*Кабыш А.С.*
*Научный руководитель: д.т.н., профессор Головко В.А.*

# МОДЕЛЬ КООРДИНАЦИИ ПОВЕДЕНИЯ АГЕНТОВ НА ОСНОВЕ ОБУЧЕНИЯ С ПОДКРЕПЛЕНИЕМ

**Резюме:** *В данной работе описывается модель для нахождения оптимального поведения многоагентной структуры через организацию в ней оптимальных взаимодействий между агентами. Модель включает две основные техники. Модель графов координации позволяет явно выразить зависимость между агентами, что позволяет разбить целевую функцию поведения в линейную сумму индивидуальных целевых функций. Модель оценки влияний позволяет оценить влияния других агентов на действия друг друга и в результате позволяет им координировать свои действия. В работе приведена реализация данной модели на основе обучения с подкреплением и экспериментальные результаты применения данной модели.*

**Ключевые слова:** *многоагентные системы, обучение с подкреплением, Q-Learning, обучение через влияние, граф координации*

## Введение

Огромный класс задач сводится к теории многоагентных систем, где рассматриваются способы взаимодействия между агентами. Типовые задачи, решаемые в теории многоагентных систем, включают в себя [1]: автоматизированный трейдинг, ведение переговоров, управление нагрузкой сети, организацию коллектива роботов-футболистов, распределенную оптимизацию, распределенное планирование задач [2] и другие.

Традиционно **многоагентная система** состоит из совокупности агентов, разработанных для кооперации друг с другом для достижения некоторой цели. **Агент** понимается как сущность, обладающая состоянием, способная воспринимать окружающую среду и выполнять в ней какие-то действия.

Свойства и поведение многоагентной системы зависят от суммарных свойств и поведения входящих в неё агентов. В зависимости от алгоритма коллективного поведения многоагентная система выдвигает различные требования к агентам, которые могут быть использованы в ней для данного алгоритма. Например большинство алгоритмов стайного поведения не предполагает наличия глобальной коммуникации между агентами, а алгоритмы ведения переговоров требуют, чтобы агенты были разными и по возможности держали в секрете свои модели.

Представленная в данной статье модель также предъявляет ряд требований к агентам. Агент рассматривается как **актор** [3] – активный агент, существующий параллельно (и одновременно) с другим таким же агентом, взаимодействующий с другими посредством стандартного протокола внутреннего взаимодействия в виде сообщений. Агенты должны обладать способностью к коммуникации, а многоагентная система – обеспечивать параллельность работы агентов.

Стоит отметить, что коммуникация подразумевает обмен сообщениями в каком-либо виде, а не восприятие другого агента посредством сенсоров. Агент должен иметь восприятие среды и восприятие сообщений, как нечто разделенное. Модель не специфицирует конкретный способ коммуникации между аген-

тами. Это может быть как прямой обмен сообщениями, так и использование среды, в качестве посредника для передачи сообщений, однако это повлияет на динамику работы алгоритма.

При обучении многоагентных систем либо изобретаются новые, гибридные алгоритмы, например в [4], либо адаптируются уже существующие. Во втором подходе разрабатывается какой-либо новый алгоритм, регулирующий способ совместной работы набора уже существующих алгоритмов, примененных к отдельным агентам.

Существуют 4 основные парадигмы адаптации алгоритмов обучения к многоагентным системам [5]:

• **Командное обучение** (*team learning*) развивает идею, что команду агентов можно обучать так, как если бы это был один агент. Этот подход не требует никаких изменений в алгоритмах обучения, но имеет ограничения в применимости. При увеличении количества агентов экспоненциально растет и размер пространства состояний/действий, в котором ведется поиск решения (проблема проклятья размерности).

• **Индивидуальное обучение** (*individual learning*) рассматривает применение индивидуального алгоритма обучения каждому агенту, игнорируя дополнительные данные от других агентов. Проблемой данного подхода является то, что совокупность оптимальных индивидуальных стратегий не обязательно представляет оптимальную командную стратегию.

