А.С. КАБЫШ, В.А. ГОЛОВКО
Брестский государственный технический университет, Республика Беларусь
anton.kabysh@gmail.com, gva@bstu.by

КОЛЛЕКТИВНОЕ ПОВЕДЕНИЕ АГЕНТОВ НА ОСНОВЕ ПОДКРЕПЛЯЮЩЕГО ОБУЧЕНИЯ

Исследован многоагентный подход к решению интеллектуальных задач на основе подкрепляющего обучения. Модифицирован алгоритм подкрепляющего обучения для группы агентов, целью обучения является согласованное передвижение агентов в пространстве. В результате обучения сформировались интересные патерны поведения группы: «лидер», «цепочка действий», «группирование».

Введение

В данной работе исследовано коллективное поведение агентов на основе подкрепляющего обучения. Агент определяется как сущность, способная вести себя автономно, воспринимать среду, выполнять план действий и распознавать другие сущности [1, 2]. Среда и множество взаимодействующих агентов в ней образуют многоагентную систему (МАС). Общие подходы для разработки МАС включают стандарты на архитектуру многоагентной системы «Foundation of Intelligent Physical Agent» [2,3] и «Open Agent Architecture», языки коммуникации агентов KQML и FIPA-ACL [3], языки описания онтологий и баз знаний. Схема коммуникации между агентами строится по принципу «запрос-ответ» [4]. Если агенту не хватает информации для принятия решения, он отправляет запрос «кто может мне помочь?» во внешнюю среду. Если ответов больше одного, чаще всего они фильтруются, либо выбирается ответ от наиболее авторитетного источника.

Теория подкрепляющего обучения

При использовании обучения с подкреплением [5] в качестве учителя выступает среда, в которой находится агент (рис. 1). Время предполагается дискретным: $t = 1,2,...,\infty$. В текущей ситуации $s(t)$ агент выполняет действие $a(t)$. На следующем шаге $t = t + 1$ он получает подкрепление $r(t) = r(s(t), a(t))$ за совершенное им на предыдущем шаге действие.

Подкрепление может быть положительным (награда), $r(t) > 0$, или отрицательным (наказание), $r(t) < 0$. Цель агента – максимизировать сум-

марную награду, которую можно получить в будущем в течение длительного периода времени.

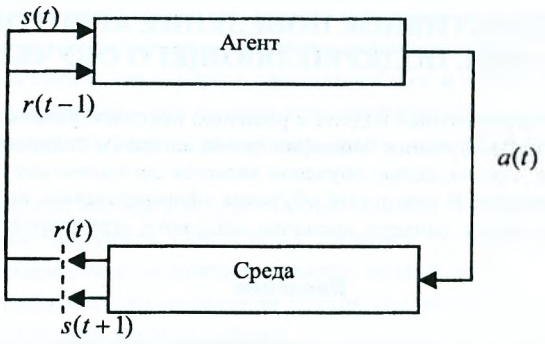

Рис. 1. Схема обучения с подкреплением

Для формализации процесса поиска оптимальной политики π^* поведения агента вводится функция ценности (value function) $V^{\pi}(s(t))$, представляющая суммарную награду, которую получит агент, начиная с состояния $s(t)$, руководствуясь политикой π. Оценка величины V^{π} определяется с учетом дисконтного фактора γ ($0 < \gamma < 1$):

$$V^{\pi} = E\{\sum_{i=t}^{\infty} \gamma^{i-t} r(i)\}. \qquad (1)$$

Дисконтный фактор γ учитывает, что чем дальше агент «заглядывает» в будущее, тем меньше у него уверенности в оценке награды $r(t)$.

Оценку действия *a* в ситуации *s* определяет Q-функция:

$$Q^{\pi}(s_{(t)}, a_{(t)}) = E\{r(t) + \gamma V^{\pi}(s(t+1))\}; \qquad (2)$$

где $V^{\pi}(s(t+1))$ – оценка будущего состояния системы в соответствии с политикой π. Базовый алгоритм подкрепляющего обучения, известный как *Q-learning* использует формулу (2), в которой $V^{\pi}(s(t+1))$ заменен на $\max Q(s_{(t+1)}, a_{(t)})$:

$$Q(s_{(t)}, a_{(t)}) = r(t) + \gamma \max Q(s_{(t+1)}, a_{(t)}). \quad (3)$$

Оценки Q-значений хранятся в 2-мерной таблице, строками и столбцами которой являются состояние и действие. При табличном представлении Q-функции и Марковской среде имеется доказательство сходимости алгоритма Q- learning.

