Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н. Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ

НА ТЕМУ:

«Метод координации агентов в игровой системе безопасности с использованием потенциальных полей»

Студент	ИУ7-86Б		Дремин К. А.
	(Группа)	(Подпись, дата)	(И. О. Фамилия)
Руководитель ВКР			Москвичев Н. В.
		(Подпись, дата)	(И. О. Фамилия)
Нормоконтролер			Кострицкий А. С.
		(Подпись, дата)	(И. О. Фамилия)

2025 г.

РЕФЕРАТ

Расчетно-пояснительная записка 30 с., 2 табл., 17 источн., 1 прил.

ЗАГУРАЖЕМЫЙ МОДУЛЬ, ПРОМАХИ КЕША, TLB.

Работа посвящена разработке загружаемого модуля ядра Linux для сбора статистики промахов в кеше TLB.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

1 Аналитический раздел

1.1 Постановка задачи координации агентов в многоагентной системе

Координация агентов в многоагентных системах (МАС) представляет собой задачу организации взаимодействия между автономными субъектами (агентами) с целью достижения общей цели или выполнения множества задач [6] [4]. В данной работе задача координации рассматривается в следующих условиях:

— общей целью агентов является визуальный контроль критических областей и минимизация угроз нарушения их безопасности;

— рассматриваемая МАС является частью игрового приложения, вследствие чего необходимо достичь внешне реалистичного поведения агентов;

— метод координации агентов не должен быть слишком трудоемким – теоретическая реализация выбранного алгоритма должна быть пригодна для применения в игровых приложениях.

Рассматриваемая в рамках данной работы многоагентная система относится к следующим категориям, описанным в [1]:

1. **Лидерство:** Система является **безлидерной**, так как агенты действуют независимо, принимая решения на основе локальных данных и целей, согласованных с общей моделью.

2. **Функция принятия решений:** Принятие решений в данной МАС **нелинейное**, так как действия агентов зависят от сложных взаимодействий между препятствиями, критическими областями и угрозами.

3. **Гетерогенность:** Система является **гомогенной**, так как все агенты обладают одинаковым функционалом и характеристиками.

4. **Топология:** Топология системы **динамическая**, так как агенты перемещаются в пространстве и их взаимодействия изменяются в зависимости от положения и состояния среды.

5. **Мобильность:** Агенты в системе **мобильные**, так как перемещаются в пространстве для выполнения своих задач, таких как патрулирование и реагирование на угрозы.

1.2 Описание среды координации

Рассмотрим формальное описание среды, в которой функционируют агенты при решении данной задачи, чтобы в дальнейшем определить наиболее применимые методы.

1. Доступные для перемещения области. Среда моделируется как множество областей $\mathcal{N} \subset \mathbb{R}^2$, доступных для перемещения агентов, описываемых навигационной картой. Навигационная карта представлена графом $G = (V, E)$ [7] [1], где V — множество вершин, соответствующих дискретным точкам в $\mathcal{N}$, а E — множество ребер, определяющих пути перемещения между вершинами, как показано в формуле (1.1):

$$\mathcal{N} = \bigcup_{v \in V} \mathcal{A}_v, \quad \mathcal{A}_v \subset \mathbb{R}^2, \tag{1.1}$$

где $\mathcal{A}_v$ — выпуклый многоугольник, ассоциированный с вершиной v.

2. Препятствия, ограничивающие обзор. Препятствия в среде $\mathcal{E}$ задаются множеством $\mathcal{B} = \{B_1, B_2, \ldots, B_k\}$, где каждый объект B_i определяется ограничивающей областью в пространстве $\mathbb{R}^2$ и определен в соответствии с формулой (1.2):

$$B_i = [x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}], \quad i \in \{1, 2, \ldots, k\}. \tag{1.2}$$

Присутствие объектов $\mathcal{B}$ влияет на обзор агентов, ограничивая видимость в направлении, пересекающем препятствия.

3. Критические области. Критические области, требующие визуального контроля, заданы множеством точек $\mathcal{C} = \{c_1, c_2, \ldots, c_m\} \subset \mathcal{N}$. Каждая точка c_j характеризуется радиусом влияния $r_j > 0$, определяющим зону контроля, как описано в формуле (1.3):

$$\mathcal{Z}(c_j) = \{p \in \mathbb{R}^2 \mid \|p - c_j\| \leq r_j\}, \quad j \in \{1, 2, \ldots, m\}. \tag{1.3}$$

Задача агентов заключается в том, чтобы обеспечить покрытие всех зон $\mathcal{Z}(c_j)$

при учете ограничения видимости, задаваемого препятствиями.

4. Видимость в среде. Модель видимости агента a_i определяется его текущим положением $p_i \in \mathcal{N}$ и углом обзора ϕ_i. Область видимости агента формируется как сектор окружности, определяемый формулой (1.4):

$$\mathcal{V}(p_i, \phi_i, r_{\max}) = \{p \in \mathbb{R}^2 \mid \|p - p_i\| \leq r_{\max},\, \theta(\dot{p}_i(t), p) \leq \phi_i\}, \tag{1.4}$$

где $r_{\max}$ — максимальная дальность обзора, $\theta(\dot{p}_i(t), p)$ — угол между направлением агента и вектором к точке p.