• **Совместное обучение** (*joint action learning*) фокусируется на обучении оптимальным действиям в ответ на действия других агентов. Каждый агент обучается выполнять наилучшие действия в объединенном контексте действий других агентов. Этот подход позволяет построить оптимальную коллективную стратегию, но обладает тем же проклятьем размерности, что и командное обучение.

• **Обучение через влияние** (*influence-value learning*) основано на идее изменения поведения агента под влиянием мнения других агентов. Данный подход берет на вооружение множественные социальные факторы и использует их в качестве эвристик для конкретизации отношений внутри многоагентной системы.

В данной работе будет рассмотрена обобщенная, адаптивная модель организации взаимодействий внутри многоагентной системы на основе обучения через влияние для формирования в ней целенаправленного поведения посредством нахождения оптимальных взаимодействий между агентами. Под ожидаемым поведением многоагентной системы понимается совокупное поведение входящих в неё агентов, которое можно оценить как одно целое, приводящее к достижению некоторой цели. Примерами таких целей могут быть формирование и поддержание формации, коллективное управление нагрузкой или ресурсами, построение целостной карты окружающего пространства и др.

Модель обладает следующими свойствами:

• Данная модель является обобщенной, т.е. она не предъявляет никаких первоначальных требований к агентам, кроме способности получать и отправлять сообщения другим агентам.

• Модель является адаптивной в том, что она позволяет изменить структуру многоагентной системы и поведение входящих в неё агентов, если изменилась целевая функция поведения.

• Модель фокусируется на определении тех оптимальных взаимодействий между агентами, которые приводят к решению поставленной задачи, сохраняя аспекты индивидуального обучения.

## 1. Модель организации коллективного поведения

Ключевой концепцией является понятие **взаимодействия** или **влияния** между агентами. В реальном мире, при взаимодействии людей друг с другом, все действия отдельного индивида оцениваются, как и им самим, так и другими людьми в разрезе получаемого опыта. Примерами таких оценок являются ожидания, похвалы или наказания до или после совершенного действия.

В коллаборативной многоагентной системе все агенты могут потенциально влиять друг на друга. Под **влиянием** (*influence*) понимается оценка другими агентами, направленная на текущего агента. Другие агенты могут оценивать поведение агента и его регулировать с той целью, чтобы действия, выбранные индивидуальным агентом, соответствовали оптимальным решениям группы в целом. Любой агент, инициирующий влияние на другого агента, называется *отправителем*. Агент, принимающий влияние и как-то на него реагирующий, называется *получателем*.

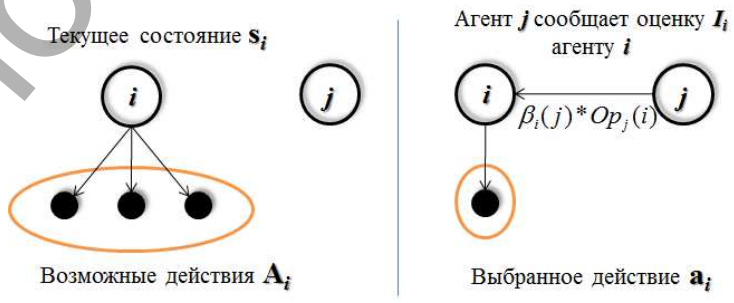
Каждый агент $i$ выбирает индивидуальное действие $a_i$ из множества $A_i$, находясь в некотором состоянии. Значение влияния между агентом $i$ и группой из N агентов относительно выбранного агентом действия определяется следующей формулой [5]:

$$I_i = \sum_{j=1, i \neq j}^{N} \beta_i(j) * Op_j(i), \qquad (1)$$

где $\beta_i(j)$ – коэффициент влияния агента $j$ на агента $i$ $(0 \leq \beta \leq 1)$, $Op_j(i)$– коэффициент оценки агентом $j$ действия агента $i$.

