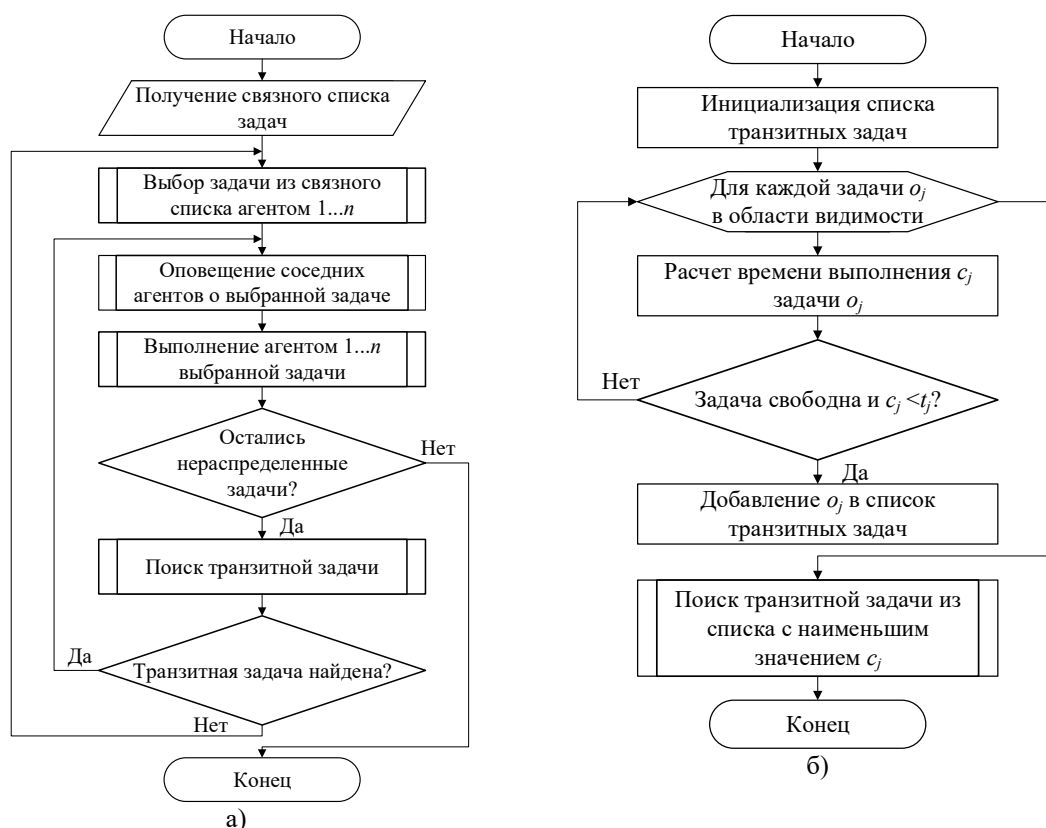


Рисунок 6 – Блок-схемы: а) этапа распределения задач в РРТС; б) поиска транзитных задач

В данном решении достижение консенсуса между агентами наиболее значимо на самой первой итерации этапа распределения задач, когда каждый агент РРТС выбирает горящую задачу. В том случае, если несколько агентов n_c выбирают одну и ту же задачу o_j (такую ситуацию здесь и далее будем называть «коллизией»), алгоритм достижения консенсуса будет следующим. Каждый из n_c агентов рассчитывает расстояние d_i от текущей позиции p_i до позиции начала выполнения задачи p_j . Функция вычисления расстояния будет аналогична представленной в выражении (4), т.е. $d_i = D(p_j, p_i)$, $i = \overline{1, n_c}$. Далее путем информационного обмена агенты, попавшие в коллизию, обмениваются полученными значениями d_i и сравнивают их между собой. В результате этого задачу o_j закрепляет за собой тот агент, у которого значение d_i оказалось наименьшим. В том случае, если у агентов значения d_i оказались равны, то описанная последовательность действий повторяется только между этими агентами с той разницей, что значение d_i не рассчитывается, а выбирается случайно из некоторого диапазона чисел $[0, N]$. Как и ранее, задачу o_j закрепляет за собой агент с минимальным значением d_i . А все остальные агенты продолжают поиск подходящей горящей задачи по связному списку.

Все последующие итерации для агентов РРТС будут асинхронными. В случае повторной коллизии агентов на k -й итерации работы алгоритма, выбор того или иного агента также будет осуществляться на основе метрики расстояния до задачи или случайно, если показатели равны.

Блок-схема процедуры достижения консенсуса между агентами РРТС представлена на рисунке 7.

Необходимо отметить, что при появлении каждой новой задачи связный список L формируется заново на центральном вычислительном устройстве и передается по каналу связи агентам РРТС. При этом каждая задача $o_j \in O$ остается закрепленной за тем агентом, который ее выбрал ранее или уже выполняет. Таким образом, функционирование агентов РРТС при динамическом изменении множества задач O продолжается в штатном режиме.

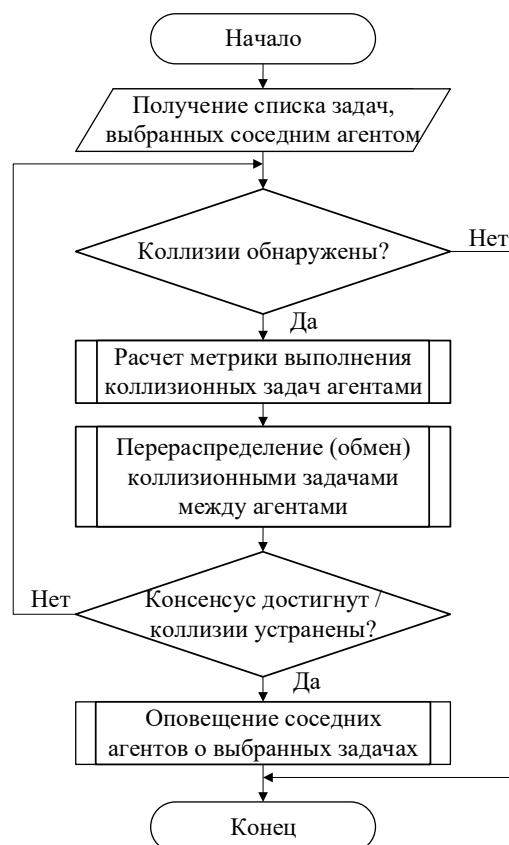


Рисунок 7 – Блок-схема процедуры достижения консенсуса между агентами РРТС при планировании выполнения задач

Таким образом, отличием второго этапа предложенного метода от метода-аналога является то, что:

- агенты РРТС выполняют итеративное распределение задач децентрализованно, при этом каждая следующая задача i -го агента априори известна (без учета транзитных задач);
- в предложенном методе реализуется концепция выполнения транзитных задач, если дедайн следующей задачи связанного списка позволяет выполнить транзитную задачу без получения штрафа;
- в отличие от метода-аналога и других известных методов планирование выполнения задач ограничено только двумя задачами (текущей задачей и задачей-указателем), что позволяет учитывать специфику РРТС, а также позволяет выбирать агентам наиболее оптимальные планы выполнения задач в текущий момент времени с учетом собственного состояния агента;
- в результате масштабирования численности агентов РРТС нет необходимости осуществлять перераспределение задач между агентами – каждый новый агент гибко интегрируется в процесс функционирования РРТС.

