

DEMIN ALEXANDER

A.P.Ershov Institute of Informatics Systems SB RAS, Novosibirsk

VITYAEV EVGENII

Sobolev Institute of Mathematics SB RAS, Novosibirsk

THE MODEL OF ADAPTIVE BEHAVIOUR BASED ON THE SEMANTIC PROBABILISTIC INFERENCE

Annotation. The model of adaptive behavior of the autonomous adaptive agents (artificial organism) based on the semantic probabilistic inference and functional system theory by P.K.Anokhin is presented in the paper. The control system is built on the hierarchy of functional systems that are forming for the useful purposes achievement. The semantic probabilistic inference used for the development of the optimal ways of purposes achievement. The main advantage of the model is the possibility of the new purposes formation. Based on this model the autonomous adaptive agent was developed. The comparative experiments of the autonomous agent behavior with the other models based on the reinforcement learning and neural networks were performed. The presented results are demonstrate that the adaptive agent is learning and function more efficient.

А. В. ДЕМИН

Институт систем информатики имени А. П. Ершова СО РАН, г. Новосибирск

Е. Е. ВИТЯЕВ

Институт математики СО РАН, г. Новосибирск

E-mail: vityaev@math.nsc.ru

РАЗРАБОТКА МОДЕЛИ АДАПТИВНОГО ПОВЕДЕНИЯ АНИМАТА НА ОСНОВЕ СЕМАНТИЧЕСКОГО ВЕРОЯТНОСТНОГО ВЫВОДА

Аннотация. В работе предложена адаптивная система управления аниматом (искусственным организмом), основанная на семантическом вероятностном выводе и теории функциональных систем П.К.Анохина. Система управления строится на основе иерархии функциональных систем, формирующихся для достижения полезных для анимата целей. Семантический вероятностный вывод используется для нахождения наиболее оптимальных способов достижения цели. Основным отличием данной модели является возможность автоматического выявления новых подцелей. Основываясь на предложенной модели, был построен анимат и проведен ряд экспериментов по его обучению и сравнению с существующими подходами, основанными на нейронных сетях и потактовом обучении (Reinforcement Learning). Результаты сравнения показали, что предложенная модель обучается и действует эффективнее.

1. Введение. В последнее время активно развивается направление исследований «Адаптивное поведение», связанное с изучением фундаментальных принципов, позволяющих естественным или искусственным организмам приспосабливаться к переменной внешней среде. Один из основных подходов этого направления является создание и исследование агентов (компьютерных программ или роботов), поведение которых

основано на принципах поведения живых организмов. Подобные агенты были названы «аниматами» (animal + automat = animat).

В данной работе предложена общая схема адаптивной системы управления аниматом, которая включает в себя архитектуру на основе иерархию функциональных систем и подцелей, алгоритм обучения, использующий семантический вероятностный вывод и возможность автоматического формирования новых подцелей. На основе предложенной модели в виде компьютерной программы был реализован простейший анимат и среда его обитания. При помощи данной программы был поставлен ряд экспериментов по обучению анимата и проведено тестовое сравнение с существующими подходами, основанными на нейронных сетях и потоковом обучении (Reinforcement Learning).

2. Теория функциональных систем. Архитектура предложенной нами системы управления основана на теории функциональных систем, разработанной в 1930-1970 годах известным русским нейрофизиологом П.К. Анохиным [9]. Согласно этой теории единицей деятельности организма является функциональная система, формирующаяся для достижения полезных для организма результатов (например, удовлетворение потребностей). Организация функциональных систем при целенаправленном поведении, осуществляется в соответствии с двумя правилами: последовательностью и иерархией результатов. Последовательность результатов выстраивается по принципу “доминанты”: доминирующая потребность возбуждает доминирующую функциональную систему и строит поведенческий акт, направленный на ее удовлетворение. По отношению к доминирующей функциональной системе все остальные функциональные системы выстраиваются в иерархию по принципу “иерархии результатов”: когда результат деятельности одной функциональной системы входит в качестве компонента в результат деятельности другой.