Метод SARSA

Саттон и Барто выделили три семейства алгоритмов обучения с подкреплением: обучение по правилу временных разностей, метод динамического программирования и метод Монте-Карло [8]. Метод временных разностей (temporal difference, TD) основан на пересчете значения ценности текущего состояния на основе ценности последующего состояния. В задачах обучения с подкреплением требуется прогнозировать последовательный ряд значений функции ценности $V(t), V(t+1)....$ по формуле (1). Изменения значений $V(t)$ происходят следующим образом.

$$V(t+1) = V(t) + \alpha\delta(t), \quad (4)$$

где α – коэффициент прогноза; $\delta(t)$ – разность между той оценкой суммарной величины награды, которая формируется у агента для момента времени t после выбора действия $a(t)$ в следующей ситуации $s(t+1)$ в момент времени $t+1$, и предыдущей оценкой этой же величины, которая была у агента в момент времени t.

Значение ошибки $\delta(t)$ находится следующим образом:

$$\delta(t) = r(t) + \gamma V(t+1) - V(t). \quad (5)$$

Для функции Q ошибку $\delta(t)$ естественно определить по формуле [1,5]:

$$\delta(t) = r_{(t)} + \gamma Q(s_{(t+1)}, a_{(t+1)}) - Q(s_{(t)}, a_{(t)}). \quad (6)$$

В соответствии с ошибкой $\delta(t)$ агент обучается. В каждый такт времени происходит как выбор действия, так и обучение агента. Выбор действия происходит так [6]:

момент t с вероятностью $1-\varepsilon$ выбирается действие, соответствующее максимальному значению $Q(s_{(t)}, a_{(t)})$;

с вероятностью ε выбирается произвольное действие случайным образом, $0 < \varepsilon << 1$ Такую схему выбора действия называют «ε-жадным правилом».

Обучение происходит путем градиентного спуска:

$$Q(s_{(t+1)}, a_{(t+1)}) = Q(s_{(t)}, a_{(t)}) - \alpha \frac{\partial \delta^2}{\partial Q(s_{(t)}, a_{(t)})}. \quad (7)$$

Находя частные производные в уравнении (7), получаем [4]:

$$Q(s_{(t+1)}, a_{(t+1)}) = Q(s_{(t)}, a_{(t)}) + \alpha\delta;$$

$$Q(s_{(t+1)}, a_{(t+1)}) = Q(s_{(t)}, a_{(t)}) + \alpha[r_{(t)} + \gamma Q(s_{(t+1)}, a_{(t+1)}) - Q(s_{(t)}, a_{(t)})]. \quad (8)$$

Данная модификация алгоритма Q-Learning, называется SARSA [6] (State-Action-Reward-State-Action), или модифицированный Q-Learning [7, 8].

Подкрепляющее обучение в нейронных сетях

При большом числе состояний и/или действий целесообразно использовать аппроксимацию Q-значений с помощью нейронных сетей [7, 8]. При этом на входы сети подаются состояния $s(t) = \{x_1, x_2, ...x_n\}$, а выходными данными являются оценки Q-значений (рис. 2). Веса нейронной сети корректируются методом градиентного спуска.

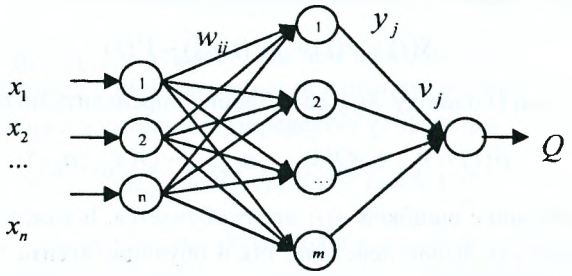

Рис. 2. Кодирование нейронной сетью пар состояние-действие

Распространение сигнала по сети происходит следующим образом:

$$Q = \sum_j y_j v_j - T, \quad y_j = F(s_j), \qquad s_j = \sum_j w_{ij} x_i - T_j .$$

Коррекция весов осуществляется по формулам:

$$v_j(t+1) = v_j(t) + \alpha\delta(t)e_1(t); \tag{8}$$

$$w_{ij}(t+1) = w_{ij}(t) + \alpha\delta(t)e_2(t); \tag{9}$$

где $e_1(t), e_2(t)$ – значения *следов преемственности* для выходного и скрытого слоев соответственно. *Преемственность e обозначает, насколько сильные изменения веса необходимо произвести в ответ на текущую ошибку* [8]. Использование «следов преемственности» позволяет при обновлении весов учитывать ошибку на предыдущих шагах, так как они хранят взвешенную сумму выходных градиентов.