Таким образом, среда представляет собой совокупность доступных областей $\mathcal{N}$, препятствий $\mathcal{B}$ и критических точек $\mathcal{C}$, которые агенты обязаны контролировать с учетом ограничений видимости. Основная задача координации заключается в определении таких траекторий и позиций агентов, которые минимизируют неохваченные зоны $\mathcal{Z}(c_j)$.

5. Агенты. Агенты в системе заданы множеством $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$. Каждый агент a_i характеризуется следующими параметрами:

— текущей позицией $p_i(t) \in \mathcal{N}$ в момент времени t;

— направлением движения $\theta_i(t) \in [0, 2\pi)$;

— скоростью перемещения $v_i(t) \in [0, v_{\max}]$.

Траектория движения агента задается уравнением (1.5):

$$\dot{p}_i(t) = v_i(t) \cdot \begin{bmatrix} \cos(\theta_i(t)) \\ \sin(\theta_i(t)) \end{bmatrix}. \tag{1.5}$$

6. Угрозы. Угрозы представлены множеством $\mathcal{T}_{\text{threat}} = \{\tau_1, \tau_2, \ldots, \tau_k\}$, где каждая угроза τ_j является динамическим объектом, имеющим параметры:

— позицию $q_j(t) \in \mathcal{N}$ в момент времени t;

— направление движения $\phi_j(t) \in [0, 2\pi)$;

— скорость $v_j(t) \in [0, v_{\text{threat}}]$.;

— радиус влияния угрозы r_τ.

Движение угроз также описывается уравнением (1.5).

Достижение угрозой одной из критических точек $c \in \mathcal{C}$ считается нарушением безопасности и должно быть предотвращено агентами.

7. Динамическое патрулирование. Для обеспечения контроля над всей областью $\mathcal{N}$ агенты реализуют стратегию динамического патрулирования. При этом вводится функция *опасности* $U(p,t)$, описывающая степень опасности в точке $p \in \mathcal{N}$ в момент времени t.

Общая динамика изменения $U(p,t)$ описывается уравнением (1.6):

$$\frac{\partial U(p,t)}{\partial t} = \alpha - \beta \sum_{i=1}^{n} \mathbf{1}\{p \in \mathcal{V}(p_i(t), \phi_i(t), r_{\max})\}, \tag{1.6}$$

где: $\alpha > 0$ — скорость естественного роста опасности в непосещаемых зонах; $\beta > 0$ — скорость снижения опасности за счет патрулирования агентами.

В области с радиусом r_τ вокруг угрозы значение функции опасности увеличивается согласно формуле (1.7):

$$U(p,t) = U(p,t) + \gamma \cdot \mathbf{1}\{\|p - q_\tau(t)\| \leq r_\tau\}, \tag{1.7}$$

где: $\gamma > 0$ — интенсивность роста опасности, связанная с угрозой τ; r_τ — радиус влияния угрозы τ.

Таким образом, значение функции опасности увеличивается в моменте, когда угроза находится рядом, и прекращает расти, как только угроза удаляется.

8. Задача агентов. Задача агентов заключается в следующем:

— минимизировать функцию общей опасности (1.8):

$$U_{\text{total}}(t) = \int_{\mathcal{N}} U(p,t)\, dp; \tag{1.8}$$

— обнаруживать угрозы $\tau_j \in \mathcal{T}_{\text{threat}}$ и предотвращать их достижение критических точек $\mathcal{C}$.

Если угроза τ_j обнаружена агентом a_i, то остальные агенты $a_{k \neq i}$ координируют свои действия для перехвата угрозы, чтобы нейтрализовать ее и предотвратить достижение критической области.

2 Обзор методов координации агентов в многоагентных системах

2.1 Классификация методов координации агентов в многоагентных системах

В данной работе классификация методов проводится с учетом особенностей задачи визуального покрытия критических областей, а также требований к моделированию в игровых приложениях [8]. Для описания методов предлагается следующая классификация:

1. Тип взаимодействия между агентами

Методы координации могут быть разделены на централизованные и децентрализованные [1]:

— **Централизованные методы**: предполагают наличие центрального узла, который координирует действия всех агентов. Такие методы обеспечивают глобальную оптимальность решений, но требуют глобальной видимости среды и ее полный анализ;

— **Децентрализованные методы**: агенты принимают решения только на основе полученной ими информации и действуют независимо от других. Эти методы более устойчивы к сбоям и масштабируемы, но могут приводить к субоптимальным решениям.

2. Область восприятия агента

Методы различаются по тому, какую часть среды может учитывать агент при принятии решений [1]:

— **Методы с глобальным восприятием**: агенты имеют доступ к информации обо всей среде, включая местоположение всех критических точек, других агентов и угроз. Это требует высокой вычислительной мощности и связности сети.

