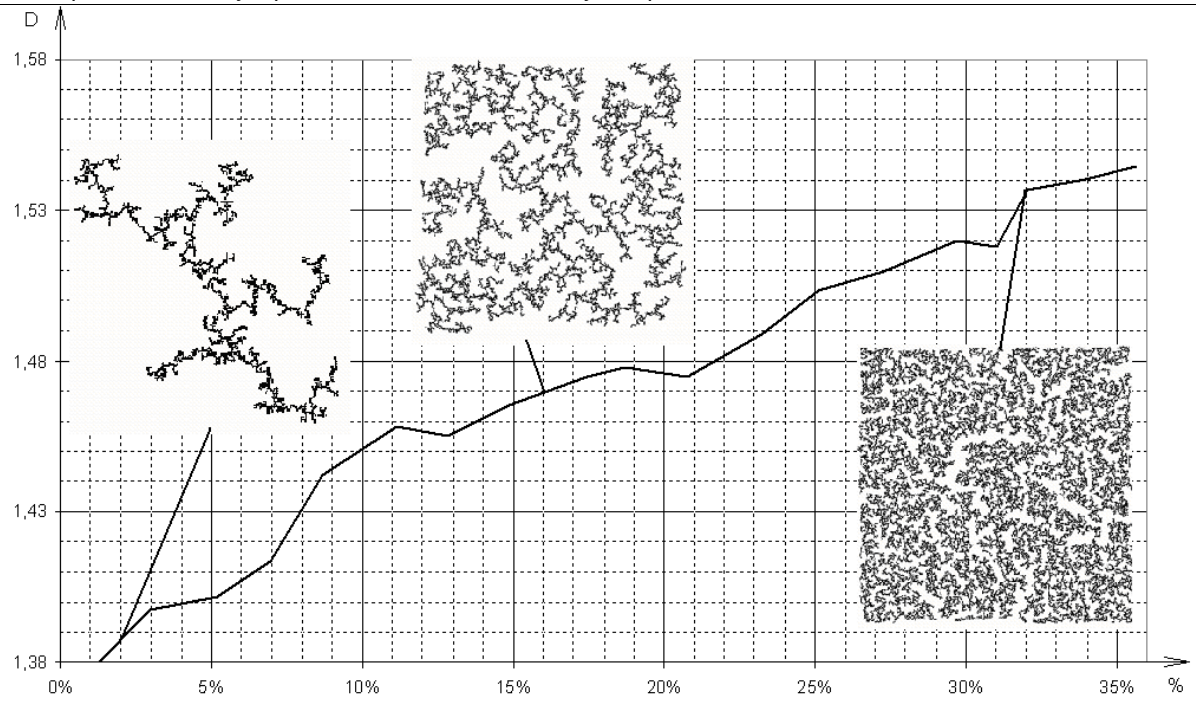


Рис. 10. Фрактальные характеристики кластер-кластерных агрегатов различной плотности

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Суздалев, И.П. Нанотехнология: физико-химия нанокластеров, наноструктур и наноматериалов / И.П. Суздалев – М.: КомКнига, 2006. – 592 с.
2. Кулак, М.И. Фрактальная механика материалов / М.И. Кулак. – Минск: Выш. шк., 2002. – 304 с.
3. Золотухин, И.В. Твердотельные фрактальные структуры / И.В. Золотухин // Международный научный журнал "АЭЭ". – 2005. – № 9. – С. 56–66.
4. Золотухин, И.В. Фрактальная структура фуллерита / И.В. Золотухин, Л.И. Янченко, Е.К. Белоногов // Письма в ЖЭТФ – 1998. – № 9. – С. 684–685.
5. Смирнов, Б.М. Физика фрактальных кластеров / Б.М. Смирнов. – М.: Наука, 1991. – 134 с.
6. Волков, Е.Г. Фрактальная размерность бидисперсных кластеров / Е.Г. Волков // Сб. конкурсных и научных работ студентов и магистрантов / БрГТУ; редкол.: В.В. Тур [и др.]. – Брест, 2006. – С. 97–99.
7. Волков, Е.Г. Моделирование фрактальных кластеров с изменением коэффициента длины пробега / Е.Г. Волков // Современные проблемы математики и вычислительной техники: матер. V республиканской научной конференции молодых ученых и студентов, 28-30 ноября 2007 г. / БрГТУ; редкол.: В.В. Тур [и др.]. – Брест, 2007. – С. 51–53.
8. Конвей, Дж. Упаковки шаров, решётки и группы: в 2-х т. / Дж. Конвей, Н. Слоэн. – Т.2. – М.: Мир, 1990. – 376 с.