Коэффициент влияния β определяет, будет ли агент подвержен влиянию других агентов. Так, если β будет стремиться к нулю, то агент предпочтет действовать индивидуально. *Op* – это оценка другими агентами действия выполняемого текущим агентом.

Рисунок 1 показывает пример формирования влияния агентом $j$ после выбора агентом $i$ действия.




(a) (b)

***Рисунок 1 – Пример модели влияний. (a) – Агент i выбирает действие. Выбрав действие, агент j сообщил свою оценку i, сформировав влияние***

Многоагентная система формирует результирующее объединенное действие (joint action) $a = \{a_1, a_2, ..., a_N\}$. *Проблема координации* [6] поведения состоит в том, чтобы найти такое оптимальное объединенное действие $a^*$, которое мак-

симизирует полезность объединённого действия коллективного поведения системы $u(a)$, что $a^* = arg\,max_a\, u(a)$.

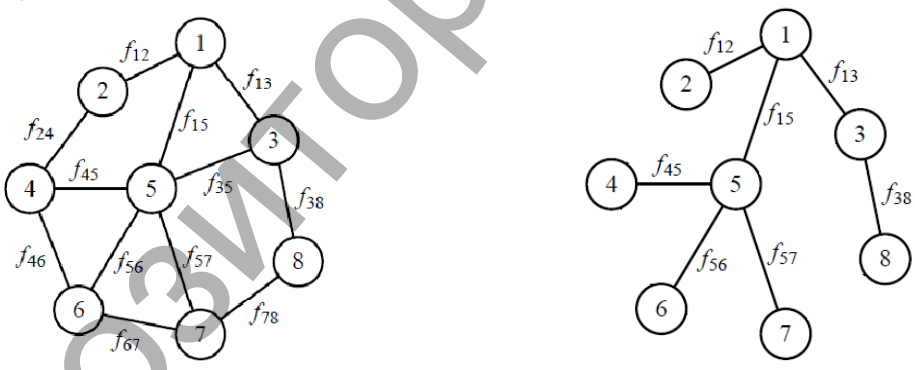
В большинстве реальных проблем действия одного агента зависят лишь от небольшого числа других агентов. Например, в задаче коллективного сбора ресурсов, агенты могут координировать свои действия только с соседями. Для явного отслеживания таких зависимостей предложена модель **графов координации** (coordination graphs, (CG)) [7].

Пускай действия агента $i$ зависят только от некоторого подмножества других агентов, $j \in M(i)$. Тогда целевая функция многоагентной системы $u(a)$ может быть разбита на линейную комбинацию локальных целевых функций по следующей формуле:

$$u(a) = \sum_{i=1}^{N} f_i(sub_i), \qquad (2)$$

где $sub_i$ подмножество всего множества действий $a$, соответствующее действиям агентов $j \in M(i)$, от которых зависит агент $i$. Таким образом, глобальная проблема координации заменена на совокупность локальных проблем координации, каждая из которых включает малое подмножество агентов.

Зависимость между агентами может быть отражена в виде графа координации, где вершины соответствуют агентам, а ребра связывают зависимых агентов, которые обязаны координировать свои действия. По ребрам зависимостей агенты могут обмениваться сообщениями, а сами ребра специфицировать конкретный протокол общения между агентами. Пример графа координации изображен на рисунке 2.




(a) (b)

***Рисунок 2 – Пример графа координации для восьми агентов до (a) и после (b) декомпозиции зависимостей; каждое ребро представляет зависимость координации***

В данном разделе не были специфицированы следующие основные пункты:

- как происходит расчет коэффициента $Op_j(i)$ между агентами;
- как происходит перерасчет составляющих формулы (1) с течением времени;
- как агент изменяет свое поведение под воздействием влияний от других агентов;
- как агент оценивает влияние, оказываемое на него другими агентами;
- каков алгоритм нахождения оптимальных зависимостей в многоагентной системе;

Эти пункты специально были оставлены открытыми для максимальной обобщенности алгоритма под конкретную задачу и алгоритм.