— **Методы с локальным восприятием**: решения принимаются на основе информации из ограниченной области вокруг агента. Это снижает требо-

вания к вычислительным ресурсам, но может увеличить риск неполного охвата критических точек.

3. Способ распределения задач между агентами

Эффективность координации зависит от способа распределения задач [9]:

— **Жестко распределенные задачи**: каждому агенту заранее назначается определенная область или роль, что упрощает координацию, но снижает адаптивность к изменяющимся условиям.

— **Динамическое распределение задач**: задачи перераспределяются в процессе выполнения на основе текущей информации о среде. Этот подход обеспечивает большую гибкость, но требует дополнительных вычислений.

4. Сложность алгоритмов реализации

Хотя сложность конкретных алгоритмов реализации методов координации агентов может различаться, для сравнения методов полезной является приблизительная оценка трудоёмкости вычисления методов в зависимости от основных параметров модели:

— a — количество агентов в системе. Это определяет степень взаимодействия между агентами и влияет на необходимость синхронизации и перерасчёта их состояний.

— v — количество вершин графа среды. Вершины представляют возможные положения агентов и критических точек.

— e — количество рёбер графа среды, связанное с числом вершин соотношением $e \sim c \cdot v$, где c — среднее число связей для каждой вершины.

— b — количество препятствий, влияющих на построение областей видимости и маршрутов.

— t — количество угроз – вражеских объектов, которые нужно обнаружить и нейтрализовать.

— c — количество критических областей, требующих визуального контроля.

Сводя эти параметры, можно выделить минимально необходимые переменные для оценки сложности:

— a — количество агентов;

— v — количество вершин графа среды;

— c - количество критических областей;

— b — количество препятствий;

— t — количество угроз.

Оценка трудоемкости производится в нотации «O» большое для сравнения асимптотического поведения функций.

5. Гибкость метода к изменяющимся условиям

Из-за того, что среда является высоко динамичной за счет наличия движущихся агентов и изменяющегося уровня опасности, методы могут быть классифицированы по гибкости к изменяющимся условиям следующим образом:

— **Методы с низкой гибкостью**: эффективно работают в статических средах, но требуют значительного времени для перерасчета при изменении условий.

— **Методы с высокой гибкостью**: автоматически корректируют действия агентов в ответ на изменения в среде, что делает их подходящими для задач в динамических игровых приложениях.

6. Визуальная правдоподобность движений агентов

Для игровых приложений важно, чтобы движения агентов выглядели естественно с точки зрения игрока. Методы могут быть классифицированы по уровню визуальной правдоподобности:

— **Прямолинейные методы**: движения агентов строго следуют оптимальной траектории. Это может быть эффективно с точки зрения минимизации затрат, но выглядит механистично и неестественно.

— **Методы с естественным поведением**: включают элементы непредсказуемости или плавности в траекториях агентов, что улучшает восприятие их действий игроком.

2.2 Методы координации агентов в многоагентных системах

2.2.1 Метод потенциальных полей

Метод потенциальных полей основывается на вычислении градиента искусственного потенциала, который направляет движение агентов [2]. Для применения к нашей задаче метод должен учитывать следующие элементы: препятствия, критические области, угрозы, а также общую функцию опасности.

Общее описание метода потенциалов

Потенциал для агента a_i задается как функция (2.1):

$$V(p_i) = V_{\text{аттрактор}}(p_i) + V_{\text{репеллент}}(p_i), \tag{2.1}$$

где p_i — позиция агента. Компоненты потенциала определяются следующим образом:

— $V_{\text{акттрактор}}(p_i)$ — компонент, притягивающий агента к целевым областям (например, критическим точкам).

— $V_{\text{репеллент}}(p_i)$ — компонент, отталкивающий агента от препятствий, других агентов и областей с высоким уровнем опасности.

Градиент потенциала (2.2) определяет направление движения агента:

$$\dot{p_i} = -\nabla V(p_i), \tag{2.2}$$

где $\dot{p_i}$ — скорость агента.

Адаптация метода к задаче

Для нашей задачи потенциал должен учитывать:

— Привлечение агентов к критическим точкам и областям с высокой функцией опасности.

— Отталкивание агентов от препятствий и других агентов.

— Отталкивание агентов от областей с высокой плотностью угроз.

Потенциал определяется по формуле (2.3):

$$V(p_i) = \sum_{k=1}^{c} w_k \cdot V_{\text{кр}}(p_i, q_k) + \sum_{j=1}^{b} w_j \cdot V_{\text{преп}}(p_i, o_j) + \sum_{\tau=1}^{t} w_\tau \cdot V_{\text{угр}}(p_i, \tau), \quad (2.3)$$

где:

— $V_{\text{кр}}(p_i, q_k)$ — аттрактор (2.4), притягивающий a_i к критической точке q_k.

— $V_{\text{преп}}(p_i, o_j)$ — репеллент (2.5), отталкивающий a_i от препятствия o_j.

— $V_{\text{угр}}(p_i, \tau)$ — репеллент (**??**), отталкивающий a_i от угрозы τ.