Значения $e_1(t), e_2(t)$ рассчитываются по формулам:

$$e_1(t+1) = (\lambda\gamma)e_1(t) + \frac{\partial Q}{\partial v_j}, \quad \frac{\partial Q}{\partial v_j} = y_j, \quad e_1(t+1) = (\lambda\gamma)e_1(t) + y_j . \tag{10}$$

$$e_2(t+1) = (\lambda\gamma)e_2(t) + \frac{\partial Q}{\partial w_{ij}}, \quad \frac{\partial Q}{\partial w_{ij}} = \frac{\partial Q}{\partial y_j}\frac{\partial y_j}{\partial S_j}\frac{\partial S_j}{\partial w_{ij}} = v_j F'(s_j)x_i ,$$

$$e_2(t+1) = (\lambda\gamma)e_2(t) + v_j F'(s_j)x_i , \tag{11}$$

где λ $(0 \le \lambda \le 1)$ – параметр, экспоненциально уменьшающий вклад поздних ошибок временной разности.

Алгоритм обучения имеет вид:

1. При $t = 0$ положить $e_1(0) = 0, e_2(0) = 0$.

2. Выбрать действие $a(t)$.

3. Выполнить действие $a(t)$ и получить награду $r(t)$.

4. Если $t > 0$ произвести корректировку весов по формулам (8) и (9), учитывая (10) и (11).

5. Если условие останова достигнуто, то конец. Иначе $t = t + 1$, переход к шагу 2.

Преимущества использования метода SARSA на основе нейросетей растут пропорционально увеличению сложности среды (увеличению чис-

ла состояний и максимального числа шагов до цели). При решении задач, имеющих большое пространство состояний, рассмотренный метод требует меньшее количество ресурсов и обеспечивает быструю сходимость.

Коллективное подкрепляющее обучение

Вводим понятие группы агентов, образующих МАС и связанных друг с другом локальными связями. МАС реагирует со средой как единый организм, обеспечивая распределение состояний и сбор действий. Суть коллективного подхода – в рассмотрении группы агентов как единого существа с учетом локальной значимости каждого агента [9].

Пусть имеется система N агентов в некоторой среде. В текущий момент времени t система собирает у всех агентов информацию о выбранных ими действиях и формирует «комплексное действие» путем объединения всех действий агентов

$$a^*(t) = \{a_1(t), a_2(t)...., a_N(t)\} = \bigcup_{a \in A} a_i(t), \qquad (12)$$

МАС передает действие в окружающую среду и выполняет его. В ответ МАС ожидает новое состояние глобального среды $s^*(t+1)$ и награду за текущее действие $r(t)$. В следующий момент времени $t = t+1$ на вход системы поступает новое состояние среды $s^*(t)$ и подкрепление за действие, выполненное на прошлом шаге. Награда передается обратно к каждому агенту, который использует эту информацию для обучения. Состояние среды $s^*(t)$ распределяется по каждому агенту, но воспринимается агентом локально, в соответствии с восприятием его сенсоров (фильтров) f_i. Формально, данная процедура аналогична пересечению множеств (13), где $s^*(t)$ – множество всех состояний среды, а f_i – множество состояний, доступных агенту, и задача состоит в том, что бы найти для них общие состояния.

$$s_i(t) = s^*(t) \bigcap f_i \,. \qquad (13)$$

Агенты связаны друг с другом локальными связями. Кооперация агентов подразумевает обмен информацией друг с другом. В качестве информации может выступать значение текущего состояния, прошлое выбранное действие. Это позволяет учитывать в обучении агента информацию от соседних агентов и тем самым учитывать в обучении взаимосвязи между

агентами. Тем самым, обучая агентов индивидуально, мы обучаем всю систему как одно существо.

Алгоритм коллективного обучения состоит в следующем.

1. MAC получает описание $s^*(t)$ начального состояния окружающей среды, $t = 0$.

2. MAC передает $s^*(t)$ состояния каждому входящему в неё агенту.

3. Каждый агент воспринимает полученное состояние в ограниченных рамках, в соответствии со своими сенсорами f_i.

4. Каждый агент определяет действие, которое он собирается выполнить $a_i(t)$.

5. MAC собирает все индивидуальные действия в одно комплексное действие $a^*(t)$ и исполняет его в окружающей среде.

6. Окружающая среда на основании комплексного действия вычисляет новое состояние $s^*(t+1)$ и пересылает его MAC вместе с подкреплением $r(t)$.

7. MAC информирует всех агентов о получаемом подкреплении, после чего агенты могут обучиться.

8. $t = t + 1$ Переход на шаг 2.