3.2. Нейросетевой метод планирования последовательности выполнения задач агентами РРТС. К недостаткам аналитического метода, предложенного в пункте 3.1, можно отнести его декомпозицию на два отдельных этапа: предобработка множества задач на стороне центрального вычислительного устройства и распределение задач между агентами с выбором транзитных задач. На каждом из этапов предложенного метода выполняется полный перебор задач в глубину. Количество вычислений можно уменьшить, если использовать предложенный метод для обучения искусственной нейронной сети (ИНС), а обученную ИНС – для распределения и планирования выполнения задач с использованием вычислительных платформ агентов РРТС. Такой вариант реализации предложенного решения будет полностью децентрализованным, что может быть полезным, если множество задач изменяется непрерывно. С одной стороны, такой подход уменьшает нагрузку на центральное вычислительное устройство, а с другой стороны, использование ИНС для распределения задач позволит учесть те скрытые зависимости при выборе задач, которые аналитический метод не позволяет увидеть явно. Основной идеей реализации нейросетевого метода распределения и планирования последовательности выполнения задач агентами РРТС является использование метода Word2Vec [32] для выбора следующих задач (задачи), которые будут выполнены,

на основе меры их сходства. В отличие от предложенного в пункте 3.1 аналитического метода, этот подход позволит при выборе следующей задачи агентом РРТС найти компромисс между срочностью выполнения задач и расстоянием, которое необходимо пройти в процессе их выполнения, что в конечном итоге повысит эффективность функционирования РРТС.

Метод Word2Vec может быть реализован на основе двух алгоритмов обучения: «непрерывный мешок со словами» (CBOW, continuous bag-of-words model) и «скип-граммы» (skip-gram). Первый алгоритм предназначен для подбора наиболее вероятной задачи на основе представленного контекста (последовательности задач), а второй – для предсказания контекста по исходной задаче. Более наглядно различия в схемах реализации алгоритмов CBOW и skip-gram показаны на рисунке 8, где o обозначает задачу, а значение в скобках – порядок следования этой задачи в последовательности задач.

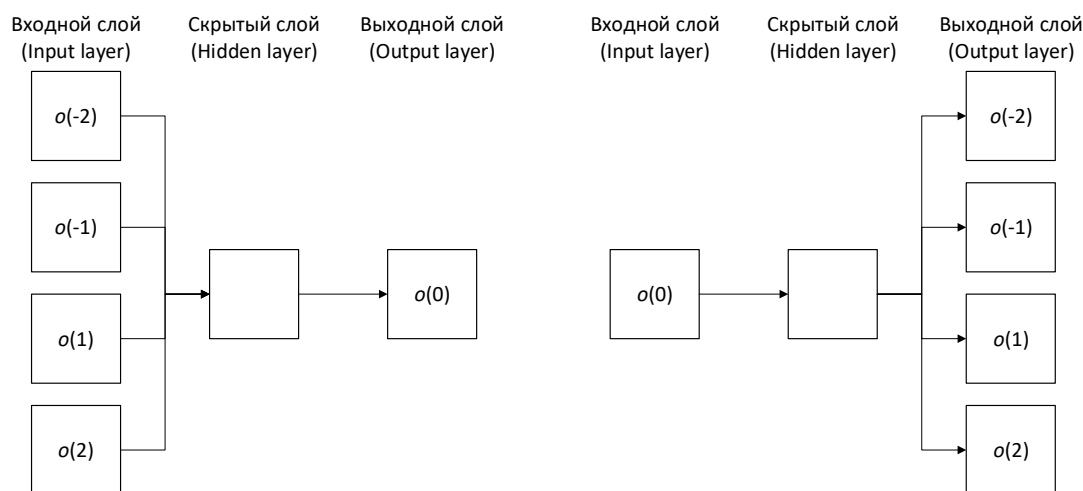


Рисунок 8 – Обобщенные схемы реализации алгоритмов CBOW и skip-gram

В процессе обучения ИНС получает на вход «one-hot» вектор $h = (h_1, h_2, \dots, h_m)$, где $h_j = 1$, если элемент h_j соответствует позиции задачи o_j , и $h_j = 0$ в противном случае, $j = \overline{1, m}$. Задача обучения ИНС заключается в том, чтобы осуществить распределение вероятностей каждой задачи оказаться в контексте входной задачи. В соответствии с оригинальной работой [32] основы процесса обучения ИНС могут быть описаны следующим образом. Вектор h исходной задачи подается на вход ИНС, которая пытается предсказать выходную задачу. После сравнения предсказанной задачи и той задачи, которая на самом деле находится в контекстном окне, вычисляется функция потерь, которая вместе со стохастическим градиентным спуском используется для оптимизации ИНС.

Для рассматриваемой задачи распределения и планирования задач наиболее подходящим является алгоритм skip-gram, так как на вход ИНС может быть подана текущая позиция p_i агента π_i , а на выходе получено распределение вероятностей выбора следующей задачи согласно их позициям. При этом количество выбираемых задач может быть не ограничено. Однако такая реализация соответствует только второму этапу аналитического метода, предложенного в пункте 3.1, так как фактически выбор следующих задач основывается на метрике расстояния между задачами без учета их дедлайна.