— w_k, w_j, w_τ — весовые коэффициенты.

Каждая компонента определяется как:

$$V_{\text{кр}}(p_i, q_k) = -\frac{1}{\|p_i - q_k\| + \epsilon}, \quad (2.4)$$

$$V_{\text{преп}}(p_i, o_j) = \frac{1}{\|p_i - o_j\|^2 + \epsilon}, \quad (2.5)$$

$$V_{\text{угр}}(p_i, \tau) = \frac{1}{\|p_i - \tau\|^2 + \epsilon}, \quad (2.6)$$

где $\epsilon > 0$ предотвращает деление на ноль.

Алгоритмическая сложность метода

Сложность метода определяется трудоемкостью вычислений потенциала для каждого агента.

Требуется учесть каждую критическую точку q_k, препятствие o_j и угрозу τ, а также влияние функции опасности.

Итоговая сложность для одного агента характеризуется формулой (2.7):

$$O_{\text{агент}} = O(c + b + t). \tag{2.7}$$

Суммарная сложность для всех агентов характеризуется формулой (2.8):

$$O_{\text{общая}} = O(a \cdot (c + b + t)). \tag{2.8}$$

Классификация метода

Тип взаимодействия: Метод является **децентрализованным**, так как каждый агент принимает решения на основе локальных вычислений потенциала.

Область восприятия: Метод использует **локальное восприятие**, ограниченное областью действия потенциала [10].

Распределение задач: Задачи распределяются **динамически** в процессе вычисления градиента.

Сложность: Итоговая сложность $O(a \cdot (c + b + t))$ является линейной относительно числа агентов a и элементов среды. Это позволяет применять метод в режиме реального времени.

Гибкость: Метод обладает **высокой адаптивностью**, так как параметры потенциалов можно изменять в зависимости от текущих условий.

Правдоподобность: Метод обеспечивает **высокий уровень правдоподобности** при тщательной настройке параметров, что показано в [2].

2.2.2 Метод ролей

Метод ролей основывается на назначении фиксированных функций или ролей агентам, которые определяют их поведение и задачи в системе [9].

Для применения к нашей задаче данный метод должен учитывать распределение агентов по функциям патрулирования, защиты критических областей и нейтрализации угроз.

Общее описание метода ролей

В методе ролей каждому агенту a_i назначается роль r_i из множества допустимых ролей R, определяемого формулой (2.9):

$$r_i \in R = \{\text{патрулирование}, \text{защита}, \text{перехват}\}. \tag{2.9}$$

Каждая роль имеет свои задачи:

— **Патрулирование:** агент перемещается по маршруту, покрывающему определенную область.

— **Защита:** агент остается вблизи критической точки и контролирует угрозы в ее окружении.

— **Перехват:** агент направляется к обнаруженной угрозе для ее нейтрализации.

Назначение ролей может быть статическим (фиксированное распределение) или динамическим (меняется в зависимости от ситуации). В рассматриваемой задаче применимо **динамическое распределение**, где роли пересматриваются в реальном времени на основе состояния среды.

Адаптация метода к задаче

Адаптация метода ролей требует учета следующих факторов:

— Выбор ролей агентов в зависимости от текущего уровня опасности $\mathcal{U}(x, y, t)$, положения критических точек и угроз.

— Оптимизация распределения ролей для минимизации времени реакции на угрозы и покрытия областей.

Процесс распределения ролей описывается формулой (2.10):

$$r_i = \arg\min_{r \in R} C(r, p_i, \mathcal{U}, T), \tag{2.10}$$

где $C(r, p_i, \mathcal{U}, T)$ — функция стоимости назначения роли r агенту a_i, зависящая от его положения p_i, функции опасности $\mathcal{U}$ и текущего набора угроз T.

Функция стоимости для ролей определяется формулами (2.11)-(2.13):

$$C_{\text{патрулирование}} = \sum_{(x,y)\in A_i} \mathcal{U}(x, y, t), \tag{2.11}$$

$$C_{\text{защита}} = \sum_{q_k \in Q} \frac{1}{\|p_i - q_k\| + \epsilon}, \tag{2.12}$$

$$C_{\text{перехват}} = \min_{\tau \in T} \frac{\|p_i - \tau\|}{v_i}, \tag{2.13}$$

где:

— A_i — область, закрепленная за агентом a_i для патрулирования.

— Q — множество критических точек.

— v_i — скорость агента a_i.

Алгоритмическая сложность метода

Рассмотрим сложность распределения ролей.

— Для патрулирования необходимо вычислить сумму значений функции опасности по закрепленной области A_i, что требует $O(v)$ операций (по числу вершин графа).

— Для защиты требуется рассчитать расстояние до всех критических точек, что требует $O(c)$ операций.

— Для перехвата необходимо вычислить расстояние до всех угроз, что требует $O(t)$ операций.