Центральные механизмы функциональных систем, обеспечивающих целенаправленные поведенческие акты, имеют однотипную архитектуру. Начальную стадию поведенческого акта любой степени сложности составляет *афферентный синтез*, включающий в себя синтез мотивационного возбуждения, памяти и информации об окружающей среде. В результате афферентного синтеза из памяти извлекаются все возможные способы достижения цели в данной ситуации. На стадии *принятия решений* в соответствии с исходной потребностью выбирается только один конкретный способ действий. Для обеспечения достижения результата еще перед началом действий формируется *акцептор результатов действий*, представляющий собой модель параметров ожидаемого результата. Выполнение каждого действия постоянно сопровождается сигнализацией о получении результата, называемой *обратной афферентацией*. Действия по достижению цели продолжаются до тех пор, пока параметры результата действия, поступающие в центральную нервную систему в форме соответствующей обратной афферентации, не будут полностью соответствовать свойствам акцептора результатов действия.

Отдельная ветвь общей теории функциональных систем – теория системогенеза, изучающая закономерности формирования функциональных систем. В данной работе мы также рассмотрим механизм формирования новых функциональных систем на основе выявления подцелей.

3. Модель анимата. Приведем схему работы анимата (рис.1) реализующую схему функциональных систем [1-5]. Будем предполагать, что система управления аниматом функционирует в дискретном времени $t = 0, 1, 2, \dots$. Пусть анимат имеет некоторый набор сенсоров S_1, S_2, \dots, S_n , характеризующих состояние внешней и внутренней среды, и набор возможных действий A_1, A_2, \dots, A_m . Среди множества сенсоров выделим сенсор SA , который представляет информацию о совершенном действии. Считаем, что история деятельности анимата хранится в таблице данных $X = \{X_1, \dots, X_t\}$, где t -я строка таблицы содержит показания

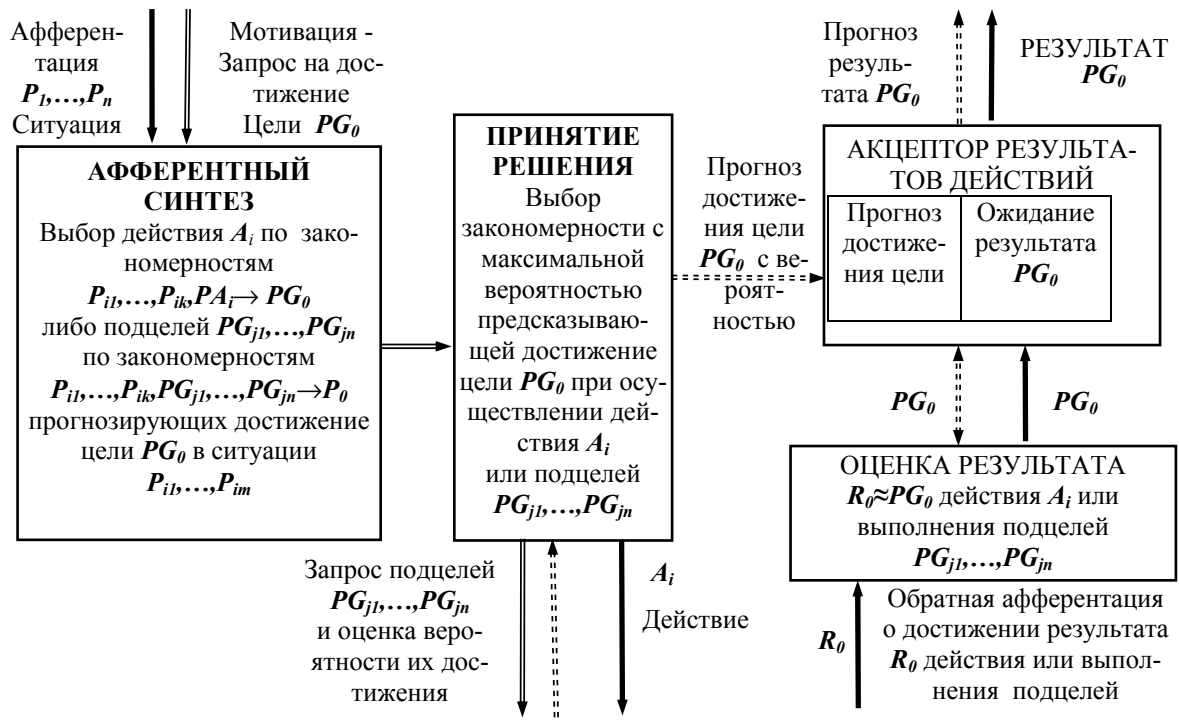


Рис. 1. Схема работы функциональной системы.

