

Бурилина М.А.¹, Ахмадеев Б.А.¹

¹ Центральный экономико-математический институт Российской академии наук

Анализ многообразия архитектур и методов моделирования децентрализованных систем на основе агент-ориентированного подхода

АННОТАЦИЯ:

В статье представлена методология построения искусственного интеллекта агентов и гибридных агент-ориентированных моделей с использованием нейронных сетей, позволяющих преодолеть недостатки известных методов математической формализации поведения агентов микроуровня. Авторами разработан детальный обзор обучающихся и не обучающихся агентов, описаны математические подходы к проектированию взаимоотношений агентов в сети и предложены классификаторы искусственного интеллекта.

Данная работа может быть полезна при разработке оптимизационных моделей на основе методов имитационного компьютерного моделирования, нейронных сетей, а также при построении децентрализованных агент-ориентированных моделей социальных систем.

Работа выполнена при финансовой поддержке РФФИ, грант № 16-18-10296 «Разработка комплекса агент-ориентированных моделей, имитирующих социально-экономическую систему Евразийского континента, и детализированных до уровня отдельных индивидуумов».

КЛЮЧЕВЫЕ СЛОВА: агент-ориентированные модели, нейронные сети, логические алгоритмы, искусственное общество, поведение агентов, архитектура агентов, математическое моделирование поведения агентов

JEL: C45, C55, C63

ДЛЯ ЦИТИРОВАНИЯ:

Бурилина М.А., Ахмадеев Б.А. Анализ многообразия архитектур и методов моделирования децентрализованных систем на основе агент-ориентированного подхода // Креативная экономика. — 2016. — Т. 10. — № 7. — С. 829–848. — doi: [10.18334/ce.10.7.35364](https://doi.org/10.18334/ce.10.7.35364)

Бурилина Мария Александровна, научный сотрудник, Центральный экономико-математический институт Российской академии наук (maribu@mail.ru)

Ахмадеев Булат Анасович, научный сотрудник, Центральный экономико-математический институт Российской академии наук

ПОСТУПИЛО В РЕДАКЦИЮ: 16.07.2016 / ОПУБЛИКОВАНО: 30.07.2016

ОТКРЫТЫЙ ДОСТУП: <http://dx.doi.org/10.18334/ce.10.7.35364>

(с) Бурилина М.А., Ахмадеев Б.А. / Публикация: ООО Издательство "Креативная экономика"

Статья распространяется по лицензии Creative Commons CC BY-NC-ND (<http://creativecommons.org/licenses/by-nc-nd/3.0/>)

ЯЗЫК ПУБЛИКАЦИИ: русский



Введение

В настоящее время агент-ориентированный подход стал новым мощным направлением в социальном моделировании, применяемым во многих общественных науках. В основе данного подхода лежит взаимодействие отдельных сущностей, называемых агентами. Агентом является некий объект, воспринимающий информацию и действующий в определенной среде, у которого, в ответ на последовательность актов восприятия, выполняются различные действия.

У агентов могут быть показатели производительности, оценивающие поведение агента в среде. Рациональный агент действует так, чтобы максимизировать ожидаемые значения показателей производительности с учетом исторической последовательности воспринятой и обработанной информации, полученной агентом в прошлом.

Агенты могут обладать следующими характеристиками:

1. *Автономность* – если они действуют без чьего-либо вмешательства, они сами контролируют свои действия и внутреннее состояние.

2. *Социальность* – они взаимодействуют с другими агентами с помощью некоторого языка, алгоритма взаимодействия и т.д.

3. *Реактивность* – они воспринимают свое окружение, состоящее из других агентов, среды и т.д. и отвечают в режиме реального времени.

4. *Целенаправленность (проактивность)* – они не просто реагируют на окружение, но способны на целенаправленное поведение и проявление инициативы.

В работе [2] предложено обобщенное определение агента. Агент представляет собой набор качеств, характеризующий рассматриваемую единицу или индивида, находящегося внутри своей окружающей среды и являющаяся ее частью, воспринимающая сигналы из окружения и воздействующая на него для выполнения собственной программы действий.

Архитектуры интеллектуальных агентов

Интеллектуальные агенты можно описать как автономные сущности, способные заданным образом воспринимать информацию об окружающей среде, воздействовать на нее, а также получать и обрабатывать информацию, на основе которой принимают решения. Интеллектуальные агенты имеют способность «к принятию решений в

условиях неопределенности ситуации, действовать при отсутствии полной информации». Чаще всего такие агенты являются обученными, нежели запрограммированными для достижения конкретных целей. У архитектуры интеллектуальных агентов существует несколько видов:

- логическая архитектура
- реактивная архитектура
- архитектура BDI (belief – desire – intention)
- гибридная архитектура.