Для того чтобы учесть дедлайн выполнения задач, предлагается подавать на вход ИНС не «one-hot» вектор, а вектор h' , содержащий индекс текущей позиции агента $p_i \in P$ и множество значений потенциального штрафа за нарушение дедлайна при выполнении задачи $z = (z_1, z_2, \dots, z_m)$. Расчет значения потенциального штрафа за нарушение дедлайна при выполнении задачи o_j основывается на формуле (2), при этом отличие будет заключаться в том, что время завершения задачи c_j рассчитывается «грубо» в зависимости от расстояния от позиции завершения выполнения текущей задачи b_c (или позиции агента) до позиции начала выполнения следующей задачи a_n , а также расстояния от позиции начала выполнения текущей задачи a_n до позиции завершения выполнения задачи b_n . Под грубым расчетом в данном случае понимается допущение о равномерном движении агента с постоянной скоростью, для более точного расчета можно воспользоваться методами планирования пути перемещения агента или траектории его движения [33–35]. Таким образом, расчет потенциального штрафа z_j за нарушение дедлайна при выполнении задачи o_j имеет следующий вид:

$$z_j = \left(\frac{D(b_c, a_n) + D(a_n, b_n)}{v} - t_j \right) p_j, \quad (6)$$

где v – скорость перемещения агентов (представляет собой константу, и это значение одинаково для всех агентов РРТС). В том случае если задача o_j уже выполнена или выполняется в текущий момент времени другим агентом, то значение потенциального штрафа z_j будет равно 0.

Таким образом, на основе предложенного подхода на выходе ИНС будет получен вектор y , содержащий вероятности выбора задачи агентом согласно метрики, включающей не только расстояние между задачами, но и срочность их выполнения. Чем выше значение элемента вектора y , тем выше приоритет выполнения той задачи из множества O , индекс которой соответствует индексу элемента y . То есть агенту РРТС необходимо найти индекс элемента с наибольшим значением и оповестить других агентов о выборе этой задачи.

В качестве ИНС предлагается использовать многослойный персептрон с двумя скрытыми слоями. Входной и выходной слои будут состоять из $m + 1$ и m нейронов соответственно. Количество нейронов в скрытых слоях выбирается произвольно (с учетом возможностей вычислительных платформ робототехнических устройств, которые используются в качестве агентов РРТС). Однако, по мнению авторов данной работы, размерности скрытых слоев должны быть не меньше размерности входного слоя, т.е. $m + 1$. Схематически структура предлагаемой ИНС представлена на рисунке 9.

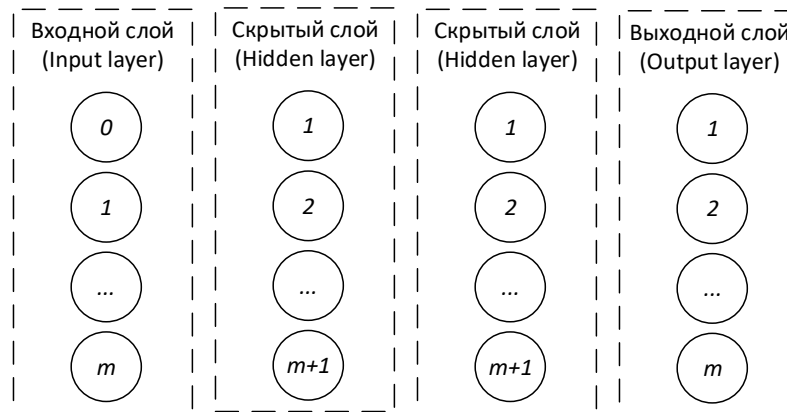


Рисунок 9 – Структура ИНС

Обучение ИНС предлагается выполнить на основе данных, полученных с использованием аналитического метода. Особенность обучающей выборки для ИНС будет заключаться в том, что необходимо использовать не решения отдельных агентов в рамках выполнения множества задач (каждый агент берет на себя только часть задач), а результаты построения плана выполнения всего множества задач каждым из агентов РРТС (каждый агент берет на себя все множество задач). Пусть $A = (A_1, A_2, \dots, A_n)$ – множество решений, полученных при использовании аналитического метода от n агентов. Каждый элемент $A_i, i = \overline{1, n}$ представляет собой множество, состоящее из множества результатов выбора агентом r_i задачи для выполнения, т.е. $A_i = (a_1, a_2, \dots, a_m)$. В свою очередь каждый результат выбора задачи агентом r_i будет содержать позицию агента p_i на момент выбора задачи, значение потенциального штрафа z_j за нарушение дедлайна при выполнении каждой из задач $o_j, j = \overline{1, m}$, рассчитанное с помощью формулы (6), и индекс задачи, которая была выбрана для выполнения $u \in [1, m]$, то есть $a_j = (p_i, z_1, z_2, \dots, z_m, u), j = \overline{1, m}$. Каждый из результатов выбора задачи будет непосредственно использован для обучения ИНС согласно ранее описанному подходу.

Аналогично оригинальной реализации алгоритма skip-gram [32] функция потерь при обучении ИНС определяется следующим образом:

$$\sum_{A_i \in A} \sum_{1 \leq j \leq m} \sum_{-c \leq k \leq c} \log P(o_{j+k} | o_j), \quad (7)$$

где c – размер контекстного окна (в данной работе $c = 1$), а встречаемость задач o_{j+k} и o_j рассматривается независимо от отдельно взятого плана выполнения задач a_j . Вероятность встречаемости задач $P(o_{j+k} | o_j)$ определяется с помощью функции перекрестной энтропии:

$$\log P(o_{j+k} | o_j) = \frac{\exp(v_{o_j}^T v_{o_{j+k}})}{\sum_{p=1}^M \exp(v_{o_j}^T v_p)}, \quad (8)$$

где v_o и v'_o – входное и выходное векторные представления задач; M – количество возможных позиций задач.

Из уравнений (7) и (8) видно, что предложенная модель ИНС моделирует последовательность задач, где задачи с похожим контекстом будут иметь сходные векторные представления.

Таким образом, можно выделить следующие особенности предложенного метода:

- для выбора следующей задачи необходим единовременный запуск алгоритма (например, триггером может быть завершение выполнения текущей задачи), что уменьшает нагрузку на вычислительное устройство агента (вместо перебора всех задач в каждый дискретный момент срабатывания вычислительного устройства агента при поиске транзитной задачи);
- аналитический метод позволяет явно выбрать в качестве следующей задачи ближайшую, однако ИНС позволяет учесть и скрытые зависимости при расчете подобия векторного представления задач, которые не могут быть учтены при аналитическом варианте метода;
- итеративный запуск метода позволит запланировать выполнение любого необходимого количества задач. Соответственно, такой подход может быть использован как при децентрализованном управлении, так и при централизованном, что обуславливает универсальность предложенного метода.