Итоговая сложность распределения ролей для одного агента характеризуется формулой (2.14):

$$O_{\text{агент}} = O(v + c + t). \tag{2.14}$$

Суммарная сложность для всех агентов оцененивается формулой (2.15):

$$O_{\text{общая}} = O(a \cdot (v + c + t)). \tag{2.15}$$

Классификация метода

Тип взаимодействия: Метод является **централизованным**, так как роли назначаются всем агентам на основе глобальных данных о среде.

Область восприятия: Метод использует **глобальное восприятие**, так как распределение ролей требует информации о всей системе.

Распределение задач: Распределение задач является **динамическим**, так как роли пересматриваются на основе текущего состояния среды.

Сложность: Сложность $O(a \cdot (v+c+t))$ является линейной относительно числа агентов a, что позволяет использовать метод в режиме реального времени, если количество вершин v остается не слишком большим.

Гибкость: Метод обладает **высокой гибкостью**, так как роли могут быть адаптированы к изменениям в среде.

Правдоподобность: Метод обеспечивает **низкий уровень правдоподобности**, так как агенты обладают ограниченным количеством паттернов поведения, которое задается числом ролей.

2.2.3 Метод роя частиц

Метод роя – это метод роевого интеллекта, используемый для оптимизации сложных нелинейных целевых функций, основанный на итеративном улучшении решения-кандидата с учетом заданного показателя качества [11].

Общее описание метода роя

Поведение каждого агента a_i определяется рядом локальных правил:

1. **Притяжение:** движение к центру масс соседних агентов описывается формулой (2.16):

$$f_{\text{притяжение}}(p_i) = k_{\text{пр}} \cdot \left(\frac{1}{|N_i|} \sum_{a_j \in N_i} p_j - p_i \right), \tag{2.16}$$

где N_i — множество агентов в радиусе восприятия $r_{\text{воспр}}$, $k_{\text{пр}}$ — коэффициент притяжения.

2. **Избегание:** отталкивание от слишком близко расположенных агентов

описывается формулой (2.17):

$$f_{\text{избегание}}(p_i) = \sum_{a_j \in N_i^{\text{близ}}} k_{\text{изб}} \cdot \frac{p_i - p_j}{\|p_i - p_j\|^3}, \tag{2.17}$$

где $N_i^{\text{близ}}$ — множество агентов в пределах малого радиуса $r_{\text{мин}}$, $k_{\text{изб}}$ — коэффициент избегания.

3. **Выравнивание:** согласование направления движения описывается формулой (2.18):

$$f_{\text{выравнивание}}(p_i) = k_{\text{выр}} \cdot \left(\frac{1}{|N_i|} \sum_{a_j \in N_i} \dot{p}_j - \dot{p}_i \right), \tag{2.18}$$

где $k_{\text{выр}}$ — коэффициент выравнивания.

Суммарное движение агента может быть описано формулой (2.19):

$$\dot{p}_i = f_{\text{притяжение}}(p_i) + f_{\text{избегание}}(p_i) + f_{\text{выравнивание}}(p_i). \tag{2.19}$$

Адаптация метода к задаче

Для нашей задачи метод роя модифицируется следующим образом: 1. **Учет функции опасности** $\mathcal{U}(x, y, t)$: агенты перемещаются в области, где $\mathcal{U}$ имеет высокие значения. Это обеспечивается добавлением аттрактора (2.20):

$$f_{\text{опасность}}(p_i) = -k_{\text{оп}} \cdot \nabla \mathcal{U}(p_i, t), \tag{2.20}$$

где $k_{\text{оп}}$ — коэффициент чувствительности к опасности.

2. **Нейтрализация угроз**: агенты в области видимости угроз τ перенаправляются к ним. Дополнительное движение определяется как (2.21):

$$f_{\text{угроза}}(p_i) = -k_{\text{угр}} \cdot \sum_{\tau \in T_{\text{вид}}} \frac{p_i - \tau}{\|p_i - \tau\|^3}, \tag{2.21}$$

где $T_{\text{вид}}$ — множество угроз в радиусе видимости агента.

3. **Балансирование покрытия и плотности**: для предотвращения

скопления агентов используется штраф за высокую плотность (2.22):

$$f_{\text{разрежение}}(p_i) = k_{\text{разр}} \cdot \left(\frac{1}{|N_i|} - \rho_{\text{целевая}} \right), \tag{2.22}$$

где $\rho_{\text{целевая}}$ — целевая плотность агентов.

Итоговое движение агента описывается формулой (2.23):

$$\dot{p}_i^{total} = \dot{p}_i + f_{\text{опасность}}(p_i) + f_{\text{угроза}}(p_i) + f_{\text{разрежение}}(p_i). \tag{2.23}$$

Алгоритмическая сложность метода

Сложность метода роя определяется числом соседей каждого агента. Обозначим среднее число соседей как $|N_i|$.

— На вычисление взаимодействий для одного агента требуется $O(|N_i|)$.

— Учет функции опасности требует $O(v)$ операций для каждого агента, так как $\mathcal{U}$ задается на графе.

— Для обнаружения угроз в радиусе видимости необходимо $O(t)$.