Рассмотрим каждую архитектуру интеллектуальных агентов подробнее. В основе логической архитектуры лежит принятие решений по определенным правилам. В этой архитектуре каждый агент действует согласно заданному ему представлению о мире, а решения принимаются с помощью сравнений и рассуждений по заданному образцу. В блоке принятия решений агентом используются знания об окружающей среде. Поочередно в результате проверки информации об окружающей среде агент принимает решение. Стоит отметить, что данная архитектура довольно проста и понятна, но может быть неэффективной в силу сложности формирования модели среды в символьном виде.

В реактивной архитектуре принятие решений происходит с помощью установления прямых связей между состояниями и действиями. Такие принципы были предложены Бруксом [17] в качестве альтернативного подхода к логической архитектуре. Данная архитектура основывается на двух особенностях. Первое заключается в том, что каждому действию, основанному на блоке наблюдения, соответствует определенное состояние окружающей среды. Реактивный агент не сохраняет данные о среде в базе знаний. Решение принимается мгновенно, исходя из входных данных. Агент наделен способностью реагировать не только на окружающую среду, но и на поведение других агентов.

Второй принцип заключается в том, что представление агента об окружающей среде и способах реагирования задается в виде отдельных уровней. Каждый из уровней обладает определенным приоритетом. При получении агентом сигналов из окружающей среды сначала решаются наиболее важные и приоритетные задачи, а после второстепенные.

В отличие от реактивных, проактивные агенты наделены способностью к предсказанию изменений в окружающей среде, планированию своих действий. В социологии используется термин

«проактивное управление»¹, с помощью которого стало возможным достижение ряда задач, к которым можно отнести быстрое реагирование на предупреждающие действия, с помощью которых можно достичь изменения будущих событий и предостеречь от неблагоприятных процессов, происходящих в объекте управления.

Одним из недостатков этой архитектуры является довольно узкая направленность систем реального времени и отсутствие инструментов для прогнозирования ситуации в будущем. Когда от агента требуется сложный анализ целей, выбор поведения и прогнозирование ситуации, реактивная архитектура не всегда может быть правильно задана.

Архитектура BDI заключается в том, что агенты принимают решения, основываясь на манипуляции структурами базы данных, включающих убеждения (beliefs), его цели (intentions) и пути достижения целей (desires) агентов [18]. Все динамические данные агента делятся на убеждения, желания и намерения. Агенты одновременно выполняют планирование своих дальнейших действий и выполнение запланированных целей. Поведение агентов в этой среде по своим свойствам дублирует процесс мышления человека при выборе необходимого поведения для достижения поставленных целей. Архитектура BDI основана на теории Братмана [15] о роли намерений практического рассуждения. В частности, Братман утверждал, что рациональные агенты имеют способность концентрировать свое практическое рассуждение на входе блока принятия решения и будут стремиться обойти варианты, которые конфликтуют с их намерениями. Агенты с архитектурой BDI способны оперативно реагировать на действия окружающей среды, своевременно анализировать поставленные цели и способы их достижения. Высокая интеллектуальная способность таких агентов отличает их от реактивных и логических. Один из недостатков состоит в сложности реализации архитектуры и высоких требованиях к вычислительной способности отдельных агентов, а также описанию среды программирования для разработки сложных приложений, где программист только задает правила поведения агентов, в которой агенты уже самостоятельно достигают своих целей, с использованием BDI понятий.

¹ Проактивное управление в социологии: предметное поле и терминологическая специфика [Электронный ресурс] // Агентное моделирование. — Режим доступа: <http://www.agentnoe-modelirovanie.ru/proactive-management.php>.

Гибридная архитектура является прототипом логической и реактивной, объединенной в один подход, реализуемый с помощью «программных слоев, каждый из которых представляет отдельный уровень абстракции» [21]. Для каждого отдельного вида поведения создается свой слой, то есть поведение агента выполняется несколькими подсистемами. Одной из проблем гибридной архитектуры является проектирование блока управления отдельными подсистемами для координации их действий и выбора общего решения.

Обучающиеся агенты

Обучение – одно из отличительных свойств головного мозга. В основе нейронных сетей лежит процесс обучения, который заключается в настройке архитектуры сети, в том числе и весов связей с целью достижения поставленной задачи. Настройка веса в нейронной сети происходит с помощью заданных обучающим параметрами. Такое свойство сети делает ее более эффективным по сравнению с другими системами. Они же работают по заранее заложенным правилам и параметрам. Среди методов обучения различают следующие классы:

- детерминированный,
- стохастический.