сенсоров в момент времени t : $X_t = \{S'_1, S'_2, \dots, S'_n, SA_t\}$, где S'_1, S'_2, \dots, S'_n - значения сенсоров S_1, S_2, \dots, S_n в момент времени t . На множестве X определим множество предикатов $P = \{P_1(t), \dots, P_k(t), PA_1(t), \dots, PA_m(t)\}$, где $P_i(t)$ - сенсорные предикаты, определяющие некоторые условия на показания сенсоров в момент времени t ; $PA_i(t) \Leftrightarrow (SA(t) = A_i)$ - активирующие предикаты, показывающие, что в момент времени t было совершено действие A_i .

Введем понятие *предиката-цели*, $PG(t) = P_{i1}(t) \& P_{i2}(t) \& \dots \& P_{in}(t)$ реализующего условие достижения цели в момент времени t .

Каждой функциональной системе ΦC_j соответствует некоторая цель G_j , достижение которой является задачей данной функциональной системы, и предикат-цель PG_j , характеризующий условие достижения цели.

Каждая функциональная система ΦC_j содержит свой набор предикатов $P_j = P \cup \{PG_{j1}, \dots, PG_{jn}\}$, где PG_{jk} - предикаты-цели, соответствующие целям нижестоящих по иерархии функциональных систем, подчиненных данной функциональной системе. Каждая функциональная система ΦC_j содержит множество закономерностей PR_j вида: $P_{i1}, \dots, P_{ik}, PG_{j1}, \dots, PG_{jn}, PA_i \rightarrow PG_j$. Каждая такая закономерность характеризуется некоторой оценкой p вероятности достижения цели PG_j , при выполнении условия закономерности.

Предположим, что в некоторый момент времени t функциональная система ΦC_j получила запрос на достижение цели PG_j . Тогда из множества закономерностей PR_j извлекаются все закономерности, условие которых выполнено в текущий момент времени t . Если условие закономерности содержит предикаты-подцели PG_{j1}, \dots, PG_{jn} , то функциональная система отправляет запрос на достижение этих подцелей вниз по иерархии функциональных систем. Среди всех отобранных закономерностей выбирается та закономер-

ность, которая с учетом вероятностей выполнения подцелей дает максимальную оценку f вероятности достижения цели. Оценка f закономерности $P_{i1}, \dots, P_{ik}, PG_{j1}, \dots, PG_{jn}, PA_i \rightarrow PG_j$ вычисляется следующим образом: $f(PG_j | PS_{i1}, \dots, PS_{ik}, PG_{j1}, \dots, PG_{jn}, PA_i) = p \cdot f(PG_{j1}) \cdot \dots \cdot f(PG_{jn})$, где p – оценка вероятности данной закономерности, $f(PG_{jk})$ – оценки вероятностей достижения подцелей.

Если все условия выбранной закономерности выполнены, то действие A_i запускается на выполнение. Если множество закономерностей PR_j пусто, либо нет ни одной закономерности, применимой в данной ситуации, то действие выбирается случайно из арсенала имеющихся действий.

После совершения действия обновляются показания сенсоров, оценивается результат действия и уточняется набор правил PR_j (см. ниже).

4. Оценка результатов действий. Каждая функциональная система ΦC_j хранит оценки результатов своих действий $d^j(t)$ для каждого момента времени t . Определим способ оценки результатов действий.

Предположим, что функциональной системе ΦC_j в момент времени t_0 была поставлена задача достичь цель G_j и после достижения цели в момент времени t_1 был получен результат R_j . Тогда оценки результатов действий $d^j(t)$, начиная с момента времени t_0 и до момента времени t_1 будут рассчитаны следующим образом:

зом: $d^j(t) = r \frac{t - t_0}{(t_1 - t_0)}$, $t_0 < t < t_1$, где r – функция оценки качества полученного результата,

$r = \begin{cases} 0, & \text{если } PG_j = 0 \\ \|G - R\|, & \text{если } PG_j = 1 \end{cases}$, где $\|\dots\|$ – мера близости между полученным результатом R_j и поставленной целью G_j .

5. Генерация правил. Для получения множества закономерностей PR_j , которые использует функциональная система ΦC_j , воспользуемся семантическим вероятностным выводом [6].

Семантический вероятностный вывод позволяет находить все закономерности вида $P_{i1}, \dots, P_{in} \rightarrow P_0$ с максимальной вероятностью предсказывающие предикат P_0 . Вывод осуществляется на некотором множестве обучающих данных Y с использованием заданного множества предикатов $\{P_1, \dots, P_m\}$.