Итоговая сложность для одного агента характеризуется формулой (2.24):

$$O_{\text{агент}} = O(|N_i| + v + t). \tag{2.24}$$

Общая сложность характеризуется формулой (2.25):

$$O_{\text{общая}} = O(a \cdot (|N_i| + v + t)). \tag{2.25}$$

При фиксированном радиусе восприятия $r_{\text{воспр}}$, трудоемкость остается практически постоянной, что делает метод подходящим для реального времени.

Классификация метода

Тип взаимодействия: Метод является **децентрализованным**, так как агенты принимают решения на основе локальной информации [12].

Область восприятия: Метод использует **локальное восприятие**, ограниченное радиусом $r_{\text{воспр}}$.

Распределение задач: Задачи распределяются **динамически** на основе взаимодействия с функцией опасности и угрозами.

Сложность: Сложность $O(a \cdot (|N_i| + v + t))$ линейна относительно числа агентов a и остается практически константной относительно $|N_i|$, что подходит для игр в реальном времени.

Гибкость: Метод обладает **низкой гибкостью**, так как поведения агентов в основном продиктовано необходимостью группироваться и повторять действия соседних агентов, реагирующих на локальные изменения среды.

Правдоподобность: Метод обеспечивает **высокий уровень правдоподобности**, так как движение агентов обладает свойствами естественного поведения роя [12].

2.2.4 Метод планирования на основе теории игр

Метод планирования на основе теории игр предполагает, что агенты взаимодействуют, решая задачи оптимального поведения в многоагентной среде через формирование и решение математической модели игры [13].

Общее описание метода

Модель задачи представляется как стратегическая игра (A, U), где:

— $A = \{A_1, A_2, \ldots, A_a\}$ — множество агентов;

— S_i — множество стратегий i-го агента;

— $U_i : S_1 \times S_2 \times \cdots \times S_a \to \mathbb{R}$ — функция выигрыша i-го агента, зависящая от стратегий всех агентов.

В ходе игры каждый агент выбирает стратегию $s_i \in S_i$, стремясь максимизировать свою функцию выигрыша U_i [14]. Решение задачи игры определяется через нахождение равновесий, например, равновесия Нэша, которые удовлетворяют условию (2.26):

$$U_i(s^*_{-i}, s^*_i) \geq U_i(s^*_{-i}, s_i) \quad \forall s_i \in S_i, \tag{2.26}$$

где s^*_{-i} — стратегии всех агентов, кроме i, в равновесии.

Адаптация метода к задаче

Для нашей задачи метод планирования на основе теории игр модифицируется следующим образом:

1. **Множество стратегий**: Каждый агент выбирает маршрут и целевую область покрытия. Множество стратегий S_i для агента i включает все возможные пути вдоль графа среды, ведущие к областям покрытия.

2. **Функция выигрыша**: Функция выигрыша U_i определена как (2.27):

$$U_i(s_i, s_{-i}) = -\alpha \mathcal{U}(p_i) - \beta \sum_{c_k \in C} \mathcal{U}(c_k) + \gamma \sum_{\tau \in T} d(p_i, \tau), \tag{2.27}$$

где:

— $\mathcal{U}(p_i)$ — значение функции опасности в целевом положении p_i агента;

— $\mathcal{U}(c_k)$ — значение функции опасности в критической области c_k;

— $d(p_i, \tau)$ — расстояние до угрозы τ;

— α, β, γ — веса, задающие приоритеты поведения.

3. **Решение игры**: Игра решается в реальном времени через итеративное приближение равновесия Нэша [14]. Для этого каждый агент оптимизирует свою стратегию s_i, исходя из стратегий остальных агентов s_{-i} согласно (2.28).

$$s_i^* = \arg\max_{s_i \in S_i} U_i(s_i, s_{-i}^*). \tag{2.28}$$

4. **Нейтрализация угроз**: Если угроза τ находится в области видимости агента i, стратегия агента автоматически переходит к ее преследованию и нейтрализации выбирая стратегию по формуле (2.29):

$$s_i = \arg\min_{s_i \in S_i} d(p_i, \tau). \tag{2.29}$$

Алгоритмическая сложность метода

Сложность метода определяется числом агентов, стратегий и итераций поиска равновесия:

— Для каждого агента построение множества стратегий S_i требует $O(v^2)$, при применении алгоритма Дейкстры для построения кратчайших путей.

— Оценка функции выигрыша U_i для всех стратегий S_i требует $O(|S_i|)$.

— Поиск равновесия итеративным методом (например, методом наивной оптимизации) требует $O(k \cdot a \cdot |S_i|)$, где k — число итераций до сходимости.

При этом можно принять, что $|S_i| \leq v^2$ и $k = const$, тогда итоговая сложность характеризуется формулой (2.30):

$$O_{\text{общая}} = O(a \cdot v^2). \tag{2.30}$$

Классификация метода

Тип взаимодействия: Метод относится к **централизованному типу**, так как агенты используют информацию о стратегиях, выбранных другими агентами [13].