Материал поступил в редакцию23.09.09

VOLKOV E.G., DERECHENNIK S.S. Features of calculation fractal dimensions of cluster-cluster agregates

The internal structure cluster-clusters fractal agregates presented by models of multipartial disperse systems is researched. The range of change of the size covering cluster cells necessary for calculation hausdorff dimension in which the internal structure aggregate is most correctly estimated is certain. Palculation fractal dimensions in this scale range is tolerant neither to the size of cluster, nor to size of modelling area. Thus fractal dimension depends only on cluster density and increases from value 1,38 up to 1,54 at change of concentration of particles in a range 1 … 36 %. Also it has been found out, that cluster-clusters agregates with density above 36 % cannot refer to fractals.

УДК 004.8.032.20

Кабыш А.С., Головко В.А.

НЕКОТОРЫЕ ПОДХОДЫ К МНОГОАГЕНТНОМУ ПОДКРЕПЛЯЮЩЕМУ ОБУЧЕНИЮ

Многоагентное подкрепляющее обучение. Подкрепляющее обучение (Reinforcement Learning, RL) – это область искусственного интеллекта и теория машинного обучения, предназначенная для обучения автономных агентов через их взаимодействие с внешней средой для достижения в ней оптимального поведения [1]. Подкрепляющее обучение возникло на пересечении таких областей наук как динамическое программирование, машинное обучение, исследование рефлексов, когнитивные процессы [1–3].

Идеи подкрепляющего обучения первоначальным образом возникли в опытах по обучению животных, при построении теории рефлексов, как модель ответов на стимул. Биологический взгляд на подкрепляющее обучение дан в [3]. Позже был разработан математический аппарат RL теории [1], который является комбинацией идей Монте-Карло и Динамического программирования и основан на итеративном варианте формулы Беллмана.

***Кабыш А.С.**, аспирант кафедры интеллектуальных информационных технологий Брестского государственного технического университета.*

Беларусь, БрГТУ, 224017, г. Брест, ул. Московская, 267.

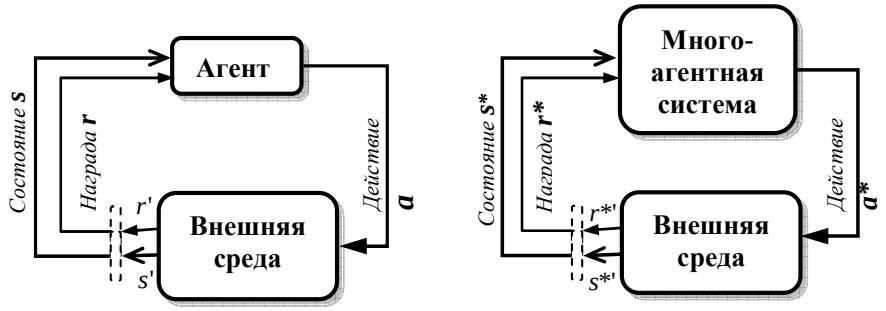


Рис. 1. Стандартная (слева) и объединенная (справа) модели RL

Коллективное (многоагентное) обучение необходимо для формирования желаемого *коллективного* поведения многоагентной системы. Теория многоагентных систем изучает поведение системы, состоящей из взаимодействующих агентов с сильной внутренней связью и большим числом степеней свободы [4]. Так, известно [5], что коллективное поведение взаимодействующих, и даже мешающих друг другу агентов приводит в целом к многоагентной системе с совершенно экзотическими (эмерджентными), неожиданными свойствами иногда вообще не реализуемых с точки зрения отдельных агентов. Поведение такой системы принципиально невозможно разделить на поведение её составляющих; так и обучение такой системы невозможно разделить на раздельное обучение входящих в многоагентную систему агентов.