Данный метод основывается на следующем определении вероятностной закономерности, предложенном в работе [7].

Правило $P_{i1}, \dots, P_{in} \rightarrow P_0$ является *закономерностью*, если оно удовлетворяет следующим условиям:

1. $p(P_{i1}, \dots, P_{in}) > 0$,
2. $\forall \{P_{j1}, \dots, P_{jk}\} \subset \{P_{i1}, \dots, P_{in}\} \quad p(P_0 | P_{i1}, \dots, P_{in}) > p(P_0 | P_{j1}, \dots, P_{jk})$.

Здесь p – оценка условной вероятности правила.

Введем понятие *уточнения* правила. Правило $P_{i1}, \dots, P_{in}, P_{in+1} \rightarrow P_0$ является *уточнением* правила $P_{i1}, \dots, P_{in} \rightarrow P_0$, если оно получено добавлением в посылку правила $P_{i1}, \dots, P_{in} \rightarrow P_0$ произвольного предиката P_{in+1} , и $p(P_0 | P_{i1}, \dots, P_{in+1}) > p(P_0 | P_{i1}, \dots, P_{in})$.

Алгоритм семантического вероятностного вывода:

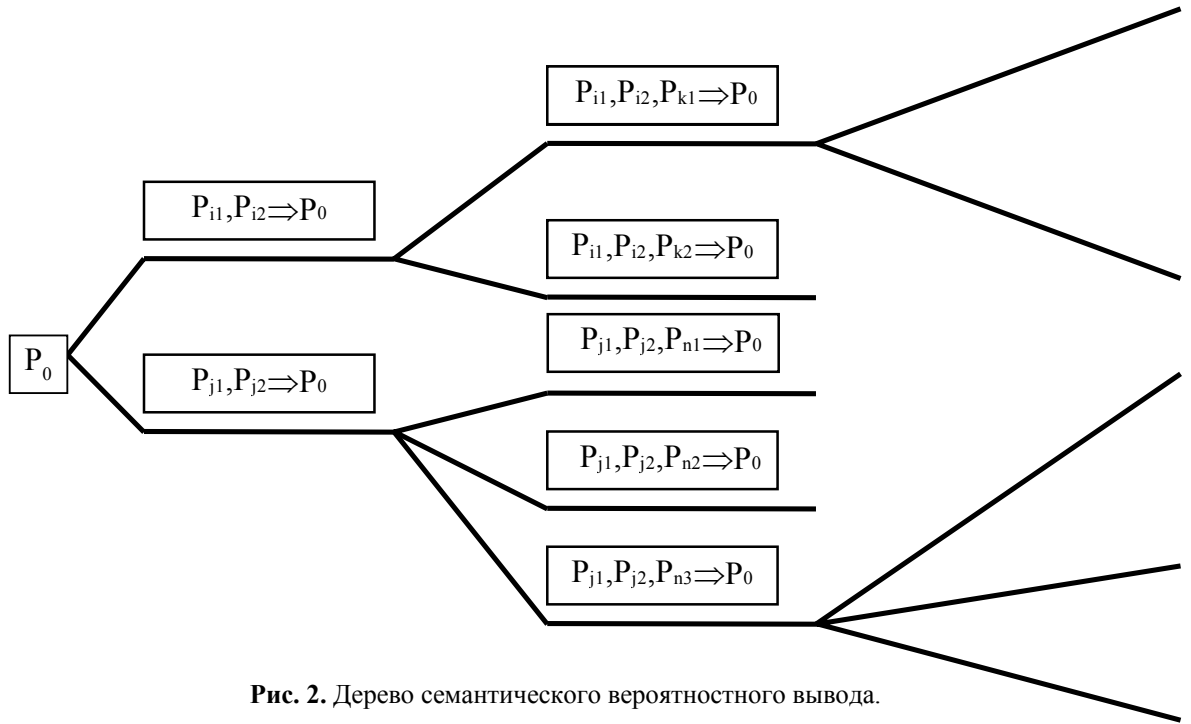


Рис. 2. Дерево семантического вероятностного вывода.