Область восприятия: Метод использует **локальное восприятие**, так как стратегии формируются на основе окружения агента.

Распределение задач: Задачи распределяются **статически**, поскольку стратегия выбирается на основе рассчитаного оптимума [15].

Сложность: Метод имеет сложность $O(a \cdot v^2)$, что делает его трудоемким для сред с большим размером локации.

Гибкость: Метод обладает **высокой** гибкостью, поскольку стратегия агента адаптируется на основе его окружения.

Правдоподобность: Правдоподобность метода **высокая**, так как поведение агентов, основанное на игровых стратегиях, соответствует ожиданиям от разумной координации [15].

2.2.5 Метод на основе обучения с подкреплением

Общее описание метода

Метод Q-обучения относится к виду методов обучения с подкреплением. Данный метод позволяет моделировать поведение агентов как процесс последовательного принятия решений в среде. Среда представляется в виде

марковского процесса принятия решений [16], который задается пятеркой (S, A, P, R, γ):

— S — множество состояний среды,

— A — множество возможных действий агента,

— $P(s'|s, a)$ — функция переходов между состояниями при выполнении действия a,

— $R(s, a)$ — функция вознаграждения за выполнение действия a в состоянии s,

— $\gamma \in [0, 1]$ — коэффициент дисконтирования будущих вознаграждений.

Цель агента заключается в максимизации ожидаемой суммарной дисконтированной награды (2.31):

$$G_t = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R(s_{t+k}, a_{t+k})\right]. \tag{2.31}$$

Агент учится стратегии $\pi(a|s)$, которая задает вероятность выбора действия a в состоянии s. Обучение стратегии происходит на основе значений функции полезности (2.32):

$$Q^{\pi}(s, a) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R(s_{t+k}, a_{t+k}) \middle| s_t = s, a_t = a\right]. \tag{2.32}$$

Адаптация метода к задаче

Для задачи визуального покрытия критических областей вводится специфическая структура состояния, действий, и функции награды:

— **Состояния (S):** Состояние агента включает:

- Текущую позицию p_i и скорость $\dot{p}_i$ агента,
- Значения функции опасности $\mathcal{U}(x, y, t)$ в окрестности агента,
- Расположение ближайших критических точек и угроз.

— **Действия (A):** Агенты могут выбирать движение в одном из k направлений, задаваемых дискретизацией пространства, или оставаться на месте.

— **Функция вознаграждения (R):** Вознаграждение определяется функцией (2.33):

$$R(s,a) = -\alpha \cdot \mathcal{U}(p_i,t) + \beta \cdot I_{\text{угроза\_нейтрализована}} - \gamma \cdot \mathcal{L}(p_i), \qquad (2.33)$$

где $\mathcal{L}(p_i)$ — штраф за выход за границы области, α, β, γ — коэффициенты весов, $I_{\text{угроза\_нейтрализована}}$ — индикатор нейтрализации угрозы.

Для обучения стратегии используется симуляция среды: агенты взаимодействуют с функцией опасности $\mathcal{U}(x,y,t)$, перемещаются между вершинами графа и реагируют на появление угроз τ. Модель среды обновляется согласно описанной динамике.

Алгоритмическая сложность метода

Обучение с подкреплением включает два основных этапа:

1. **Симуляция среды:** При фиксированном числе агентов a симуляция одного шага занимает $O(a \cdot v)$ операций, так как необходимо обновить функцию $\mathcal{U}$ и обработать поведение каждого агента.

2. **Обновление стратегии:** Для алгоритма Q-обучения требуется обновление таблицы $Q(s,a)$, что занимает $O(|S| \cdot |A|)$ операций.

При использовании Deep Q-Learning сложность определяется числом параметров нейронной сети $n_{\text{параметры}}$.

Значения $|S|$, $|A|$, $n_{\text{параметры}}$ фиксированы, так что итоговая сложность обучения характеризуется формулой (2.34):

$$O_{\text{обучение}} = O(N \cdot (a \cdot v + |S| \cdot |A|)), \qquad (2.34)$$

где N — число шагов симуляции.

Применение обученной стратегии в реальном времени требует $O(a)$ операций на каждом шаге.

Классификация метода

Тип взаимодействия: Метод является **децентрализованным**, агенты обучаются индивидуально и учитывают окружение через локальное состояние.

Область восприятия: Метод использует **локальное восприятие**, так как состояние включает информацию об окружении агента и его параметрах.

Распределение задач: Распределение задач **статическое** во время выполнения, так как стратегия фиксируется после этапа обучения [16].

Сложность: Обучение требует больших вычислительных ресурсов ($O_{\text{обучение}}$), однако применение стратегии после обучения не требует трудоемких вычислений.

Гибкость: Метод обладает **высокой гибкостью**, так как стратегия может адаптироваться к сложным динамическим сценариям.

Правдоподобность: Метод обеспечивает **высокую правдоподобность**, так как обученные стратегии могут воспроизводить реалистичное поведение при корректной настройке параметров [17].