4. Эксперимент. Для проведения эксперимента была выполнена программная реализация метода-аналога и предложенных решений на языке программирования Python. Визуализация взаимодействий агентов РРТС, а также формирование графиков для оценки эффективности рассматриваемых методов выполнены с помощью библиотеки Matplotlib. При проведении моделирования был использован компьютер со следующими характеристиками: процессор Intel Core i7-8550U с тактовой частотой 1,8 ГГц, 8 ГБ оперативной памяти. Использованы параметры моделирования, указанные в таблице 1. Среда моделирования аналогична представленной на рисунке 3.

На рисунках 10–12 представлены показатели времени выполнения задач, количества просроченных задач и суммарное значение штрафа для всех агентов РРТС, полученные в результате моделирования. Полупрозрачные линии определяют результаты каждого из 100 экспериментов, а непрозрачные линии – среднее полученное значение по всем проведенным экспериментам. Аббревиатуры МА, РАМ и РНМ, представленные на рисунках 10–12, обозначают метод-аналог, разработанный аналитический метод и разработанный нейросетевой метод соответственно. Эти обозначения будут использованы и далее.

Таблица 1 – Параметры моделирования

Наименование параметра	Значение
Количество агентов РРТС, n , ед.	50
Количество задач, m , ед.	100
Количество экспериментов, ед.	100
Скорость перемещения агентов, м/с	0,5
Размер карты, м×м	10×10

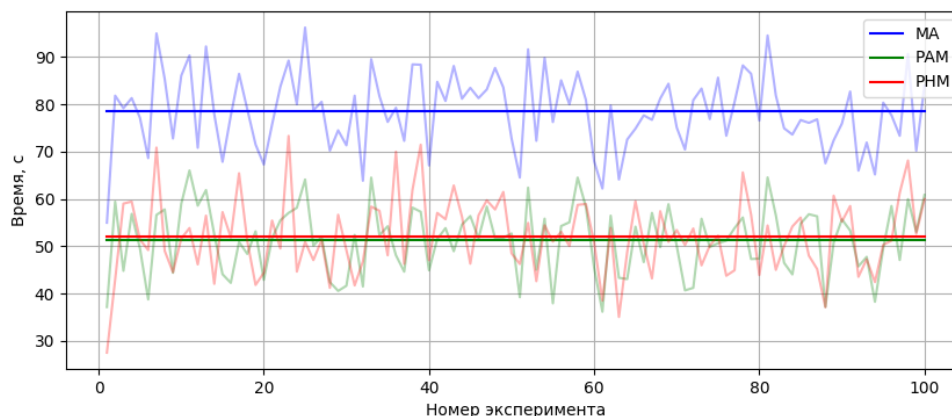


Рисунок 10 – Показатели времени, необходимого агентам РРТС для выполнения задач

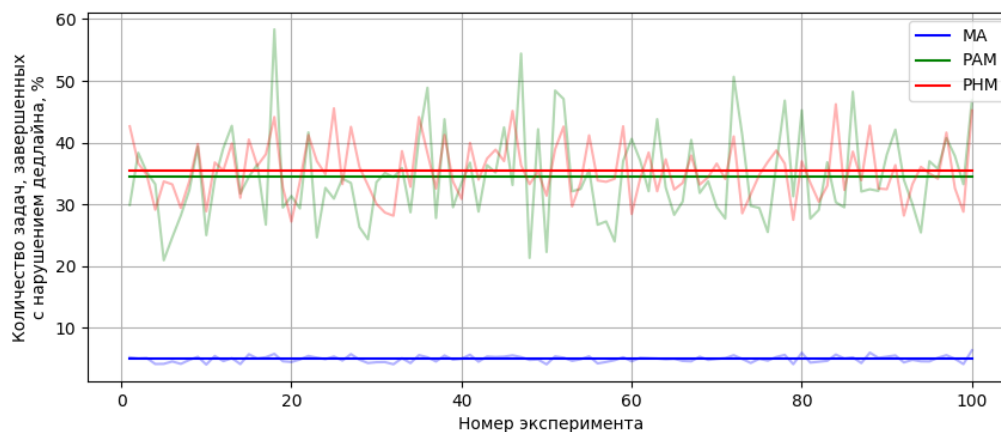


Рисунок 11 – Показатели количества задач, завершенных агентами PPTC с нарушением дедлайна

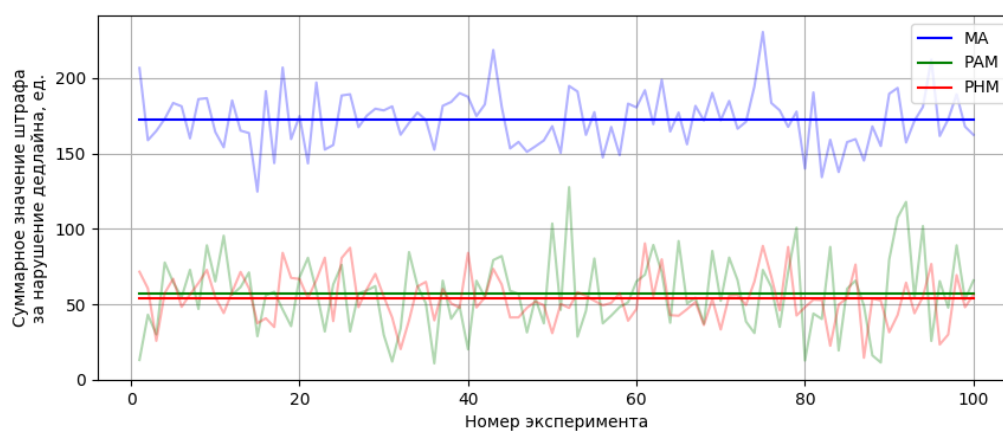


Рисунок 12 – Показатели суммарного штрафа, полученного всеми агентами PPTC в результате нарушения дедлайна при выполнении задач

На рисунках 13–15 показаны результаты обработки данных, представленных на рисунках 10–12, с использованием диаграмм размаха для более наглядного представления полученных результатов.