Во многих источниках [6–11] многоагентное подкрепляющее обучение рассматривается только в контексте теории игр и используется для нахождения точки равенства Нэша для группы игроков (агентов). Работы [1, 6] является обобщающими с этой точки зрения, но в [1] и [12] указывается, что многоагентное обучение, как свойство, присущее сложным системам, все еще остается открытым вопросом.

В предыдущих работах по многоагентному подкрепляющему обучению [13–14] был использован подход, при котором вся Многоагентная система рассматривалась как один агент – *модель объединенного обучения*. С агентов собиралось суммарное действие a^*, на которое возвращалась суммарная награда r^* и суммарное состояние s^*. Каждый агент получал своё, личное видение общего состояния и награды.

Объединенная модель позволяет осуществить успешное обучение системы агентов, но ничем, в сущности, **не отличается** от стандартной RL модели. Ей свойственны все недостатки, имеющиеся у стандартной модели, а именно[11] – «проклятие размерности», сложность обучения и ориентация модели обучения на одного агента.

Проклятие размерности - наследие от динамического программирования, обозначающее, что длительность обучения растет пропорционально размеру пространства состояний-действий. В больших пространствах алгоритм может и не сойтись. В задаче кооперативного движения группы агентов [17–18] размерность пространства состояний-действий была равна количеству агентов. В связи с этим сходимость алгоритма была медленной; в некоторых случаях, алгоритм не смог достичь заданной величины ошибки.

Сложный процесс обучения и долгая сходимость алгоритма при использовании функционального аппроксиматора делают обучение трудным процессом. Эффективные алгоритмы функциональной аппроксимакции все еще находятся в стадии разработки. Большую роль начинает играть опыт использования НС в RL обучении [16, 17].

Стандартная модель подкрепляющего обучения предназначена для одиночного обучения. То есть, алгоритм обучения не учитывает распределенность и топологию внутренних связей многоагентной системы.

Многоагентное обучение – достаточно сложная и трудоемкая задача. Возрастающая сложность обучения связана с большим количеством обучаемых агентов и их взаимосвязью друг с другом. Так, в [11] отмечают, что пространство состояний-действий экспоненциально растет в соответствии с числом обучаемых агентов и необходимо использовать обобщение (н.с.), чтобы решить эту проблему.

С учетом существующих недостатков, возникла потребность в новой версии многоагентного подкрепляющего обучения, которая решала бы описанные проблемы.

Подход к обучению на основе декомпозиции. В подкрепляющем обучении всегда имеется некоторая внешняя среда, на которой нужно построить *оптимальную политику поведения*. Аппроксимация такой политики, в особенности для нескольких агентов, может стать очень трудоемкой задачей ввиду сложности пространства, и не факт, что политика будет построена с заданной точностью. Подход к подкрепляющему обучению на основе декомпозиции указывает на то, каким образом исходную политику можно разбить на множество взаимодействующих подполитик и тем самым редуцировать сложность среды. За аппроксимацию каждой политики отвечает отдельный агент, а все агенты объединены в одну систему, глобальное поведение которой в итоге дает ожидаемую оптимальную политику.

В подкрепляющем обучении направляющим фактором агента является награда; она задает некоторую цель и корректирует процесс её достижения. Следовательно, *выделив награды, можно выделить и цели*, т.е. направление построения политики. Отсюда можно сформулировать принцип декомпозиции – *главным критерием разделения на политики на подполитики является разделение фазового пространства наград на непересекающиеся области*. Иными словами, разделение на подполитики происходит по принципу «за что начисляется награда?». Начисления награды должны зависеть от структуры многоагентной системы, и тогда цели агента будут связаны с целями многоагентной системы.

Рис. 2 иллюстрирует использование принципа декомпозиции. Квадратом, символически, представлена среда. В классическом подходе, одна политика P аппроксимируется на всю среду, в связи, с чем возможны компромиссы, либо недоисследование среды (неохваченные политикой области среды). С учетом декомпозиции, одна политика разбивается на несколько подполитик по какому-либо признаку, и аппроксимация каждой политики поручается собственному агенту. Критерии разбиения на подполитики могут быть различными – разбиение по состояниям, по действиям, по наградам, по состояниям-действиям.