Например, в работе [5] коэффициент $Op_j(i)$ увеличивался, если агент $j$ выполнял действие с меньшей ценностью, чем агент $i$, и наоборот, уменьшался, если агент $j$ выполнял более выгодное действие. Это отражает факт того, что люди думают хорошо о тех действиях, которые приносят другим больше прибыли. В следующем разделе представлена реализация данной модели на основе подкрепляющего обучения.

Сформулируем задачу организации многоагентной системы следующим образом:

- Пусть имеется N агентов, каждый их которых может выполнять некоторые действия (агенты могут быть как гетерогенными, так и гомогенными). Агенты имеют некоторое начальное состояние. Агенты объединены в многоагентную систему, обеспечивающую их параллельность и коммуникацию.
- Пусть имеется некоторая известная цель, достижение которой одним агентом либо невозможно, либо неэффективно по сравнению с многоагентным подходом. Агент или многоагентная система может узнать, достигнута цель или нет.
- Требуется определить такое поведение агентов, которое приводит к достижению цели оптимальным образом. В частных случаях данная задача уже может решаться различными способами [1], без коммуникации между агентами.
- С учетом коммуникации между агентами требуется определить оптимальные взаимодействия между ними, приводящие к коллективному решению задачи.
- Необходимо декомпозировать граф координации, при условии, что не имеется априорной информации о взаимодействиях между агентами, или она ограниченна.

Решение поставленной задачи выполняется посредством двух процессов, которые могут выполняться как последовательно, так и параллельно, в зависимости от используемых алгоритмов:

- Определение оптимальной структуры графа координации многоагентной системы с целью декомпозиции пространства решений.
- Определение оптимальных взаимодействий между агентами порождает оптимальное поведение.

## 2. Обучение с подкреплением для нахождения оптимальной структуры взаимодействий

Особенностью поставленной задачи является динамическое определение оптимальных влияний между агентами и графа координации. Для достижения этой цели требуется итеративный процесс настройки структуры многоагентной системы. За время функционирования многоагентной системы можно оценить разные взаимодействия между агентами, приводящими к решению задачи, и усилить конструктивные и ослабить деструктивные. Если взаимодействие между агентами не является необходимым, то, следовательно, координация между ними может не выполняться и зависимость на графе координации может быть устранена.

В качестве итеративного процесса оценки и формирования структуры многоагентной системы в большинстве работ [1] выступает **обучение с подкреплением** (*Reinforcement Learning*, (RL)) [9].

Ключевой особенностью данного метода является то, что он является активным методом обучения, направленным на взаимодействие со средой и с другими агентами в случае многоагентной системы. В классической трактовке, обучение с подкреплением – это нахождение оптимального поведения агента методом проб и ошибок через его взаимодействие со средой. В данной работе, по отношению к многоагентной системе, обучение с подкреплением – это нахождение оптимальной структуры системы методом проб и ошибок через взаимодействия агентов.

Модель обучения с подкреплением формулируется следующим образом. Агент $i$ выбирает действие $a_i(t)$ из множества $A_i$, находясь в состоянии $s_i(t)$. После выполнения действия агент переходит в следующее состояние $s_i(t+1)$ и получает из внешней среды числовое значение награды $r_i(t+1)$, являющееся оценкой выполненного действия и совершенного изменения состояния. В состоянии $s_i(t+1)$ агент выбирает и выполняет следующее действие $a_i(t+1)$. Разница между ценностью следующего состояния с учетом награды и ценностью текущего состояния называется **ошибкой временной разности** (temporal-difference error).

Последовательность изменений состояния агента изображена на рис. 3 в виде **графа переходов** (*transition graph*). Вершинами графа переходов являются состояния агента, а ребрами отмечаются выбранные действия и переходы между состояниями вместе с ассоциированной с переходом значением награды. Глобальное – вся среда, в которой действует агент – описывается в виде Марковского процесса принятия решений.