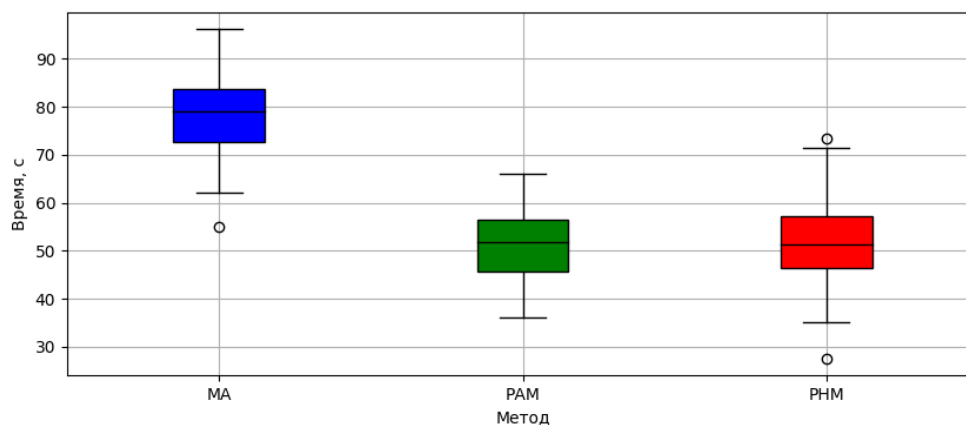


Рисунок 13 – Показатели времени, необходимого агентам PPTC для выполнения задач

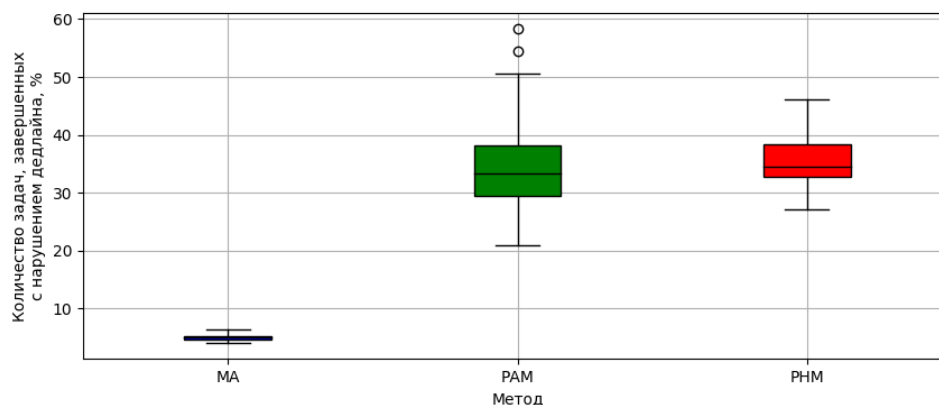


Рисунок 14 – Показатели количества задач, завершенных агентами РРТС с нарушением дедлайна

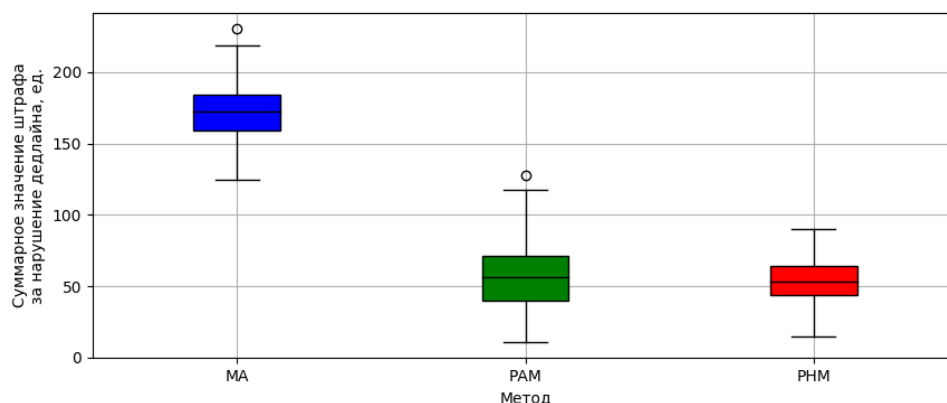


Рисунок 15 – Показатели суммарного штрафа, полученного всеми агентами РРТС в результате нарушения дедлайна при выполнении задач

Для оценки эффективности функционирования РРТС интерес также представляют максимальные, средние и минимальные значения рассматриваемых показателей эффективности выполнения задач, представленные в таблице 2.

Основной идеей МА является недопущение возникновения новых просроченных задач, поэтому показатель количества таких задач составил в среднем 4,98 % (против 34,5 % у РАМ) от общего количества задач согласно результатам, представленным в таблице 2. Однако значение суммарного штрафа для РАМ меньше на 66,8 % (57,33 ед. против 172,61 ед.). Это объясняется тем, что выполнение уже просроченных задач в МА откладывается на более поздний срок, что не всегда допустимо в реальных сценариях автоматизированного склада. Например, некоторый товар может иметь лимитированный срок годности, а с учетом времени, необходимого для доставки товара конечному потребителю, этот товар придет в негодность, либо же товар должен быть утилизирован, что принесет убыток автоматизированному складу. В то же время в РАМ основной идеей является определение очередности горящих задач и очередности их выполнения. Также необходимо отметить, что реализованная концепция транзитных задач в РАМ позволила уменьшить среднее время выполнения задач агентами РРТС на 34,7 % (51,32 с против 78,55 с).

Таблица 2 – Показатели качества выполнения задач агентами РРТС

Критерий	МА	РАМ	PHM
Время, с			
Максимальное значение	96,2	66,01	73,3
Среднее значение	78,55	51,32	52,06
Минимальное значение	54,97	36,16	27,52

Продолжение таблицы 2

Количество задач, завершенных с нарушением дедлайна, %			
Максимальное значение	6,39	58,34	46,22
Среднее значение	4,98	34,5	35,52
Минимальное значение	4,08	20,96	27,22
Суммарное значение штрафа за нарушение дедлайна, ед.			
Максимальное значение	230,52	127,7	90,45
Среднее значение	172,61	57,33	54,26
Минимальное значение	124,77	10,96	14,68

Результаты моделирования РНМ и РАМ сопоставимы, однако необходимо отметить, что РНМ является более универсальным решением, которое может быть использовано не только для РРТС, но и для любых других типов робототехнических систем. При моделировании РНМ наблюдается незначительное увеличение средних значений времени выполнения задач и процента задач, завершенных с нарушением дедлайна. Однако суммарное значение штрафа за нарушение дедлайна меньше, чем у РАМ (54,26 ед. против 57,33 ед.), что также подтверждают максимальные значения этого показателя (90,45 ед. против 127,7 ед.). Исходя из этого, можно заключить, что использование РНМ позволяет уменьшить значение штрафа при выполнении каждой отдельно взятой задачи при сопоставимых значениях остальных показателей.

Таким образом, результаты проведенного моделирования свидетельствуют о повышении эффективности функционирования РРТС за счет разработанных решений.