Рассмотрим эти подходы более подробно. К примеру, детерминированный подход влияет на изменения параметров сети, основываясь на текущих параметрах, а также величинах входов, существующих и желаемых выходов. Один из примеров таких методов является метод обратного распределения ошибки.

Стохастический метод заключается в том, что при обучении, сеть изменяет параметры случайным образом, а сам агент запоминает только те параметры сети, которые привели к положительным результатам. Примером стохастического метода обучения может служить блок-схема, представленная на *рисунке 1*.

В нейронных сетях существуют несколько алгоритмов обучения:

- алгоритмы обучения с учителем;
- алгоритм обучения без учителя.

Рассмотрим их подробнее. Процесс обучения с учителем подразумевает собой набор обучающих примеров сети. Каждый вид обучения поступает на входы сети, после чего проходит обработку внутри структуры нейронной сети, где вычисляется выходной сигнал сети, «который сравнивается с соответствующим значением целевого

вектора, представляющего собой требуемый выход сети»². В следствии чего по определенному правилу вычисляется ошибка, далее происходит изменение весовых коэффициентов связей внутри сети в зависимости от выбранного алгоритма. «Векторы обучающего множества предъявляются последовательно, вычисляются ошибки и веса подстраиваются для каждого вектора до тех пор, пока ошибка по всему обучающему массиву не достигнет приемлемо низкого уровня» [22].

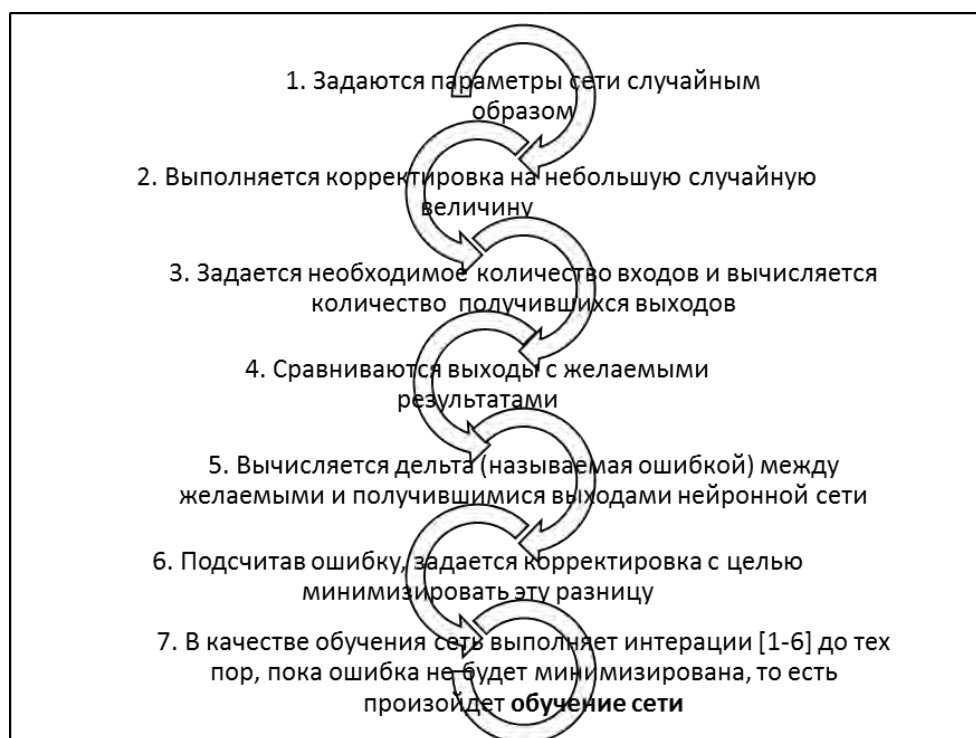


Рисунок 1. Пример построения архитектуры обучения нейронной сети (НС)³

Источник: составлено автором.

Важным свойством нейронных сетей является способность к обучению на основе взаимодействия с окружающей средой и повышение производительности в результате обучения.

«Обучение – это процесс, в котором свободные параметры нейронной сети настраиваются посредством моделирования среды, в которую эта сеть встроена. Тип обучения определяется способом подстройки этих параметров» [16]. Однако нейронная сеть не способна отображать желаемое поведение агентов, если она не настроена. В этом

² Обучение нейронной сети [Электронный ресурс] // Neuronus.com. – 2014. – 14 июня. – Режим доступа: <http://neuronus.com/nn/38-theory/240-algoritmy-obucheniya-iskusstvennykh-nejronnykh-setej.html>.

³ Нейронные сети – далее НС.