Принцип декомпозиции позволяет разбить задачу обучения на подзадачи, чем во многом решает проблему проклятья размерности. Аппроксимация выделенных в процессе декомпозиции политик происходит значительно быстрее, что решает проблему сложности обучения. Принцип декомпозиции разработан самостоятельно, но похожие механизмы используются в иерархическом подкрепляющем обучении [18] и методе масштабирования среды, называемом RL with Macro Actions [19].

Подход на основе относительного подкрепляющего обучения. Тесная взаимосвязь агентов, выраженная в направленности на решение общей задачи, делает их обучение взаимосвязанным. Так, в методе обратного распространения ошибки, ошибка нейронов последующего слоя включена в формулу корректировки весовых коэффициентов нейронов текущего слоя; отметим, что данное решение является принципиальным для обучения многослойных нейронных сетей. Аналогичная логика была применена и к подкрепляющему обучению. В многоагентной системе действия агентов могут быть направлены на других агентов, не только во внешнюю среду.

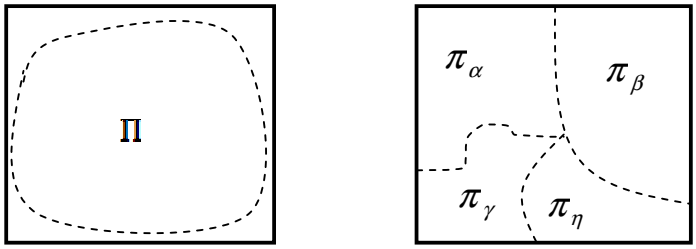
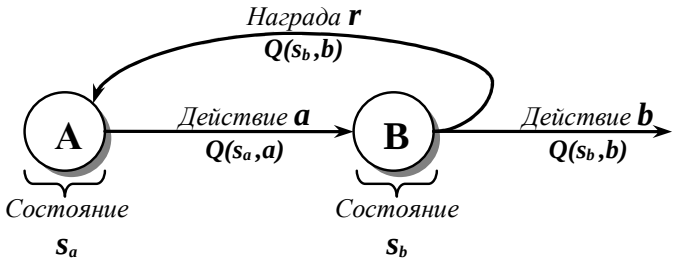


Рис. 2. Наглядное представление различий между обычным (слева) и проекционным подходом (справа) к подкрепляющему обучению



$$d_{AB} = \underbrace{r + gQ(s_b,b)}_{\text{Обратная связь от последующего агента}} - Q(s_a,a) \quad (1)$$

$$Q(s_a,a) = Q(s_a,a) + a\,d_{AB} \quad (2)$$

Рис. 3. Относительное обучение в многоагентной системе

Рассмотрим ситуацию в многоагентной системе, показанную на рис. 3.

Агент A совершает действие a с ценностью $Q(s_a,a)$ над другим агентом B и переводит его в новое состояние s_b. Второй агент также совершает действие b (над другим агентом, или над средой) и сообщает первому награду, отражающую полезность полученного воздействия и ценность совершенного действия – r, $Q(s_b,b)$. Таким образом, можно определить ошибку временной разности d_{AB} одного агента, относительно другого (1).

Обратная связь между агентами делает возможным обучение агентов относительно друг друга по стандартному правилу обучения (2). Формула (1) легко модифицируется в случай произвольного количества агентов. Таким образом, относительность обучения позволяет локально использовать стандартную RL модель в многоагентной системе.

Важно отметить, что принципы декомпозиции и относительности обучения прекрасно дополняют друг друга и их можно использовать одновременно. Введенные принципы изменяют стандартную модель подкрепляющего обучения с «один агент» на «один агент внутри многоагентной системы», в предыдущих работах, в модели объединенного обучения, использовался принцип «все агенты».