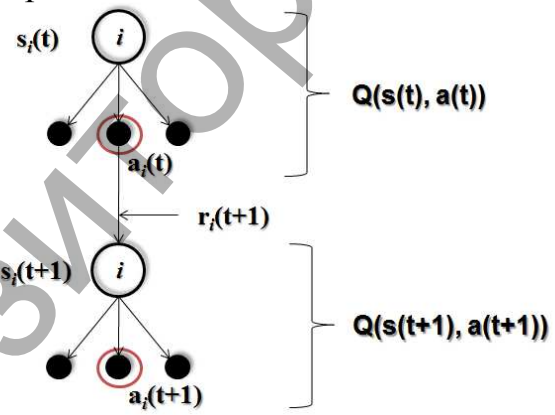



***Рисунок 3 – Граф переходов агента.***
***Изображен переход агента из состояния s(t) в состояние s(t+1)***

С любой парой «состояние-действие» $(s,a)$ ассоциировано значение ценности $Q(s,a)$, обозначающее полезность выполнения действия $a$ в состоянии $s$. Значения $Q(s,a)$ рассчитываются $Q$ – функцией и заранее неизвестны. Цель обучения – аппроксимировать истинные значения оценок $Q$ – функции путем последовательного посещения пар $(s,a)$ и получения ассоциированной с ними награды $r$, используя значение награды в качестве фактора, изменяющего ценность состояния. Способ отображения состояния $s$ на ассоциированное с ним действие $a$ называется политикой $\pi(s,a)$. Следовательно, задача обучения с подкреплением сводится к нахождению оптимальной политики, которая на ка-

ждом шаге выбирает оптимальные действия, ведущие к максимальной суммарной награде в будущем.

Самым популярным алгоритмом обучения с подкреплением является *Q – learning* [9], обновляющий значения функции ценности на каждой итерации по формуле (3):

$$Q(s(t),a(t)) = Q(s(t),a(t)) + \alpha[r(t+1) + \gamma max_{a \in A} Q(s(t+1),a(t+1)) - Q(s(t),a(t))], \quad (1)$$

где в квадратных скобках [.] рассчитывается ошибка временной разности, α – шаг обучения $(0 < \alpha \leq 1)$, а γ – коэффициент обесценивания $(0 < \gamma \leq 1)$ отдаленных ценностей.

Из алгоритма видно, что последующие значения ценностей влияют на предыдущие значения. Следовательно, в момент времени t(t+1) могут быть обновлены значения ценностей всех пар «состояние-действие», которые привели агента к текущему состоянию. Такая последовательность пар «состояние-действие» называется **следами преемственности** (*eligibility traces*). Модифицированная версия алгоритма называется Watkins-Q($\lambda$) [9].

Поскольку на начальном этапе обучения неизвестно, какие взаимодействия между агентами являются оптимальными, а какие нет, действия выбираются случайно, методом проб и ошибок (фаза исследования). В конце обучения истинные значения ценности аппроксимированы, и агент использует их как руководство для оптимального поведения (фаза использования) во внешней среде в контексте других агентов.

### 3. Поиск оптимальных команд в многоагентной системе

В разделе 1 было введено понятие влияния как контекста оценок, создаваемого другими агентами по отношению целевому. Рассмотрим частный случай модели влияний, где влияние выступает в виде частного случая – активной команды.

В отличие от влияния, **команда** – это (1) активное сообщение от отправителя к клиенту, способное изменить его состояние, принятое или проигнорированное, а также (2) оценка этого сообщения. Если влияние принято, то оно может изменить текущее состояние агента или его последующие действия. Если влияние не принято, то оно возвращается с соответствующей пометкой и отправитель может отметить данное влияние как неконструктивное либо не влияющее на получателя.

**Конструктивность команды** – это характеристика, оценивающая ценность данного влияния по отношению к задаче, решаемой многоагентной системой. Конструктивные влияния имеют положительную оценку полезности, с некоторыми ограничениями, в рамках которых эта полезность сохраняется.