случае для достижения поставленной задачи необходимо условие, при котором обучающий алгоритм выстраивает вектора сети и ее связи в соответствии с желаемым поведением. Процесс обучения нейронной сети происходит поэтапно, которые представлены на *рисунке 2*.

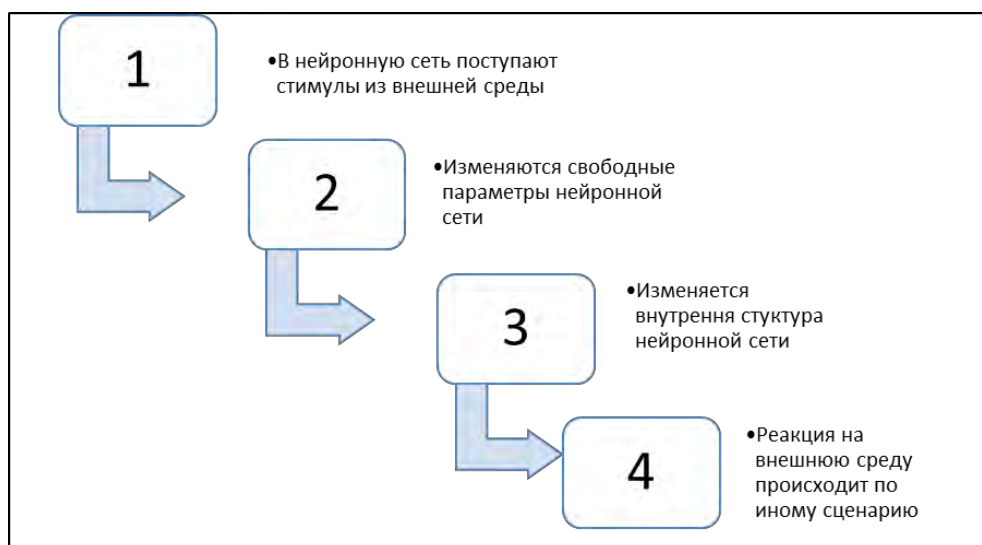


Рисунок 2. Последовательность настройки алгоритмов обучения нейронной сети

Источник: составлено автором

Однако, несмотря на представленную блок-схему алгоритма обучения сети, необходимо заметить, что «не существует универсального алгоритма обучения, который подходил бы для всех архитектур нейронных сетей. Существует лишь определенный набор средств, представленный множеством алгоритмов обучения, каждый из которых имеет свои достоинства»⁴ и недостатки. Основные отличия алгоритмов обучения заключаются в:

- различных способах настройки синоптических весов нейронов;
- способах связи обучаемой нейронной сети с внешним миром.

Выделяют два основных метода обучения нейронных сетей:

1. обучение с учителем,
2. обучение без учителя.

Обучение нейронной сети с учителем заключается в том, что в процессе обучения необходимо достичь такого поведения агентов, когда для каждого входного вектора из обучающегося множества соответствовало поведение выходного вектора. Такие вектора образуют

⁴ Обучение нейронной сети [Электронный ресурс] // Aportal.ru. – Режим доступа: <http://www.aiportal.ru/articles/neural-networks/learning-neunet.html>.

обучающую пару: «веса сети изменяют до тех пор, пока для каждого входного вектора не будет получен приемлемый уровень отклонения выходного вектора от целевого».

Если в процессе обучения сеть наделена верными ответами или выходы из сети приближены к необходимому значению, то вектор с таким набором алгоритмов называется алгоритмом обучения с учителем. Рассмотрим, как задается такой алгоритм математически.

Введем множество p -векторов, которое называется обучающимся множеством, где вектор $x^S \in X$ задает условие задачи, $a^S \in Y$ – решение задачи вектора x^S .

В процессе обучения сеть изменяет свои параметры, сохраняя при этом условие $X \rightarrow Y$.

Обучение нейронной сети без учителя имитирует биологическое поведение агента, что делает эту модель более правдоподобной с точки зрения искусственных нейронных сетей. Однако не все ученые согласны с этим высказыванием и считают метод обучения без учителя биологически неправдоподобным. Такой метод нашел эффективное применение для решения прикладных задач. Обучающее множество содержит в себе входные вектора, на основе которых сеть стремится выдать такие вектора, чтобы их значения были максимально приближены к заданным для обучения параметрам. «Алгоритм обучения нейронной сети подстраивает веса сети так, чтобы получались согласованные выходные векторы, т.е. чтобы предъявление достаточно близких входных векторов давало одинаковые выходы»⁵.