- На первом шаге генерируется множество уточнений правила $\rightarrow P_0$ (т.е. правила с пустой посылкой). Это множество будет состоять из правил единичной длины, имеющих вид $P_{ij} \rightarrow P_0$, для которых $p(P_0 | P_{ij}) > p(P_0)$.
- На k -ом ($k > 1$) шаге генерируется множество уточнений всех правил, созданных на предыдущем шаге. Т.е. для каждого правила $P_{i1}, \dots, P_{ik-1} \rightarrow P_0$, сгенерированного на $(k-1)$ -ом шаге, создается множество правил вида $P_{i1}, \dots, P_{ik-1}, P_{ik} \rightarrow P_0$, таких, что $p(P_0 | P_{i1}, \dots, P_{ik-1}, P_{ik}) > p(P_0 | P_{i1}, \dots, P_{ik-1})$.
- Проверяется, являются ли полученные правила закономерностями. Правила, не удовлетворяющие условиям закономерности, отсеиваются.
- Алгоритм останавливается, когда больше невозможно уточнить ни одно правило,

Для того чтобы избежать генерации статистически незначимых правил вводится дополнительный критерий – оценка на статистическую значимость. Правила, не удовлетворяющие этому критерию, отсеиваются, даже если они имеют высокую точность на обучающем множестве. Для оценки статистической значимости в алгоритме используется критерий Фишера (точный критерий Фишера для таблиц сопряженности) [8].

Очевидно, что все правила, полученные при помощи данного алгоритма, будут являться закономерностями. На рисунке 2 представлено дерево вывода, соответствующее описанному процессу.

Чтобы найти все закономерности $P_{i1}, \dots, P_{ik}, PG_{j1}, \dots, PG_{jn}, PA_i \rightarrow PG_j$ с максимальной вероятностью предсказывающие достижение цели G_j , строится дерево семантического вероятностного вывода на множестве данных истории деятельности анимата X и множестве оценок действий $d^j(t)$ с использованием набора предикатов P_j , которые использует данная функциональная система. Оценка условной вероятности p правила рассчитывается следующим образом: $p = \sum_{i \in I} d_i^j / \|I\|$, где I – множество моментов времени, когда может быть применено данное правило.

6. Извлечение подцелей. Изначально система управления аниматом имеет заданную априори иерархию функциональных систем. В простейшем случае она может состоять всего из одной функциональной систе-

мы. В процессе деятельности система управления может автоматически выявлять новые подцели и порождать соответствующие функциональные системы. Опишем процедуру порождения новых подцелей и функциональных систем.

Определим подцель как ситуацию, достижение которой значительно увеличивает вероятность достижения вышестоящей цели, и последующие действия из этой ситуации не могут быть определены однозначно.

Для выявления подцелей анализируется множество правил PR_j функциональной системы. Ситуация, описываемая предикатом $PG_i = P_1 \& \dots \& P_k$, будет являться подцелью G_i , если выполнены следующие условия:

- 1) $\forall R_1 = P_{i1}, \dots, P_{in}, PA_i \rightarrow PG_j$, такого, что $\{P_1, \dots, P_k\} \subseteq \{P_{i1}, \dots, P_{in}\}$, и $\forall R_2 = P_{j1}, \dots, P_{jm}, PA_j \rightarrow PG_j$, такого, что $\{P_{j1}, \dots, P_{jm}\} \subset \{P_{i1}, \dots, P_{in}\}$ и $\{P_1, \dots, P_k\} \not\subset \{P_{j1}, \dots, P_{jm}\}$, $p(R_1) - p(R_2) > \delta$.
- 2) $\exists R_1 = P_{i1}, \dots, P_{in}, PA_i \rightarrow PG_j$ и $R_2 = P_{j1}, \dots, P_{jm}, PA_j \rightarrow PG_j$, такие, что $\{P_1, \dots, P_k\} \subseteq \{P_{i1}, \dots, P_{in}\}$, $\{P_1, \dots, P_k\} \subseteq \{P_{j1}, \dots, P_{jm}\}$ и $A_i \neq A_j$.

Первое условие говорит о том, что добавление данной ситуации в условную часть правил должно значительно увеличивать оценку условной вероятности правил (более чем на δ , где δ - некоторый порог, например $\delta = 0.2$), что означает, что достижение этой ситуации значительно увеличивает вероятность достижения вышестоящей цели. Второе условие говорит о том, что после достижения данной ситуации возможны различные дальнейшие действия.