Заключение. Для повышения эффективности распределения и планирования задач в РРТС в условиях недетерминированной среды с учетом ограниченных возможностей агентов РРТС и специфики децентрализованного управления в данной работе были предложены новые решения в виде аналитического и нейросетевого методов распределения и планирования выполнения задач агентами РРТС в сценарии автоматизированного склада. Наиболее близким аналогом к представленным решениям является работа [29], основная идея которой заключается в том, чтобы минимизировать количество задач, выполненных с нарушением дедлайна. Отличие методов, представленных в данной работе, заключается в том, что уже «просроченные» задачи выполняются в первую очередь, причем если появляется возможность выполнить дополнительную задачу без нарушения дедлайна задачи, следующей согласно очередности, то агенты выполняют такие задачи. Элементом научной новизны являются предложенные алгоритмы сортировки задач и поиска транзитных задач, позволяющие не только уменьшить время нарушения дедлайна при выполнении задач, но и уменьшить общее время выполнения всего множества задач. Также стоит отметить, что множество задач может быть изменено (например, дополнено новыми задачами) в любой момент функционирования РРТС. После завершения текущей задачи каждый агент продолжает перебирать связный список и выбирать наиболее подходящую задачу. Элементом научной новизны нейросетевого метода распределения и планирования последовательности выполнения задач агентами РРТС является его реализация на основе векторного представления задач и определения меры их подобия аналогично методу Word2Vec. Полученное решение является полностью децентрализованным, что делает разработанный метод универсальным и позволяет использовать его не только для РРТС, но и для любых других типов робототехнических систем. Достоверность полученных результатов подтверждена в ходе программного моделирования. Однако для анализа применимости разработанного метода на практике требуется проведение натурных экспериментов в среде, близкой к условиям функционирования РРТС в автоматизированном складе. Эта задача требует дальнейшей проработки, на что и будут направлены дальнейшие исследования.

Список литературы

1. Каляев, И. А. Модели и алгоритмы коллективного управления в группах роботов / И. А. Каляев, А. Р. Гайдук, С. Г. Капустян. – Москва : ФИЗМАТЛИТ, 2009. – 280 с.
2. Alibaba Group. – Режим доступа: <https://www.alibabagroup.com/en/global/home>, свободный. – Заглавие с экрана. – Яз. рус. (дата обращения: 18.05.2022).
3. Seenu, N. Review on state-of-the-art dynamic task allocation strategies for multiple-robot systems / N. Seenu, R. M. Kuppan Chetty, M. M. Ramya, N. J. Mukund // Industrial Robot. – 2020. – Vol. 47, № 6. – P. 929–942.

4. Dai, W. Multi-robot dynamic task allocation for exploration and destruction / W. Dai, H. Lu, J. Xiao, Z. Zeng, Z. Zheng // *Journal of Intelligent and Robotic Systems: Theory and Applications*. – 2020. – Vol. 98, № 2. – P. 455–479.
5. Khamis, A. Multi-robot Task Allocation: A Review of the State-of-the-Art / A. Khamis, A. Hussein, A. Elmogy // *Cooperative Robots and Sensor Networks*. – 2015. – P. 31–51.
6. Петренко, В. И. Анализ рисков нарушения информационной безопасности в роевых робототехнических системах при масштабировании численности агентов / В. И. Петренко, Ф. Б. Тебуева, А. С. Павлов, И. В. Стручков // *Прикаспийский журнал: управление и высокие технологии*. – 2022. – № 2. – С. 92–109.
7. Kowalczyk, W. Target Assignment Strategy for Scattered Robots Building Formation / W. Kowalczyk // *Proceedings of the Third International Workshop on Robot Motion and Control (RoMoCo '02)*. – 2002. – P. 181–185.
8. Ni, M. A Lagrange relaxation method for solving weapon-target assignment problem / M. Ni, Z. Yu, F. Ma, X. Wu // *Mathematical Problems in Engineering*. – 2011. – Vol. 1. – P. 1–10.
9. Zavlanos, M. A distributed auction algorithm for the assignment problem / M. Zavlanos, L. Spesivtsev, G. Pappas // *47th IEEE Conference on Decision and Control*. – 2008. – P. 1212–1217.
10. Bertsekas, D. Parallel synchronous and asynchronous implementations of the auction algorithm / D. Bertsekas, D. Castanon // *Parallel Computing*. – 1991. – Vol. 17. – P. 707–732.