Правило обучения Хебба

В ходе физиологических и психологических исследований и наблюдений Хебб выдвинул гипотезу о способах обучения биологических нейронов. В основе правила Хебба лежит следующая теория: необходимо «увеличивать вес активного входа нейрона, если выход этого нейрона должен быть активным, и наоборот – уменьшить вес активного входа нейрона, если выход этого нейрона не должен быть активным» [19]. Рассмотрим алгоритм, который представлен на *рисунке 3*.

⁵ Обучение нейронной сети [Электронный ресурс] // Alportal.ru. – Режим доступа: <http://www.aiportal.ru/articles/neural-networks/learning-neunet.htm.l>



Рисунок 3. Цепочка последовательности обучения методом Хебба
Источник: Составлено автором

Правило обучения по Хеббу в нейронных сетях можно назвать самым ранним из всех обучающихся правил. Отличительной чертой правила Хебба считается, что от активности нейронов зависит их вес, то есть, изменяя вес нейрона, можно привести сеть к определенному состоянию.

Метод обратного распространения ошибки (error backpropagation algorithm)

Основная идея такого метода состоит в том, что связи настраиваются в обратном направлении: распространение сигналов ошибки идет от выходов сети к ее входам. В этом и заключается отличие метода от классического, в котором прямое распространение сигналов происходит в обычном режиме работы и двигается от входа к выходу.

Метод error backpropagation algorithm был предложен впервые в 1974 году в работах Галушкина А.И. [10] и Werbos P. J. [11]. В том числе метод был предложен в 1986 г. Румельхартом, Макклеландом и Вильямсом [12]. В распараллеленном процессе распределения обучение многослойной нейронной сети начинается с программирования необходимого поведения сети, а далее вычисляется необходимая реакция агента для достижения заданного результата.

В работе [13] рассматривается поведение обучающихся агентов, в которых заложена функция полезности, функция вознаграждения и сама стратегия поведения агента. Обучение агента проходит следующим образом: на каждом временном этапе агент взаимодействует с внешней

средой, откуда получает информацию. Выполняя определенное действие, агент из одного состояния переходит в следующее, за что получает вознаграждение. Стратегия отображается в вероятность выбора каждого действия агента и корректируется в процессе обучения, основываясь на предыдущем опыте. В основе был использован алгоритм (*R-learning*). В нейронных сетях, несомненно, первостепенной задачей является выбор вида сети и алгоритмом обучения. После чего на вход нейронной сети задается набор атрибутов текущего состояния, а для обучения сети был задан эволюционный подход с помощью генетического алгоритма. Стоит отметить преимущество случайно действующего агента относительно времени поиска решения (для выбранных параметров модели), а вот по времени обучения *нейросетевой агент превосходит агентов с алгоритмом R-learning около 2,5 раз.*

Генетические алгоритмы

Одним из методов имитации поведения агентов является программирование взаимодействия агентов в среде с использованием природных инстинктов. Речь идет о наложении генетических алгоритмов поведения в основу биологической популяции [19].

Зададим популяцию с размерностью N с помощью вектора $P = \{p^i\} = \{p^1, \dots, p^n\}$ и особью p^i . Каждый вектор p^i включает в себя набор параметров, необходимых для описания особей. Тогда пусть функция $E(p)$ зависит от вектора p , а элементы множества P эволюционируют по правилам:

$E(p^0) \rightarrow 0$, то особь p^0 считается удачной, соответственно, вероятность положительного исхода события при размножении выше, а вероятность гибели такой особи меньше.

$E(p^0) \rightarrow \infty$, то особь p^0 считается неудачной, вероятность размножения такой особи стремится к 0, а вероятность гибели велика.

Мутация особи происходит с отклонением Δp , вектор мутирующей особи выглядит так: $\tilde{p}^0 = p^0 + \Delta p$

Размножение особей задается по правилу, выбранному в модели, а гибель происходит с заданной вероятностью. Однако точная теория эволюции пока не построена, поэтому любая апробация модели носит сугубо экспериментальный характер.

Логическая архитектура агентов

Примером логической архитектуры агентов может служить агент-ориентированная модель взаимодействия «стартапов» и инвесторов, отраженная в работе Ахмадеева Б.А. [20]. В работе представлены алгоритмы принятия решения агентов «стартап», «венчурный инвестор» и «корпорация», взаимодействующих в рамках процессов венчурного инвестирования. Пример алгоритма агента «стартап» приведен на *рисунке 4*.