Результаты экспериментов

Модель эксперимента имеет следующий вид. Внешней средой для агентов служит двумерная сетка с периодическими границами. Размеры сетки составляют 100*20. При инициализации модели на сетке случайным образом размещаются $N = 5$ агентов. Агент, находящийся правее другого на сетке, является для него ведущим. Агент и его ведущий связаны локальными кооперативными связями. Если два агента оказались на одной линии, прежнее расположение сохраняется.

Таким образом, мы имеем группу агентов, связанных последовательно друг с другом локальными связями. Агенты являются полностью необученными, а веса нейронной сети инициализированы случайным образом.

Функционирование агентов происходит по принципу коллективного подкрепляющего обучения. Каждый такт времени происходит выбор как действия агентом, так и обучение агента. Входами нейронной сети агента являются:

- Текущее положение в сетке – координаты *x*,*y*.
- Последнее совершенное действие локально связанного агента.

- Расстояние в модельных единицах по координатам x, y до ведущего.

Множество действий агента включает перемещение на одну клетку влево, вправо, вверх, вниз. Подкрепление для агентов рассчитывается как сумма разностей координат последовательно связанных агентов:

$$r(t) = \sum_{i=0}^{N} (x_i(t) - x_{i-1}(t) + 5p + abs \mid y_i(t) - y_{i-1}(t) \mid) \cdot \qquad (14)$$

Смысл подкрепления состоит в том, что оно должно быть максимальным тогда, когда агенты сохраняют нужную формацию. Параметр p является компенсирующим параметром, допускающим отклонение по оси x на некоторое значение, т.к. агенты должны двигаться цепочкой и расстояние по оси x между ними всегда будет ненулевым. Значение для разности y-координат всегда должно быть отрицательным.

Моделирование проводилось для сравнения работы двух методов на основе подкрепляющего обучения – Q-Leraning и метода SARSA на основе нейронных сетей. Сравнение динамики подкреплений показано на графиках рис. 3. Q-Learning так и не смог достичь оптимального поведения, возможно ввиду большого пространства состояний-действий. Метод SARSA на основе нейронных сетей показал удовлетворительные результаты при моделировании.

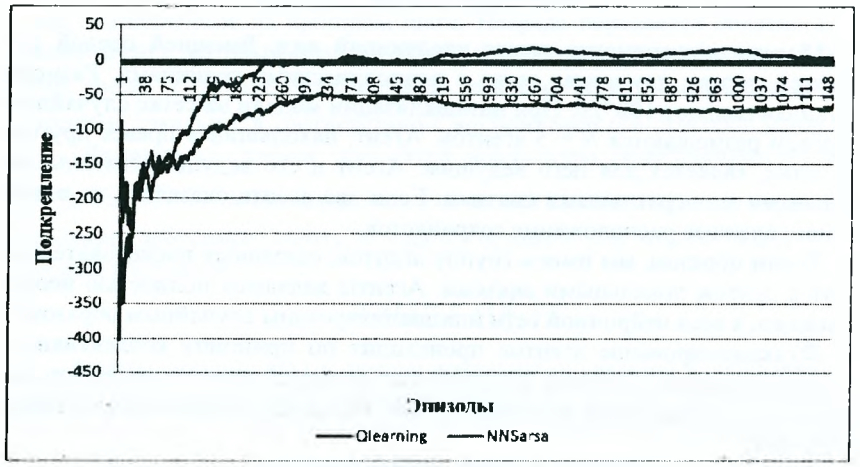

Рис. 3. Результаты эксперимента

В первые такты времени поведение агентов носит хаотичный характер. Однако в дальнейшем наблюдается общая тенденция к самоорганизации и подборке оптимальных параметров месторасположения агента. Это связано с тем, что агент стремится достичь максимума своей функции подкрепления. При достижении стационарного состояния МАС начинает жить как одно существо. Чтобы избежать группирования, в функцию подкрепления вводился «отталкивающий фактор», уменьшающий функцию подкрепления, если агенты были близко друг к другу. Также в модель не было включено действие «стоять». Ведущим агентом всегда выбирается тот, чье положение на сетке является наиболее правым. Это обеспечивает постоянное движение вперед всей модели, но не является оптимальным решением.

Эксперименты проводились со следующими значениями параметров: $\gamma = 0.1$, $\varepsilon = 0.8$, $\lambda = 0.1$.

В процессе наблюдения за коллективом агентов были выделены некоторые паттерны поведения, свойственные данной многоагентной системе. К ним относятся «Лидер», «Группирование», «Цепочка действий».