11. Zavlanos, M. Dynamic assignment in distributed motion planning with local coordination / M. Zavlanos, G. Pappas // *IEEE Transactions on Robotics*. – 2008. – Vol. 24, № 1. – P. 232–242.
12. Zavlanos, M. Sensor-based dynamic assignment in distributed motion planning / M. Zavlanos, G. Pappas // *Proceedings 2007 IEEE International Conference on Robotics and Automation*. – 2007. – P. 3333–3338.
13. Azad, M. M. Efficient Heuristic Approaches to the Weapon Target Assignment Problem / M. M. Azad, A. Mircea // *Journal of Aerospace Computing, Information and Communication*. – 2009. – Vol. 6. – P. 405–415.
14. Berman, S. Optimized Stochastic Policies for Task Allocation in Swarms of Robots / S. Berman, A. Halasz, M. A. Hsieh, V. Kumar // *IEEE Transactions on Robotics*. – 2009. – Vol. 25, № 4. – P. 927–937.
15. Mouton, H. Applying Reinforcement Learning to the Weapon Assignment Problem in Air Defense / H. Mouton, J. Roodt, H. Roux // *Scientia Militaria South African Journal of Military Studies*. – 2011. – Vol. 39, № 2. – P. 1–15.
16. Zhao, H. General Dynamic Neural Networks for the Adaptive Tuning of an Omni-Directional Drive System for Reactive Swarm Robotics / H. Zhao, M. Dorigo, M. Allwright // *25th International Conference on Methods and Models in Automation and Robotics (MMAR)*. – 2021. – P. 79–84.
17. Mukhedkar, R. Weapon Target Allocation Problem Using Fuzzy Model / R. Mukhedkar, S. Naik // *International Journal of Application or Innovation in Engineering & Management (IJAIEEM)*. – 2013. – Vol. 2, № 6. – P. 279–289.
18. Oliveira, S. Analysis of the population-based ant colony optimization algorithm for the TSP and the QAP / S. Oliveira, M. S. Hussin, A. Roli, M. Dorigo, T. Stützle // *IEEE Congress on Evolutionary Computation (CEC)*. – 2017. – P. 1734–1741.
19. Liao, T. Ant Colony Optimization for Mixed-Variable Optimization Problems / T. Liao, K. Socha, M. A. Montes de Oca, T. Stützle, M. Dorigo // *IEEE Transactions on Evolutionary Computation*. – 2014. – Vol. 18, № 4. – P. 503–518.
20. Murphy, R. Target-Based Weapon Target Assignment Problems / R. Murphy // *Nonlinear Assignment Problems. Combinatorial Optimization*. – 1999. – Vol. 7. – P. 39–53.
21. Sikanen, T. Solving Weapon Target Assignment Problem with Dynamic Programming / T. Sikanen // *Independent Research Projects in Applied Mathematics*. – 2008. – P. 32.
22. Shima, T. Multiple task assignments for cooperating uninhabited aerial vehicles using genetic algorithms / T. Shima, S. Rasmussen, A. Sparks, K. Passino // *Computers & Operations Research*. – 2006. – Vol. 33. – P. 3252–3269.
23. Zhang, J. ACGA Algorithm of Solving Weapon Target Assignment Problem / J. Zhang, X. Wang, C. Xu // *Open Journal of Applied Sciences*. – 2012. – Vol. 2. – P. 74–77.
24. Sarkar, C. A Scalable Multi-Robot Task Allocation Algorithm / C. Sarkar, H. S. Paul, A. Pal // *IEEE International Conference on Robotics and Automation (ICRA)*. – 2018. – P. 5022–5027.
25. Lu, Q. Multiple-place swarm foraging with dynamic depots / Q. Lu, J. Hecker, M. Moses // *Autonomous Robots*. – 2018. – Vol. 42, № 4. – P. 909–926.
26. Zhou, L. A Balanced Heuristic Mechanism for Multirobot Task Allocation of Intelligent Warehouses / L. Zhou, Y. Shi, J. Wang, P. Yang // *Mathematical Problems in Engineering*. – 2014. – Vol. 2014. – P. 1–10.
27. Tkach, I. A Modified Distributed Bees Algorithm for Multi-Sensor Task Allocation / I. Tkach, A. Jevtić, S. Y. Nof, Y. Edan // *Sensors*. – 2018. – Vol. 18. – P. 759.
28. Karaboga, D. An Idea Based on Honey Bee Swarm for Numerical Optimization / D. Karaboga // *Technical Report-TR06, Department of Computer Engineering, Engineering Faculty, Erciyes University*. – 2005. – P. 1–10.
29. Sarkar, C. Cannot avoid penalty? Let's minimize / C. Sarkar, M. Agarwal // *International Conference on Robotics and Automation (ICRA)*. – 2019. – P. 1052–1058. – doi: 10.1109/ICRA.2019.8794338.
30. Zakiev, A. Swarm Robotics: Remarks on Terminology and Classification / A. Zakiev, T. Tsoy, E. Magid // *Interactive Collaborative Robotics (ICR 2018)*. – 2018. – P. 291–300.
31. Koubaa, A. Performance Analysis of the MRTA Approaches for Autonomous Mobile Robot / A. Koubaa, H. Bennaceur, I. Chaari, S. Triguí, A. Ammar, M.-F. Sriti, M. Alajlan, O. Cheikhrouhou, Y. Javed // *Robot Path Planning and Cooperation. Studies in Computational Intelligence*. – 2018. – Vol. 772. – P. 169–188.