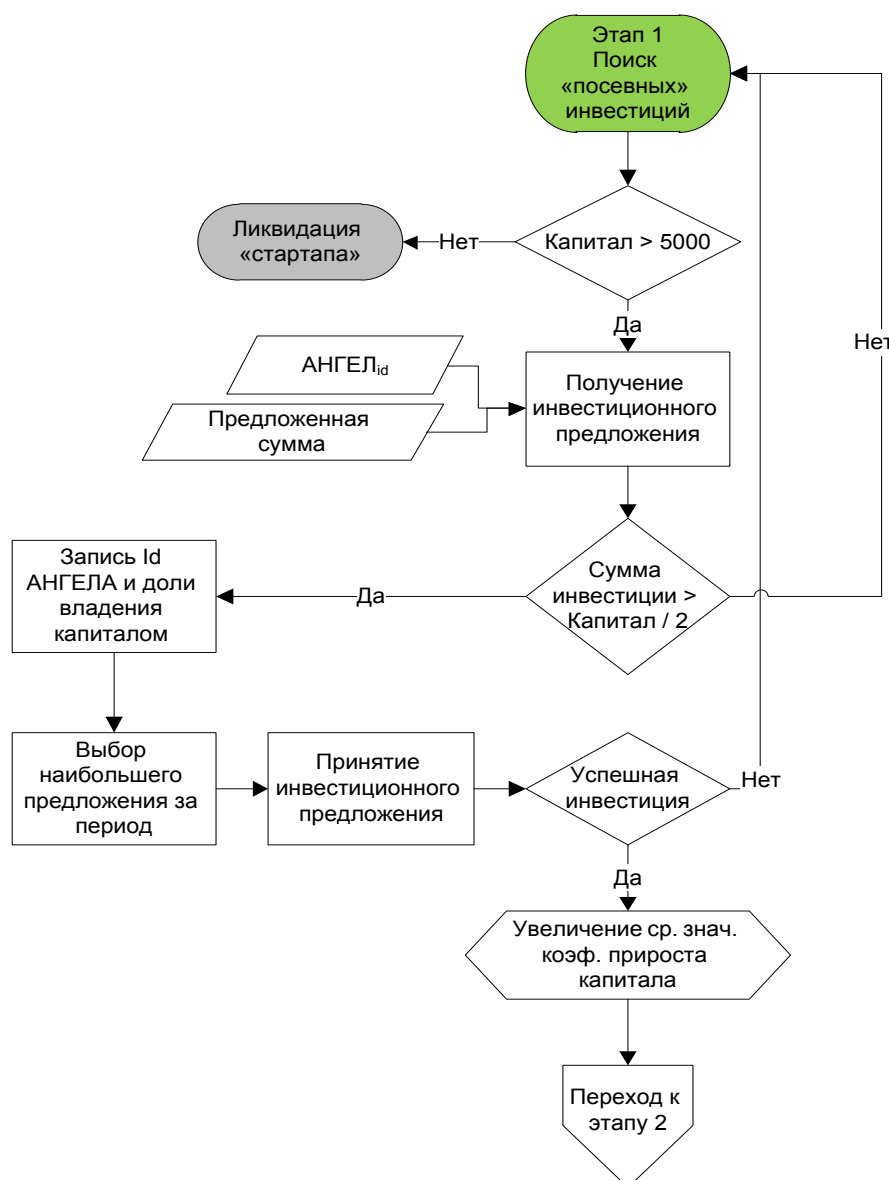


Рисунок 4. Алгоритм поиска объекта инвестиций

Источник: [20]

В данной архитектуре каждый агент-стартап действует согласно его представлению о мире венчурных инвестиций, а решения принимаются с помощью сравнений и рассуждений согласно алгоритму оценки. Поочередно в результате проверки информации об окружающей среде агент принимает решение о принятии инвестиционного предложения или отказе, а в случае недостатка собственного капитала агент уничтожается.

Реактивная архитектура агентов

Модель «жизнь»

Модель клеточного автомата «жизнь» [9] относится к играм с т.н. «нулем игроков», т.к. развитие в этой игре происходит автоматически и зависит только от начально заданных пользователем параметров. Однако в процессе эволюции самосоздаются совершенно различные алгоритмы и поведение фигур исходя из правил игры.

Классические правила игры определяются следующим образом: пространство игры – это бесконечная двумерная плоскость, состоящая из квадратных клеток, каждая из которых может принимать значения «живой» или «мертвый». Каждая клеточка взаимодействует с восьмью соседними клетками, расположенными по краям вертикально, горизонтально и диагонально. В каждый временной интервал выполняются действия согласно следующему алгоритму:

1. каждая живая клетка с меньше чем 2-мя живыми соседними клетками погибает (недонаселение);
2. каждая живая клетка с 2-мя или 3-мя живыми соседними клетками продолжает жить в следующей итерации (равновесное население);
3. каждая живая клетка с больше чем 3-мя живыми соседними умирает (перенаселение);
4. каждая мертвая клетка с 3-мя живыми соседними клетками становится живой в следующий период (репродукция).