Получатель может проигнорировать влияние либо ввиду того, что отправитель не имеет достаточно веса, с точки зрения получателя, либо получатель выполняет собственные «эгоистичные» действия, ведущие его к цели, и нечувствителен к командам.

Рассмотрим пример двух агентов *i* и *j*. Оба агента в момент времени *t* находятся в некоторых состояниях и готовы выбрать действие для перехода в следующее состояние. Пускай агент *i* может повлиять на агента *j*, указав, какое действие ему выполнить. Посылаемую команду можно трактовать как выбор

агентом $i$ некоторого действия $a_i(t)$, которое не переводит его в новое состояние $s_i(t+1)$. Агент $j$ принимает команду в качестве указания и выполняет указанное действие, $a_j(t)$, переходя в новое состояние $s_j(t+1)$ и получая награду за выполнение действия $r_j(t+1)$.

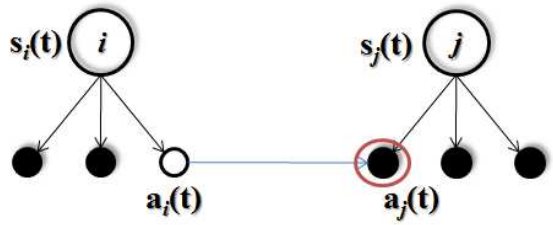



***Рисунок 4 – Диаграмма переходов для агентов i и j. Агент i инициирует команду над агентом j***

После перехода агент $j$ может скорректировать ценность перехода $Q(s_j(t),a_j(t))$ по формуле (3), а также выполнить оценку команды. В простейшем случае предполагаем, что коэффициент $\beta_i(j)$ равен 1. Значение $Op_j(i)$ содержит оценку агентом $i$ действия агента $j$. Примером оценки являются качественные показатели следующего состояния агента $j$: награда $r_j(t+1)$ и обновленное значение ценности $Q'(s_j(t),a_j(t))$. Эти показатели передаются в качестве обратной связи агенту $i$, который может использовать их для обновления значения ценности команды.

Если обратная связь постоянно отрицательна, то ценность команды $Q(s_i(t),a_i(t))$ падает. Таким образом, выполняя разные команды над агентом $j$ и получая обратную связь, агент $i$ обучается оптимальному управлению над ним, формируя диапазон состояний, при котором сохраняется конструктивность команды.

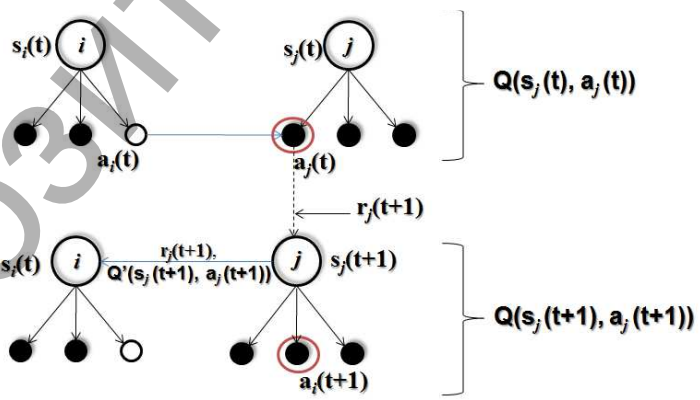



***Рисунок 5 – Диаграмма переходов для агентов i и j. Продолжение. Агент j выполнил команду, совершил переход, обновил значение ценности и вернул обратную связь агенту i***

Модифицированная формула для агента $i$, обновляющая ценность команды по оценке, полученной от агента $j$, имеет следующий вид:

$$\Delta Q(s_i(t),a_i(t)) = \alpha(Op_j(i) - Q(s_i(t),a_i(t)) \quad (1)$$

$$Op_j(i) = r_j(t+1) + \gamma Q_j^*(s_j(t+1),a_j(t+1) \quad (2)$$

$$Q_j^*(s_j(t+1),a_j(t+1)) = arg\ max_{a_j \in A_j} Q(s_j(t+1),a_j(t+1)) \quad (3)$$

Если агенты не имеют конструктивных отношений друг с другом, то зависимость в поведении между этими агентами не является необходимой, что ведет к естественной декомпозиции графа координации.