Модель эксперимента и результаты. Введенные принципы были опробованы на задаче управления многозвенным роботом [15]. Многозвенный робот (рис. 4, 5) – это робот с N степенями свободы, где N – количество узлов, действующих в клеточной среде (100x100). Каждый узел робота представляет собой одного интеллектуального агента. Каждый узел робота, кроме последнего, может изменять положение, ориентацию всех последующих узлов относительно своей позиции на 360 градусов. Центральный (корневой) узел робота не изменяет своего местоположения. Последний, терминальный узел не совершает действий; положение терминального узла зависит от согласованных действий предыдущих агентов. Цель робота – достичь терминальным узлом заданной точки в пространстве. *При обучении необходимо научиться согласовывать движение частей робота таким образом, чтобы терминальный элемент попадал в целевую точку*. После обучения многозвенный робот должен уметь самостоятельно достигать любой доступной цели во внешней среде.

Целью каждого агента является продвижение остальной части робота ближе к цели. Выделив цели, выделяем и подобласти награды, начисляемые каждому агенту. Теперь, для каждого сегмента, пара состояние-действие, имеет свое собственное значение награды. В противном случае, без декомпозиции, для каждой награды мы бы имели N-мерную точку в пространстве состояний действий, где N – количество сегментов робота. Подобная ситуация сохранялась в модели объединенного обучения, что делало сходимость в этой модели очень долгим процессом.

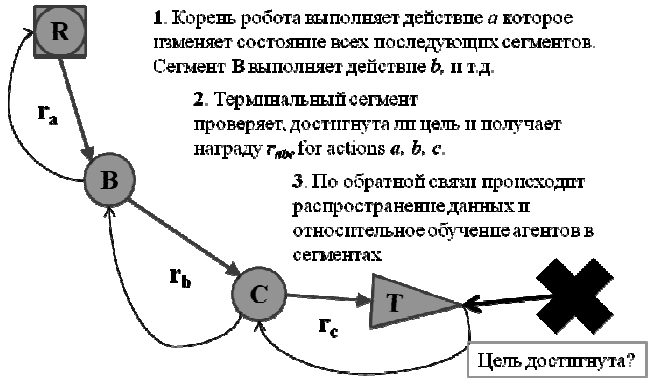


Рис. 4. Модель многозвенного робота из 3-х сегментов и алгоритм его работы

В задаче обучения пятизвенного робота (рис. 5) пятимерное пространство состояний действий было преобразовано в пять одномерных, обучение в которых значительно быстрее. Таким образом, решаются проблемы проклятия размерности и сложности обучения.