Важное значение имеет именно первоначальный паттерн расположения в пространстве жизни, далее действия над клетками (продолжение жизни, смерть или репродукция) происходят одновременно в каждый временной период.

В процессе экспериментов было выявлено, что в игре образуются устойчивые фигуры, такие как:

– «натюрморты» – не меняющиеся в процессе жизни (рис. 5);

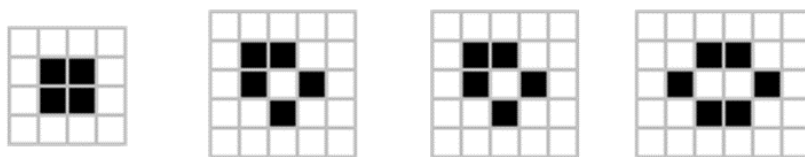


Рисунок 5. Примеры «натюрмортов» в игре «жизнь»

Источник: Conway's Game of Life [Electronic resource] // Wikipedia. – Mode of access: https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life

– «осцилляторы» – фигуры, циклично повторяющие свое поведение спустя какое-то количество периодов (рис. 6);

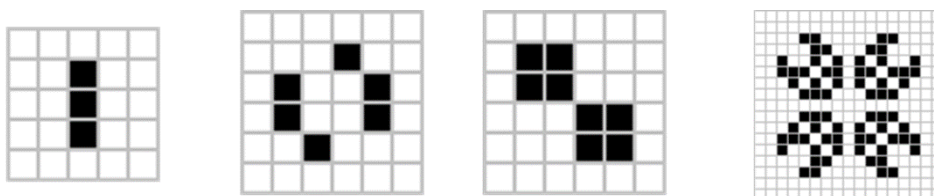


Рисунок 6. Примеры «осцилляторов» в игре «жизнь»

Источник: Conway's Game of Life [Electronic resource] // Wikipedia. – Mode of access: https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life

– «космические корабли» – фигуры, повторяющие свое поведение спустя какое-то количество периодов со смещением своего положения относительно координатной плоскости (рис. 7).

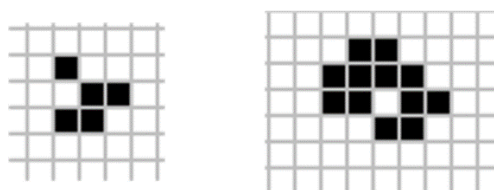


Рисунок 7. Примеры «космических кораблей» в игре «жизнь»

Источник: Conway's Game of Life [Electronic resource] // Wikipedia. – Mode of access: https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life

Несмотря на то, что «жизнь» основана на простых правилах, она привлекает внимание ученых из разных областей. Игра и ее различные модификации изучаются и имеют влияние на некоторые разделы точных наук, таких как физика, информатика и математика.

Модель Хенинга

Модель Хенинга [8] представляет собой еще один простейший пример клеточного автомата на двумерной плоскости, где агенты реактивно реагируют на изменения окружающей среды. Модель имеет n -мерный вектор состояний Ω , элементы которого могут принимать значения 0 или 1:

$$\Omega = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix},$$

где $x_i \in \{0,1\}, i = \overline{1,n}$. Если $x_i = 0$, то i -й агент мертв, если $x_i = 1$, то агент жив. В каждый период состояние меняется – агенты умирают или рождаются. Состояние системы в момент времени k представим как Ω_k .

В нулевой период все агенты живы, т.е. Ω_0 есть n -мерный вектор, все элементы которого равны 1. Далее существует $(n \times n)$ -матрица перехода M_k , которая в нулевой период совпадает с единичной матрицей. Процедура перехода из состояния M_{k-1} к M_k выглядит следующим образом:

В каждой строке матрицы M_{k-1} выбирается один элемент и заменяется на некоторую случайную величину из интервала от -1 до 1 . Так, матрица M_1 примет вид:

$$M^1 = \begin{pmatrix} 1 & \cdots & a_{1j_1} & \cdots & 0 \\ \vdots & 1 & \cdots & a_{2j_2} & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & a_{nj_n} & \cdots & 1 \end{pmatrix},$$