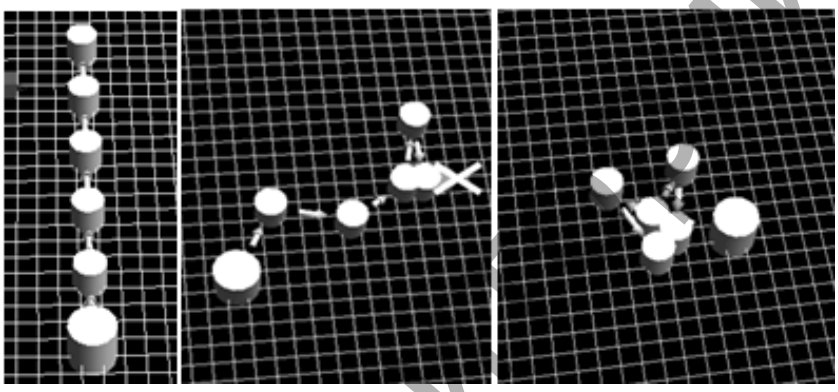
В этом примере агент $i$ не мог выполнить одновременно команду и действие, т.к. выполнение команды трактовалось как отдельное действие. В зависимости от модели многоагентной системы, агенты могут вести переговоры и смешивать команды и действия в один такт времени.

## 4. Эксперементальные результаты

Для описания графа координации, графа переходов и графа Марковских процессов принятия решений была разработана библиотека моделирования на графах [10], предназначенная для задач робототехники и многоагентного моделирования.

Описанный в главе 1 подход применялся для задачи формирования и поддержания формации заданной формы на модельных агентах [11].

Описанная в главе 3 адаптация на основе команд применялась в задаче моделирования многозвенного робота, обладающего 5-ю степенями свободы [12]. Среда моделирования и многоагентная система, имитирующая робота, изображена на рис. 6.



(a) (b) (c)

***(a) Начальное положение. (b) Оптимальная последовательность команд найдена. (c) «Организационный хаос»***

***Рисунок 6 – Моделирование многозвенного «робота-манипулятора»***

Каждый сегмент, кроме последнего, мог командами изменять положение всех последующих сегментов. Итоговое расстояние до цели определялось после выполнения действий всеми сегментами. Цель эксперимента – обучение робота достижению цели путем его самоорганизации посредством поиска оптимальных влияний между сегментами. Обученный робот оказался робастным к удалению или добавлению необученных сегментов, а также быстро переучивался при смене цели. Поскольку граф координации был задан заранее, явная декомпозиция пространства поиска по сегментам (текущий сегмент зависел только от предыдущего) показала эффективность в скорости сходимости алгоритма по сравнению с командным и совместным обучением, подверженным проклятью размерности. Рисунок 7 иллюстрирует графики уменьшения суммарной (по всем сегментам) ошибки временной разности для трех алгоритмов обучения с подкреплением в сравнении с совместным обучением (JAL).