3 Сравнение методов координации агентов в многоагентных системах

На основе классификации методов, описанных выше, проведем сравнительный анализ по критериям, сформулированным ранее. В таблицах 3.1-3.2 приведены ключевые характеристики методов координации.

Таблица 3.1 – Сравнение методов координации агентов в MAC (часть 1/2)

Метод	**Тип взаимодействия**	**Область восприятия**	**Распределение задач**
Потенциальных полей	Децентрализованный	Локальная	Динамическое
Ролей	Централизованный	Глобальная	Динамическое
Роя частиц	Децентрализованный	Локальная	Динамическое
Теоретико-игровой	Централизованный	Локальная	Статическое
Обучения с подкреплением	Децентрализованный	Локальная	Статическое

Таблица 3.2 – Сравнение методов координации агентов в MAC (часть 2/2)

Метод	**Сложность**	**Гибкость**	**Правдоподобность**
Потенциальных полей	$O(a \cdot (c + b + t))$	Высокая	Высокая
Ролей	$O(a \cdot (v + c + t))$	Высокая	Низкая
Роя частиц	$O(a \cdot (\|N_i\| + v + t))$	Низкая	Высокая
Теоретико-игровой	$O(a \cdot v^2)$	Высокая	Высокая
Обучения с подкреплением	$O(a)$	Высокая	Высокая

Вывод

Исходя из сравнительной таблицы можно сделать следующие выводы:

1. Наименее трудоемким для применения является метод обучения с подкреплением, однако он плохо интерпретируем и трудоемок при обучении.
2. Метод потенциальных полей и теоретико-игровой методы обеспечивают высокую степень гибкости и правдоподобности движения агентов, однако трудоемкость работы алгоритмов, реализующих теоретико-игровой метод выше, чем у метода потенциальных полей.

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Dorri A.*, *Kanhere S. S.*, *Jurdak R.* Multi-Agent Systems: A Survey. — 2018. — DOI: 10.1109/ACCESS.2018.2831228.
2. Path-Planning for RTS Games Based on Potential Fields / R. Silveira [и др.]. — 11.2010. — DOI: 10.1007/978-3-642-16958-8_38.
3. *Jong S. de*, *Tuyls K.*, *Sprinkhuizen-Kuyper I.* Robust and Scalable Coordination of Potential-Field Driven Agents. — 2006. — DOI: 10.1109/CIMCA.2006.191.
4. An Overview of Recent Progress in the Study of Distributed Multi-Agent Coordination / Y. Cao [и др.]. — 2013. — DOI: 10.1109/TII.2012.2219061.
5. *Stone P.*, *Veloso M.* Multiagent Systems: A Survey from a Machine Learning Perspective. — 2000.
6. *Wooldridge M.* An Introduction to MultiAgent Systems. — 2009.
7. *Ren W.*, *Cao Y.* Overview of Recent Research in Distributed Multi-agent Coordination. — London, 2011. — DOI: 10.1007/978-0-85729-169-1_2.
8. A game engine to make games as multi-agent systems / C. Marín-Lora [и др.]. — 2020. — DOI: https://doi.org/10.1016/j.advengsoft.2019.102732.
9. *Cabri G.*, *Ferrari L.*, *Leonardi L.* Agent role-based collaboration and coordination: a survey about existing approaches. — 2004. — DOI: 10.1109/ICSMC.2004.1401064.
10. Exploratory Navigation Based on Dynamical Boundary Value Problems / M. Trevisan [и др.]. — 02.2006. — DOI: 10.1007/s10846-005-9008-2.
11. *Zhang H.*, *Hui Q.* Multiagent Coordination Optimization: A control-theoretic perspective of swarm intelligence algorithms. — 2013. — DOI: 10.1109/CEC.2013.6557979.
12. *Meng Y.*, *Kazeem O.*, *Muller J. C.* A Swarm Intelligence Based Coordination Algorithm for Distributed Multi-Agent Systems. — 2007. — DOI: 10.1109/KIMAS.2007.369825.
13. *Гуревич Л. А.*, *Вахитов А. Н.* Мультиагентные системы. — 2005.

14. *Parsons S.*, *Wooldridge M.* Game Theory and Decision Theory in Multi-Agent Systems. — 09.2002. — DOI: 10.1023/A:1015575522401.

15. *Pendharkar P. C.* Game theoretical applications for multi-agent systems. — 2012. — DOI: https://doi.org/10.1016/j.eswa.2011.07.017.

16. *Xuan P.*, *Lesser V.*, *Zilberstein S.* Communication in multi-agent Markov decision processes. — 2000. — DOI: 10.1109/ICMAS.2000.858528.

17. *Matignon L.*, *Laurent G.*, *Fort-Piat N.* Hysteretic Q-learning : an algorithm for Decentralized Reinforcement Learning in Cooperative Multi-Agent Teams. — 12.2007. — DOI: 10.1109/IROS.2007.4399095.

ПРИЛОЖЕНИЕ А

Презентация к расчетно-пояснительной записке состоит из 20 слайдов.