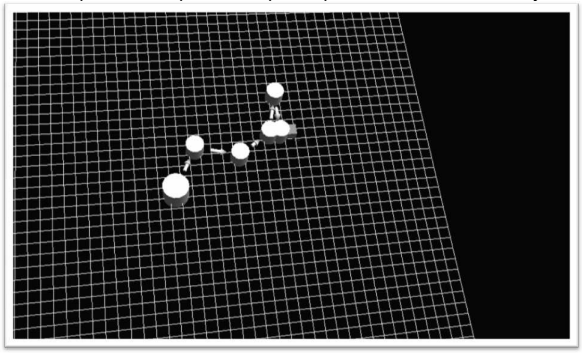


Рис. 5. Моделирование пятисегментного многозвенного робота. Робот достигает цели

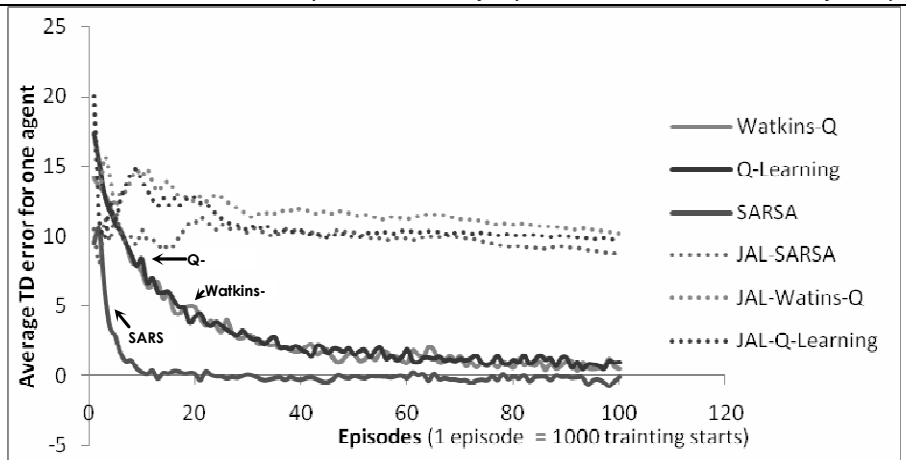


Рис. 6. График среднего значения ошибки временной разности для одного агента показывает эффективность различных алгоритмов подкрепляющего обучения для представленной задачи. На графике алгоритмы подкрепляющего обучения, построенные с учетом введенных принципов, сравниваются с алгоритмами-аналогами, построенными на модели объединенного обучения (графики обозначены пунктиром, а в легенде указаны с приставкой JAL)

Проведенное моделирование (рис. 6) описанной задачи показало следующие результаты.

Полученные результаты сводятся к следующим положениям:

1. Сходимость обучения была на порядок быстрее, чем при объединенном обучении (60-100 эпизодов против 1000).
2. При моделирования использовались следующие параметры
   a. $\lambda = 0.7 \sim 1$;
   b. $\alpha = 0.05 \sim 0.1$;
   c. $\gamma = 0.7$.
3. Была выявлена способность многоагентной системы к синхронизации действий после обучения.
4. Наблюдалось как обучение, так и не обучение. Обучение выражалось в разных стратегиях достижения цели. Необучение выражалось в сильном группировании робота и невозможности провести целенаправленную синхронизацию действий (начало робота мешало его концу достичь цели, и наоборот). Процент необучения не превышет 10% случаев.
5. Сложность структуры робота напрямую влияет на качество обучения. Если робот имеет 7-10 сегментов, вероятность обучения значительно снижается. Действия робота в начале и в конце не синхронизированы. Необходимо вводить еще один уровень организации обучения (например, иерархическое обучение [19]).
6. Стратегия поведения робота значительно изменялась в зависимости от выбранного алгоритма обучения. Существенное влияние на качество синхронизации и на внешний вид поведения оказывало применение механизма следов преемственности в алгоритме обучения. Алгоритмы со следами преемственности (SARSA, Watkins-Q) показывали в целом более гладкое поведение, чем алгоритмы без них (Q-Learning).
7. В пятых, моделирование показало наиболее оптимальное поведение робота для данной задачи, которое значительно отличалось от ожидаемого. Робот, обучаемый по SARSA алгоритму, предпочитал постоянное вращение с перестройкой структуры во время вращения. Тогда как ожидаемым поведением было бы прямое достижение цели. Подобное, прямое достижение цели, показывал алгоритм Q-Learning.
8. Не регулируется, каким образом робот достигает цели. Например, в будущих экспериментах награда может начисляться за «красоту» или скорость достижения цели, а не только за сам факт.

Заключение. В данной работе представлены новые подходы к многоагентному подкрепляющему обучению. Данные подходы были разработаны как решение проблем проклятия размерности и сложности обучения, возникающих в задачах подкрепляющего обучения. Подход на основе декомпозиции решает проблему проклятия размерности и сложности обучения. Относительность обучения позволяет агентам обучаться относительно друг друга. Была разработана модель многоагентной системы эмулирующая многозвенного робота, и проведен успешный эксперимент по её обучению. Будущие направления работы сосредоточены на разработке новых принципов обучения с акцентом на иерархическое обучение.