Таким образом, у каждой функциональной системы ΦC_j анализируется множество ее правил PR_j и выявляются новые подцели. Для каждой обнаруженной подцели G_i создается новая функциональная система ΦC_i , находящаяся ниже по иерархии системы ΦC_j и реализующая эту подцель. Для созданной функциональной системы ΦC_i при помощи семантического вероятностного вывода порождается множество закономерностей PR_i . Для этого просматривается все множество данных истории анимата X и выявляются случаи, когда подцель G_i была реализована и рассчитывается множество оценок действий $d^i(t)$ функциональной системы ΦC_i описанным выше способом. Для всех функциональных систем, находящихся на один уровень выше ΦC_i , набор предикатов обогащается еще одним предикатом PG_i , и генерируются новые правила. Тем самым, множество закономерностей этих функциональных систем обогащаются закономерностями, содержащими новую подцель G_i .

7. Описание эксперимента. Для исследования описанной выше системы управления был поставлен следующий эксперимент. При помощи компьютерной программы был смоделирован виртуальный мир и функционирующий в нем анимат, целью которого является сбор специальных объектов виртуального мира – «еды».

Мир анимата представляет собой прямоугольное поле, разбитое на клетки, и содержит четыре типа объектов: пустые клетки («трава»), препятствия («препятствие»), еду («еда») и таблетки («таблетка»). Объекты «препятствие» располагаются только по периметру виртуального мира, образуя тем самым его естественные границы. Анимат может перемещаться по полю, совершая три типа действий: шаг на клетку вперед («шаг»), поворот налево («налево»), поворот направо («направо»).

Данный эксперимент является усложнением известной тестовой поведенческой задачи фуражирования, в которой анимат должен научиться эффективно находить и собирать пищевые объекты. В нашем эксперименте виртуальный мир содержит еще один объект, который мы условно называли «таблетка». Чтобы съесть

еду анимат должен иметь при себе таблетку, которую он должен предварительно найти и подобрать. Когда он съест еду, таблетка исчезнет и, чтобы съесть следующую еду, он должен опять найти и подобрать таблетку, и т.д.

Чтобы съесть еду или подобрать таблетку анимату достаточно шагнуть на клетку, содержащую соответствующий объект. Однако если у него нет таблетки, то он не сможет съесть еду, и если анимат уже имеет одну таблетку, то пока он ее не использует для поедания еды, он больше не сможет подобрать ни одной таблетки. Изначально таблетки и еда случайным образом располагаются по полю. Когда анимат съедает еду или подбирает таблетку, то клетка, на которой находился этот объект, очищается, и новый объект такого же типа появляется в случайном месте поля, т.е. количество еды и таблеток в виртуальном мире всегда поддерживается постоянным.

Для ориентации в виртуальном мире анимат имеет десять сенсоров: «объект впереди-слева», «объект впереди», «объект впереди-справа», «объект слева», «объект в центре», «объект справа», «объект сзади-слева», «объект сзади», «объект сзади-справа» и «есть таблетка». Первые девять сенсоров, в соответствии со своими названиями, информируют анимат о типах объектов, расположенных в соответствующих клетках и принимают значения «травя», «препятствие», «еда» или «таблетка». Т.е., к примеру, сенсор «объект впереди» информирует о состоянии клетки перед аниматом, сенсор «объект в центре» - о состоянии клетки, на которой находится анимат и т.д. Еще один сенсор «есть таблетка» информирует анимат о наличии таблетки и принимает значения «да» или «нет».

Изначальный набор предикатов анимата состоит из тридцати семи сенсорных предикатов: по четыре предиката на каждый сенсор s , информирующий анимат о состоянии окружающих его клеток: (s = «травя»), (s = «препятствие»), (s = «еда»), (s = «таблетка»), и один предикат, говорящий о наличии таблетки: («есть таблетка» = «да»). А также трех активирующих предикатов: (A = «шаг»), (A = «налево») и (A = «направо»).

При старте система управления аниматом состоит только из одной, базовой, функциональной системы, целью которой является достижение ситуации одновременного наличия таблетки и ощущения еды центральным сенсором, соответствующий предикат-цель имеет вид $PG_0 = (\text{«центр»} = \text{«еда»} \text{ И } \text{«есть таблетка»} = \text{«да»})$. Когда анимат достигает эту цель он «поедает» еду.