32. Mikolov, T. Distributed Representations of Words and Phrases and their Compositionality / T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean // *Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS'13)*. – 2013. – Vol. 2. – P. 3111–3119.

33. Павлов, А. С. Методика планирования траектории движения группы мобильных роботов в неизвестной замкнутой среде с препятствиями / А. С. Павлов // *Системы управления, связи и безопасности*. – 2021. – № 3. – С. 38–59.

34. Петренко, В. И. Метод планирования траектории движения точки в пространстве с препятствием на основе итеративной кусочно-линейной аппроксимации / В. И. Петренко, Ф. Б. Тебueva, В. О. Антонов, М. М. Гурчинский // *Системы управления, связи и безопасности*. – 2018. – № 1. – С. 168–182.

35. Юдинцев, Б. С. Синтез нейросетевой системы планирования траекторий для группы мобильных роботов / Б. С. Юдинцев // *Системы управления, связи и безопасности*. – 2019. – № 4. – С. 163–186.

References

1. Kalyaev, I. A., Gayduk, A. R., Kapustyan, S. G. *Modeli i algoritmy kollektivnogo upravleniya v gruppakh robotov* [Models and Algorithms for Collective Control in Groups of Robots]. Moscow, Fizmatlit Publ., 2009.
2. Alibaba Group. Available at: <https://www.alibabagroup.com/en/global/home> (accessed 18.05.2022).
3. Seenu, N., Kuppam Chetty, R. M., Ramya, M. M., Mukund, N. J. Review on state-of-the-art dynamic task allocation strategies for multiple-robot systems. *Industrial Robot*, 2020, vol. 47, no. 6, pp. 929–942.
4. Dai, W., Lu, H., Xiao, J., Zeng, Z., Zheng, Z. Multi-robot dynamic task allocation for exploration and destruction. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 2020, vol. 98, no. 2, pp. 455–479.
5. Khamis, A., Hussein, A., Elmogy, A. Multi-robot Task Allocation: A Review of the State-of-the-Art. *Cooperative Robots and Sensor Networks*, 2015, pp. 31–51.
6. Petrenko, V. I., Tebueva, F. B., Pavlov, A. S., Struchkov, I. V. Analiz riskov narusheniya informatsionnoy bezopasnosti v roevykh robototekhnicheskikh sistemakh pri masshtabirovanii chislennosti agentov [Analysis of information security breach risks in swarm robotic systems when scaling the number of agents]. *Prikaspiyskiy zhurnal: upravlenie i vysokie tekhnologii* [Caspian Journal: Control and High Technologies], 2022, no. 2, pp. 92–109.
7. Kowalczyk, W. Target Assignment Strategy for Scattered Robots Building Formation. *Proceedings of the Third International Workshop on Robot Motion and Control (RoMoCo '02)*, 2002, pp. 181–185.
8. Ni, M., Yu, Z., Ma, F., Wu, X. A Lagrange relaxation method for solving weapon-target assignment problem. *Mathematical Problems in Engineering*, 2011, vol. 1, pp. 1–10.
9. Zavlanos, M., Spesivtsev, L., Pappas, G. A distributed auction algorithm for the assignment problem. *47th IEEE Conference on Decision and Control*, 2008, pp. 1212–1217.
10. Bertsekas, D., Castanon, D. Parallel synchronous and asynchronous implementations of the auction algorithm. *Parallel Computing*, 1991, vol. 17, pp. 707–732.
11. Zavlanos, M., Pappas, G. Dynamic assignment in distributed motion planning with local coordination. *IEEE Transactions on Robotics*, 2008, vol. 24, no. 1, pp. 232–242.
12. Zavlanos, M., Pappas, G. Sensor-based dynamic assignment in distributed motion planning. *Proceedings 2007 IEEE International Conference on Robotics and Automation*, 2007, pp. 3333–3338.
13. Azad, M. M., Mircea, A. Efficient Heuristic Approaches to the Weapon Target Assignment Problem. *Journal of Aerospace Computing, Information and Communication*, 2009, vol. 6, pp. 405–415.
14. Berman, S., Halasz, A., Hsieh, M. A., Kumar, V. Optimized Stochastic Policies for Task Allocation in Swarms of Robots. *IEEE Transactions on Robotics*, 2009, vol. 25, no. 4, pp. 927–937.
15. Mouton, H., Roodt, J., Roux, H. Applying Reinforcement Learning to the Weapon Assignment Problem in Air Defense. *Scientia Militaria South African Journal of Military Studies*, 2011, vol. 39, № 2, pp. 1–15.
16. Zhao, H., Dorigo, M., Allwright, M. General Dynamic Neural Networks for the Adaptive Tuning of an Omni-Directional Drive System for Reactive Swarm Robotics. *25th International Conference on Methods and Models in Automation and Robotics (MMAR)*, 2021, pp. 79–84.
17. Mukhedkar, R., Naik, S. Weapon Target Allocation Problem Using Fuzzy Model. *International Journal of Application or Innovation in Engineering & Management (IIAEM)*, 2013, vol. 2, no. 6, pp. 279–289.
18. Oliveira, S., Hussin, M. S., Roli, A., Dorigo, M., Stützle, T. Analysis of the population-based ant colony optimization algorithm for the TSP and the QAP. *IEEE Congress on Evolutionary Computation (CEC)*, 2017, pp. 1734–1741.
19. Liao, T., Socha, K., Montes de Oca, M. A., Stützle, T., Dorigo, M. Ant Colony Optimization for Mixed-Variable Optimization Problems. *IEEE Transactions on Evolutionary Computation*, 2014, vol. 18, no. 4, pp. 503–518.
20. Murphy, R. Target-Based Weapon Target Assignment Problems. *Nonlinear Assignment Problems. Combinatorial Optimization*, 1999, vol. 7, pp. 39–53.
21. Sikanen, T. Solving Weapon Target Assignment Problem with Dynamic Programming. *Independent Research Projects in Applied Mathematics*, 2008, p. 32.
22. Shima, T., Rasmussen, S., Sparks, A., Passino, K. Multiple task assignments for cooperating uninhabited aerial vehicles using genetic algorithms. *Computers & Operations Research*, 2006, vol. 33, pp. 3252–3269.
23. Zhang, J., Wang, X., Xu, C. ACGA Algorithm of Solving Weapon Target Assignment Problem. *Open Journal of Applied Sciences*, 2012, vol. 2, pp. 74–77.
24. Sarkar, C., Paul, H. S., Pal, A. A Scalable Multi-Robot Task Allocation Algorithm. *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 5022–5027.
25. Lu, Q., Hecker, J., Moses, M. Multiple-place swarm foraging with dynamic depots. *Autonomous Robots*, 2018, vol. 42, no. 4, pp. 909–926.

26. Zhou, L., Shi, Y., Wang, J., Yang, P. A Balanced Heuristic Mechanism for Multirobot Task Allocation of Intelligent Warehouses. *Mathematical Problems in Engineering*, 2014, vol. 2014, pp. 1–10.
27. Tkach, I., Jevtić, A., Nof, S. Y., Edan, Y. A Modified Distributed Bees Algorithm for Multi-Sensor Task Allocation. *Sensors*, 2018, vol. 18, pp. 759.
28. Karaboga, D. An Idea Based on Honey Bee Swarm for Numerical Optimization. *Technical Report-TR06, Department of Computer Engineering, Engineering Faculty, Erciyes University*, 2005, pp. 1–10.
29. Sarkar, C., Agarwal, M. Cannot avoid penalty? Let's minimize. *International Conference on Robotics and Automation (ICRA)*, 2019, pp. 1052–1058. doi: 10.1109/ICRA.2019.8794338.
30. Zakiev, A., Tsoy, T., Magid, E. Swarm Robotics: Remarks on Terminology and Classification. *Interactive Collaborative Robotics (ICR 2018)*, 2018, pp. 291–300.
31. Koubaa, A., Bennaceur, H., Chaari, I., Trigui, S., Ammar, A., Sriti, M.-F., Alajlan, M., Cheikhrouhou, O., Javed, Y. Performance Analysis of the MRTA Approaches for Autonomous Mobile Robot. *Robot Path Planning and Cooperation. Studies in Computational Intelligence*, 2018, vol. 772, pp. 169–188.
32. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J. Distributed Representations of Words and Phrases and their Compositionality. *Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS'13)*, 2013, vol. 2, pp. 3111–3119.
33. Pavlov, A. S. Metodika planirovaniya traektorii dvizheniya gruppy mobilnykh robotov v neizvestnoy zamknoy srede s prepyatstviyami [Methodology for Planning the Trajectory of a Group of Mobile Robots in Unknown Closed Environment with Obstacles]. *Sistemy upravleniya, svyazi i bezopasnosti* [Systems of Control, Communication and Security], 2021, no. 3, pp. 38–59.
34. Petrenko, V. I., Tebueva, F. B., Antonov, V. O., Gurchinskiy, V. V. Metod planirovaniya traektorii dvizheniya tochki v prostranstve s prepyatstviem na osnove iterativnoy kusochno-lineynoy approksimatsii [A method for planning the trajectory of a point in space with an obstacle based on iterative piecewise linear approximation]. *Sistemy upravleniya, svyazi i bezopasnosti* [Systems of Control, Communication and Security], 2018, no. 1, pp. 162–182.
1. Yudinsev, B. S. Sintez neyrosetevoy sistemy planirovaniya traektoriy dlya gruppy mobilnykh robotov [Synthesis of a neural network path planning system for a group of mobile robots]. *Sistemy upravleniya, svyazi i bezopasnosti* [Systems of Control, Communication and Security], 2019, no. 4, pp. 163–186.