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Richard S. Sutton, Andrew G. Barto. Reinforcement Learning: An Introduction // Pambridge : MIT Press., 1998.
2. Dr. Florentin Woergoetter, Dr. Bernd Porr . Статья *Reinforcement Learning* на ресурсе (http://www.scholarpedia.org /article/Reinforcement_learning).
3. Worgotter, F. and Porr, B. Temporal sequence learning, prediction and control - A review of different models and their relation to biological mechanisms. // Neural Pomp. (2005) 17: 245–319.
4. Hose M. Vidal. Fundamentals of Multiagent Systems with Net Logo Examples. Открытое электронное издание (www.multiagent.com).
5. Liviu Panait, Sean Luke. Pooperative Multi-Agent Learning: The State of Art. // Autonomous Agents and Multi-Agent Systems, Volume 11, 2005. – P. 387–434.
6. Yoav Shoham, Rob Powers, and Trond Grenager. If multi-agent learning is the answer, what is the question?
7. Game Bruno Bouzy, Marc M´etivier. Multi-Agent Model-Based Reinforcement Learning Experiments in the Pursuit Evasion Game. // Artificial Intelligence, Volume 171 , May 2007, P. 365–377.
8. Avraham Bab, Ronen I. Brafman . Multi-Agent Reinforcement Learning in Pommon Interest and Fixed Sum Stochastic Games: An Experimental Study. // Journal of Machine Learning Research 9 (2008) 2635–2675.
9. Spiros Kapetanakis, Daniel Kudenko. Reinforcement learning of coordination in cooperative multi-agent systems. // Proceedings of the Third International Joint Ponference on Autonomous Agents and Multiagent Systems – Volume 3. – P. 1258–1259.
10. Ivo Parashkevov. Joint Action Learners in Pompetitive Stochastic Games. // Thesis for Master of Science degree. Harvard Pollege.
11. Ming Tan. Multi Agent Reinforcement Learning Independent vs Pooperative Agents. // Autonomous *Agents* and *Multi-Agent* Systems, v.10 n.3, , 2005. – P. 273–328.
12. Peter Stone. Multiagent learning is not the answer. It is a question // Artificial Intelligence, 171: 402 – 405, May 2007.
13. Кабыш, А.С. Коллективное поведение аниматов на основе подкрепляющего обучения / А.С. Кабыш, В.А. Головко // Нейроинформатика – 2009. – Часть 1. – С. 191–200.
14. Kabysh A.S., Golovko V.A. «Pollective Behavior in Multiagent Systems Based on Reinforcement Learning» / // PRIP -2009: Proceedings of the Tenth International Ponference (19–21 May, Minsk, Republic of Belarus), 2009., P. 260–264.

15. Kabysh A.S. «Pollective Behavior in Multi-Agent Systems» // OWD 2009 Ph.D. workshop, Eastern Europe Summer School, 12-24 October, Selesian University of Technology, Poland, P. 92–97.
16. Tesauro, G. J. (1994) / TD-gammon, a self-teaching backgammon program, achieves master-level play. *Neural Computation*,6(2):215-219. (http://www.research.ibm.com/massive/tdl.html).
17. Markus Schneider. Reinforcement Learning with RBF-Networks // Scientific Project, University of Applied Sciences Weingarten.
18. Matthew M. Botvinick, Yael Niv, Andrew P. Barto. Hierarchically organized behavior and its neural foundations: A reinforcement learning perspective. // ELSEVIER, 2008 r., Elsevier "Pognition".
19. McGovern, A., and Sutton, R.S. (1998). Macro-actions in reinforcement learning: An empirical analysis. // Technical Report 98-70, University of Massachusetts, Department of Pomputer Science.

Материал поступил в редакцию 19.11.09

***KABYSH A.S., GOLOVKO A.V.* Some approaches to multiagent reinforcement learning**

In this paper introduced research result in area of multiagent reinforcement learning. Purposed two new approaches to collective reinforcement learning. Decomposition approach describes how to split learning task into number of subtasks, where every of them delegated to some agent. Relativity learning approach describes how to learn together two or more agents via reinforcement learning algorithm. Using this principles constructed model of multiagent system emulated multijointed robot. In learning experiment this multiagent system was successful learned. Efficiency of introduced approaches was shown experimentally.

УДК 004.5:004.822

Колб Д.Г.

СРЕДСТВА ПРОСМОТРА БАЗ ЗНАНИЙ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

Введение. Поиск и использование нужной информации становится все более сложным, трудоемким и неэффективным, несмотря на огромные усилия (как научно-технические, так и организационно-финансовые) по увеличению эффективности доступа и обработки уже существующей и постоянно появляющейся новой информации [1]. Развитие глобальной сети интернет привело к появлению в компьютерных сетях огромного разнообразия видов информационных конструкций.