8. Результаты эксперимента. Одной из основных задач эксперимента является демонстрация возможности автоматического формирования иерархии целей и результатов в целенаправленном поведении. В ходе эксперимента системой управления аниматом при каждом тестовом запуске стабильно обнаруживала новую подцель, описываемая предикатом-целью $PG_1 = (\text{«есть таблетка»} = \text{«да»})$ и создавала для нее соответствующую функциональную систему. Работа системы управления происходила следующим образом. При отсутствии у анимата таблетки срабатывала закономерность $PG_1 \rightarrow PG_0$ как наиболее вероятная в данной ситуации, которая передавала управление нижестоящей функциональной системе, реализующей поиск таблетки. Когда таблетка найдена у базовой функциональной системы начинали срабатывать правила с более высокой вероятностью, в результате чего она находила еду.

Для того чтобы оценить эффективность предлагаемой нами системы управления, в экспериментах также проводилось тестовое сравнение с системами, построенными на основании теории обучения с подкреплением (Reinforcement Learning), описанной в работах Р. Саттона и Э. Барто [10].

Для сравнения мы выбрали две системы управления, построенные на основе популярного алгоритма обучения с подкреплением Q-Learning. Суть алгоритма заключается в последовательном уточнении оценок

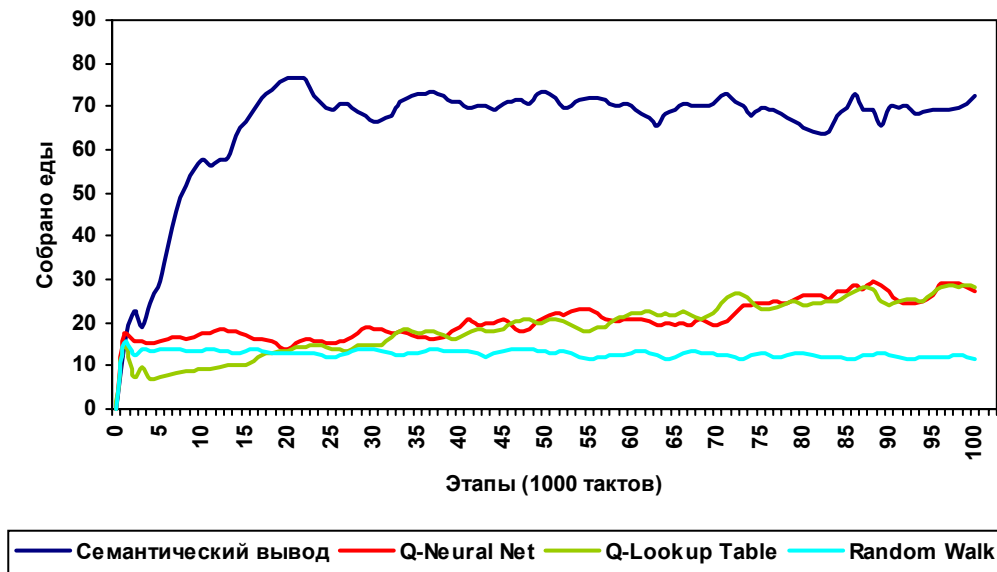


Рис. 3. Количество «еды», собранное аниматором с разными системами управления.

суммарной величины награды $Q(s_t, A_t)$, которую получит система, если в ситуации s_t она выполнит действие A_t , по формуле:

$$Q^{(i+1)}(s_t, A_t) = Q^{(i)}(s_t, A_t) + \alpha(r_t + \gamma \max_A Q^{(i)}(s_{t+1}, A) - Q^{(i)}(s_t, A_t)).$$

Первая из этих двух систем (Q-Lookup Table) основана на использовании таблицы, которая содержит Q-значения для всех возможных ситуаций и действий. Изначально эти значения таблицы заполняются случайным образом. В процессе работы в каждый такт времени система совершает действие и уточняет соответствующие Q-значения.

Вторая система (Q-Neural Net) использует аппроксимацию функции $Q(s_t, A_t)$ при помощи нейронных сетей. При этом для каждого возможного действия A_t используется своя нейронная сеть NN_t . В каждый такт времени система выбирает действие, чья нейронная сеть выдаст наибольшую оценку Q-значения, после чего действие совершается и происходит адаптация весов соответствующей нейронной сети.

Тестовое сравнение проводилось на поле размером 25 на 25 клеток. Количество таблеток и еды на поле поддерживалось постоянным: по 100 объектов каждого типа. Весь период функционирования аниматора был разбит на этапы по 1000 шагов (тактов). Оценивалось, какое количество еды соберет аниматор с разными системами управления за каждый этап работы. Очевидно, что после того как система управления полностью обучится и достигнет своего оптимального поведения, аниматор начнет собирать примерно одно и то же количество еды за один этап. Таким образом, можно оценить как эффективность каждой системы управления в целом, так и скорость ее обучения.