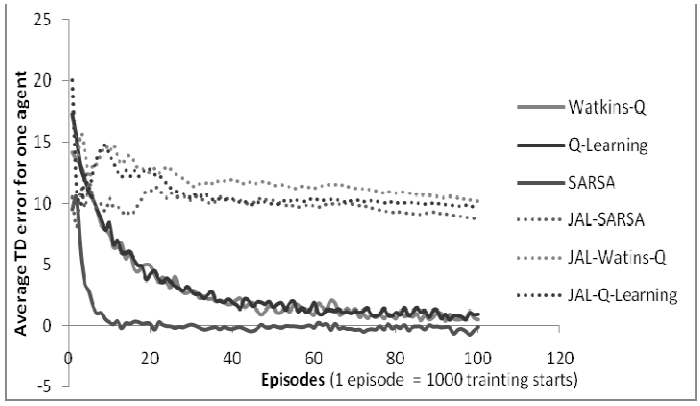



***Рисунок 7 – Графики изменения ошибки временной разности между совместным обучением и моделью команд***

## Выводы

Представленная в работе модель организации взаимодействий в многоагентной содержит набор техник, которые позволяют конкретизировать, оценить и оптимизировать отношения между агентами в системе с целью координации их действий. Представленный подход к адаптации рассмотренных концепций не единственный. В работах [5,6] представлены адаптации данного подхода для задач теории игр, коллективного фуражирования, коллективного принятия решений и ряда других.

Эксперимент с использованием модели показал, что эвристики, конкретизирующие отношения между агентами, применимы для широкого класса задач и могут быть более эффективны, чем изобретение нового алгоритма.

**Литература**

1. Panait, L. Cooperative Multi-Agent Learning: The State of the Art / L. Panait, S. Luke // *Autonomous Agents and Multi-Agent Systems*, 2005. – (11), 3. – P. 387-434.

2. Gabel, T. *Multi-Agent Reinforcement Learning Approaches for Distributed Job-Shop Scheduling Problems*. PhD Thesis / University of Osnabrueck. – 2009. – 175 p.

3. Lee, E. Actor-oriented design of embedded hardware and software systems / E. Lee, S. Neuendorffer, M.J. Wirthlin // *Journal of Circuits, Systems and Computers*. – 2003. – (12). – P. 231-260.

4. Monekosso, N. The analysis and performance evaluation of the pheromone-Q-learning algorithm / N. Monekosso, P. Remagnino // *Expert Systems* – 2004. – 21 (2). – P. 80-91.

5. Dennis, B. Goncalves, *Influence Value Q-Learning: A Reinforcement Learning Algorithm for Multi Agent Systems*, in: Meng Joo Er and Yi Zhou (Eds.), Theory and Novel Applications of Machine Learning, Book, I-Tech / B. Dennis, M. Luiz, G. Goncalves. – Vienna, Austria, 2009. – P. 376.

6. Kok, J.R. Sparse cooperative q-learning. *In Proceedings of the XXI international conference on Machine Learning* / J.R. Kok, N. Vlassis. – Banff, Alberta, Canada. – 2004. – P. 61.

7. Guestrin, C. Multiagent planning with factored MDPs. *In Advances in Neural Information Processing Systems* (NIPS) 14. The MIT Press / C. Guestrin, D. Koller, R. Parr. – 2002. – P. 1523-1530.

8. Kok, J. Using the max-plus algorithm for multiagent decision making in coordination graphs, *RoboCup 2005: Robot Soccer World Cup IX* / J Kok, N Vlassis. – 2006.

9. Sutton, R.S. *Reinforcement Learning: An Introduction*. MIT Press / R.S. Sutton, A.G. Barto. – 1998.

10. Kabysh, A. Graph Modeling Framework. BrSTU Robotics Wiki (www.robotics.bstu.by/mwiki), (2012), link: http://robotics.bstu.by/mwiki/index.php?title=Библиотека_моделирования_на_графах (in Russian).

11. Golovko, V.A. Collective Behavior in Multiagent Systems Based on Reinforcement Learning / V.A. Golovko, A.S. Kabysh // *Proceedings of the Tenth International Conference «Pattern recognition and image processing»* (PRIP–2009), Minsk, Belarus, (19–21 may 2008) – Minsk, 2009. – P. 260-264

12. Kabysh, A. Influence Learning for Multi-Agent System Based on Reinforcement Learning / A. Kabysh, V. Golovko, A. Lipnickas // *Journal of Computing*. – 2012. – (11), 1 – P. 39-44.