Паттерн «Лидер» имеет вид, аналогичный рис. 4. Поскольку в модели особая роль отводится ведущему агенту, его поведение характеризуется свободным движением, в то время как остальные агенты стараются повторять за ним все его действия и находится в максимальной близости к агенту лидеру. В целом, данный паттерн можно характеризовать как нормальное поведение модели.

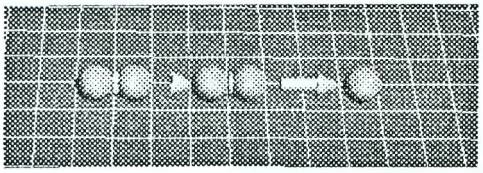

Рис. 4. Скоординированное передвижение

Паттерн «Цепочка действий» (рис. 5) характеризуется как последовательное выполнение агентами действий, совершенных их лидирующими соседями. Например, если ведущий агент совершает поворот налево, то следующий за ним также совершает поворот налево. Наблюдается последовательное эхо реакций агентов, вслед за ведущим.

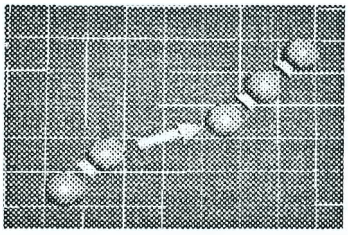

Рис. 5. Эхо реакций на действия главного агента

Большое количество параметров нейронных сетей и алгоритма обучения с подкреплением с одной стороны создает возможности для настройки модели под нужные результаты, а с другой стороны требуется много экспериментов для определения оптимальной конфигурации модели.

Заключение

Представленная модель характеризует подход для моделирования коллективного поведения МАС на основе подкрепляющего обучения. Модификация данного подхода для коллективного поведения позволяет рассматривать группу агентов как единое существо и формировать подкрепление для всей группы на основании их действий. При использовании данного подхода мы наблюдали устойчивую тенденцию к самоорганизации группы агентов. При обучении использовались два метода Q-Learning и нейросетевой SARSA. Первый не обеспечил оптимального поведения. Второй показал хорошие обобщающие способности для модели.

Список литературы

1. Fundamentals of Multiagent Systems. Hose M. Vidal. (http://multiagent.com).
2. FIPA Abstract Architecture Specification (http://www.fipa.org/specs/fipa00001/index.html).
3. FIPA ACL Message Structure Specification. См. (http://www.fipa.org).
4. Programming Multi-Agent Systems in AgentSpeak using Jason. Rafael H. Bordini (University of Durham, UK) Jomi Fred Hubner (University of Blumenau, Brazil) Michael Wooldridge(University of Liverpool, UK).
5. Sutton R., Barto A. Reinforcement Learning: An Introduction. Cambridge: MIT Press. 1998. (http://www.cs.ualberta.ca/~sutton/book/the-book.html).
6. От моделей поведения к искусственному интеллекту. Серия «Науки об искусственном» (под ред. Редько В.Г.). М.: УРСС. 2006.
7. Использование нейронных сетей в алгоритме Q-Learning. Кузьмин В. Институт Транспорта и Связи, Рига, Латвия.

8. Исследование алгоритмов обучения с подкреплением в задачах управления автономным агентом. Кузьмин В., Институт Транспорта и Связи, Рига, Латвия.
9. A Platform for Implementation of Q-Learning Experiment. Reference. Franchesco de Comite.

Н.Ю. ОДИНЦОВА, В.Ю. ПОПОВ
Уральский государственный университет им. А.М. Горького, Екатеринбург
odincova_natalya@mail.ru, Vladimir.Popov@usu.ru

ГЕНОМНЫЕ ПЕРЕСТРОЙКИ И МОДЕЛИ ЭВОЛЮЦИИ

Рассмотрена модель построения эволюционного дерева при помощи нейронных сетей. Для конструирования эволюционного дерева используется многослойная сеть с самоорганизацией на основе конкуренции. Проанализированы различные типы мутаций, которые могут при этом использоваться.

Ю.Р. ЦОЙ
Томский политехнический университет
qai@mail.ru

ОБ ЭВОЛЮЦИОНИРУЮЩИХ СЛОЖНЫХ СЕТЯХ И МОДЕЛИРОВАНИИ ОТКРЫТОЙ ЭВОЛЮЦИИ

В статье рассматривается проблема моделирования открытой эволюции, как процесса, в котором возможен неограниченный рост сложности объектов и взаимодействий между ними. Анализируется возможность применения разработанного самоадаптивного алгоритма эволюции вычислительных регуляторных сетей для моделирования таких процессов.