На рис. 3 представлены результаты тестового сравнения. Для каждой системы управления рассчитывались средние значения по результатам 20-ти испытаний. Продолжительность каждого испытания составляла 100,000 шагов, за это время аниматор должен был научиться эффективно решать поставленную задачу. Как видно на графике система управления на основе семантического вероятностного вывода превосходит системы Reinforcement Learning как по скорости обучения так и по качеству работы.

Системы управления на основе Reinforcement Learning в данном эксперименте показали плохую обучаемость и нестабильную работу. Основная проблема в работе этих систем была связана с тем, что они не могли за приемлемое время научиться стабильно адекватно реагировать на показания сенсоров о наличии таб-

леток и зачастую проходит мимо таблеток даже после 100,000 шагов обучения.

Дополнительные эксперименты показали, что система управления на основе нейронных сетей (Q-Neural Net) при увеличении длительности обучения до 300,000 – 500,000 шагов в некоторых случаях способна обучиться правильно реагировать на все показания сенсоров. Однако, по нашему мнению, столь длительный срок обучения является неприемлемым для адаптивной системы.

Система управления на основе использования таблицы Q-значений не смогла достичь оптимального поведения даже после 500,000 шагов. Во многом это связано с большим количеством возможных ситуаций: в данной задаче анимат может столкнуться с 137,538 различными ситуациями.

9. Выводы. Таким образом, результаты эксперимента показывают, что в условиях усложнения среды умение формировать и достигать подцели является принципиальным для эффективного достижения конечных целей. Несмотря на то, что в данной модели адаптивной системы управления используется достаточно простой способ формирования подцелей, уже эта возможность дает значительные преимущества в обучении. Как видно из эксперимента, использование иерархии функциональных систем и алгоритма выявления подцелей позволяет предлагаемой нами системе управления эффективно обучаться и решать поставленную задачу. Существующие подходы, основанные на нейронных сетях и Reinforcement Learning, не могут автоматически выявлять подцели и поэтому значительно проигрывают в усложненных экспериментах.

Литература

1. Витяев Е.Е. Целеполагание как принцип работы мозга // Модели когнитивных процессов (Выч. системы, 158), Труды ИМ СО РАН, Новосибирск, 1997, с.9-52.
2. Витяев Е.Е. Вероятностное прогнозирование и предсказание как принцип работы мозга // Измерение и Модели Когнитивных Процессов (Выч. системы, 162), Новосибирск, 1998, Стр. 14-40.
3. Витяев Е.Е. Формальная модель работы мозга, основанная на принципе предсказания // Модели Когнитивных Процессов. (Выч. системы, 164), Новосибирск, 1998, Стр. 3-61
4. Михиенко Е.В., Витяев Е.Е. Моделирование работы функциональной системы, VI Всероссийская научно-техническая конференция «Нейроинформатика-2004». Сборник научных трудов. В 2-х частях. Ч.2., М.: МИФИ, 2004, 124-129.
5. Витяев Е.Е. Объяснение Теории Движений Н.А.Бернштейна. VII Всероссийская научно-техническая конференция «Нейроинформатика-2005», Сборник научных трудов, часть 1., Москва, 2005, стр.234-240
6. Витяев Е.Е. Семантический подход к созданию баз знаний. Семантический вероятностный вывод. // Выч. системы, Новосибирск, 1992, вып.146, с. 19-49.
7. Витяев Е.Е. Метод обнаружения закономерностей и метод предсказания // Эмпирическое предсказание и распознавание образов (Выч. сист., вып. 67), Новосибирск, 1976, с. 54-68.
8. Кендал М., Стюарт А. Статистические выводы и связи // М., Наука, 1973, с. 899.
9. Анохин П.К. Принципиальные вопросы общей теории функциональных систем // Принципы системной организации функций. – М.: Наука, 1973. См. также:
<http://www.keldysh.ru/pages/BioCyber/RT/Functional.pdf>
10. Sutton R. and Barto A. Reinforcement Learning: An Introduction. – Cambridge: MIT Press, 1998. See also:
<http://www-anw.cs.umass.edu/~rich/book/the-book.html>