Этот факт говорит о том, что у пользователей глобальных сетей появляется потребность работы с различными видами информационных конструкций. Традиционные методы решения этой проблемы не приводят к качественным результатам. Поэтому становится очевидным, что использование методов и средств искусственного интеллекта для обработки всего многообразия информационных конструкций, чтобы обеспечить возможность удобной работы пользователя. Решение таких задач традиционно являлось одним из направлений исследований в области пользовательских интерфейсов программного обеспечения.

Научная идея предлагаемого подхода состоит в рассмотрении информационных объектов и объектов управления пользовательского интерфейса как элементов базы знаний (БЗ). Такой подход является принципиально новым и позволяет рассматривать пользовательский интерфейс как специализированную интеллектуальную систему, решающую задачу организации диалога человека и системы, обеспечивающей решение основных задач программного средства.

Для представления знаний пользовательского интерфейса предлагается использовать однородные семантические сети с базовой теоретико-множественной интерпретацией. Основным способом кодирования информации для таких сетей является SP (Semantic Pode)-код [2]. Интеллектуальные системы, построенные с использованием SP-кода, называются sc-системами.

Архитектура пользовательского интерфейса интеллектуальных систем. В соответствии с классами решаемых интерфейсом задач можно выделить следующие классы интерфейсных подсистем:

- просмотрщики информационных конструкций внешних языков, среди которых можно выделить просмотрщики информационных конструкций с временной составляющей или проигрыватели (например, видеопроигрыватели или аудиопроигрыватели) и просмотрщики информационных конструкций без временной составляющей (например, просмотрщики традиционных текстов);
- редакторы внешних информационных конструкций для различных способов отображения информации;
- трансляторы информационных конструкций из внешнего представления в SP-код;
- трансляторы информационных конструкций из SP-кода во внешнее представление.

Каждая выделенная интерфейсная подсистема трактуется как специализированная sc-система, имеющая свою БЗ и машину обработки знаний. Пользовательский интерфейс в целом является результатом интеграции всех его подсистем.

Для МОЗ пользовательского интерфейса характерны следующие классы операций:

- рецепторные операции, инициируемые пользователем и приводящие к изменению состояния БЗ sc-системы. К таким операциям относятся операция генерации изображения sc-узла заданного типа (при инициировании её пользователем), операция генерации изображения sc-дуги заданного типа (при инициировании её пользователем), операция трансляции изображения sc-конструкции с указанного sc-окна в семантически эквивалентную ей sc-конструкцию;
- эффекторные операции, отображающие sc-конструкции из БЗ sc-системы пользователю. К таким операциям относятся операция генерации изображения sc-узла заданного типа (при инициировании её системой), операция генерации sc-дуги заданного типа (при инициировании её системой), операция трансляции sc-конструкции в семантически эквивалентное ей изображение sc-конструкции в указанном sc-окне;
- операции «память-память». Условиями применения которых, являются события, происходящие в БЗ sc-системы (например, появление дуги или узла). К таким операциям относятся операция размещения информационной конструкции согласно указанному алгоритму размещения, операция подсветки последней пришедшей в указанное sc-окно информационной конструкции, операции интерпретации правил трансляции с SP-кода на внешний язык.

Язык визуального представления пользовательского интерфейса интеллектуальных систем. Диалог пользователя с интеллектуальной системой при использовании предлагаемого подхода осуществляется посредством обмена фрагментами семантической сети между пользователем и системой. Семантическая сеть, используемая в системе, визуализируется с помощью SPg(Semantic Pode graphical)-кода [2] – унифицированного способа визуализации семантических сетей, закодированных с помощью SP-кода. Такой способ организации диалога является базовым для sc-систем.

Минимальные, но семантически полные средства SPg-кода, обеспечивающие изображение любых конструкций SP-кода, назовем ядром SPg-кода или сокращенно SPg-ядром.

Для более компактной визуализации используются дополнительные визуальные средства SPg-кода, разработанные на основании семанти-

Колб Дмитрий Григорьевич, *ассистент кафедры ИИТ Белорусского государственного университета информатики и радиоэлектроники. Беларусь, БГУиР, 224017, 220013, г. Минск, ул. П. Бровки, 6.*