где для всех $i \in \{1, \dots, n\}$ случайно выбирается целое значение $j_i \sim \overline{U}[1, n]$, а также некоторое значение $a_{i,j_i} \sim U[-1, 1], i = \overline{1, n}$. Здесь $U[-1, 1]$ – равномерное распределение в диапазоне $[-1, 1]$, а $\overline{U}[1, n]$ – целочисленное равномерное распределение, в соответствии с которым случайным и равновероятным образом выбирается целое число из диапазона от 1 до n . Описанную процедуру обозначим через F , т.е. для любого $k \in \mathbb{N} \cup \{0\}$ будет справедливо $F(M_{k-1}) = M_k$. Далее вычисляется промежуточный параметр $\omega^k = M^k \Omega^k$. Тогда вектор состояний Ω^{k+1} будет определяться по правилу:

$$\Omega_j^{k+1} = \begin{cases} 1 & \text{если } \omega_j^k > 0 \\ 0 & \text{если } \omega_j^k \leq 0 \end{cases}$$

Таким образом элемент M_{ij}^k матрицы M^k , $k = 1, 2, \dots$ характеризует воздействие j -го агента на агента i в период k . Если $M_{ij}^k > 0$, то воздействие на агента i положительно, если $M_{ij}^k < 0$, то воздействие отрицательно. Итоговое воздействие на агента i в k -й период определяется по формуле

$$\sum_{j=1}^n M_{ij}^k \Omega_j^k.$$

Формально итоговое воздействие на агента i это произведение i -й строки матрицы M^k на вектор состояний Ω^k . Если это произведение больше 0, то особь либо выживает, либо рождается. В противном случае особь считается мертвой. Описанный выше алгоритм преобразования матрицы M^{k-1} в матрицу M^k означает, что в каждый период времени изменение воздействия на любого агента $i = \overline{1, n}$ происходит со стороны другого, случайно выбранного агента j , на случайную величину a_{i,j_i} .

Роевые алгоритмы

Широкий интерес для экономической науки представляют методы оптимизации, построенные на основе работы природных систем, называемые обобщенно «методы роевого интеллекта». Эти методы применяются во многих отраслях науки и техники для поиска глобального максимума или минимума целевой функции. Эффективность методов роевой оптимизации подтверждается эволюцией природы, которая за все время существования выработала наиболее эффективные методы оптимизации различных задач. Методы роевого интеллекта описывают коллективное поведение децентрализованной самоорганизующейся системы, в которой множество агентов локально взаимодействуют друг с другом и окружающей средой. Как правило, единичный агент (боид) в таком алгоритме представляет собой простой реактивный механизм, но, взаимодействуя все вместе, они создают интеллектуальный алгоритм более высокого уровня или т.н. «роевой интеллект».

Понятие роевого интеллекта (Swarm intelligence) было введено Beni G., Wang, J. [1] в 1989 году. Примером систем роевого интеллекта в природе являются муравьиная колония, пчелиный рой, стая птиц, рыб,

алгоритм капель воды и т.д. Ниже мы приводим описания и простейшие алгоритмы работы некоторых наиболее интересных методов.

Метод роя частиц

Метод роя частиц (МРЧ) – численный метод оптимизации, позволяющий произвести оптимизации какой-либо функции без знания ее точного градиента. Изначально этот метод был предназначен для имитации социального поведения [2, 3]. Когда алгоритм был упрощен, была сделана попытка его применения для выполнения численной оптимизации. Данный метод оптимизирует функцию путем поддержания популяции возможных решений, которые называют частицами, и перемещает эти частицы в пространстве решений согласно собственной формуле. Перемещение осуществляется согласно принципу наилучшего найденного в этом пространстве положения, постоянно изменяется при нахождении частицами выгодных положений.

Агентами в МРЧ являются частицы в допустимом пространстве параметров задачи оптимизации. На каждой итерации частицы имеют в этом пространстве некоторое положение и вектор скорости. Для каждого положения вычисляется соответствующее значение целевой функции, и на этой основе по определенным правилам частица меняет свое положение и скорость [16, 17]. Далее представлен алгоритм оптимизации в классическом методе роя частиц:

$$X_{i,t+1} = X_{i,t} + V_{i,t}, \quad (1)$$

$$V_{i,t+1} = V_{i,t} + U[0, \beta] \cdot (x_{i,t}^b - x_{i,t}), \quad (2)$$

где t – номер итерации;

x_i – вектор координаты i -ой частицы;

x^b – вектор координаты лучшей частицы;

$U[0, \beta]$ – вектор псевдослучайных чисел в интервале от 0 до β ;

V_i – скорость i -ой частицы

Муравьиный алгоритм

Муравьиный алгоритм основан на методе природной оптимизации, создавшемся в ходе эволюции муравьиных колоний. Это один из эффективных алгоритмов для нахождения приближенных решений задачи коммивояжера и других задач, решаемых с помощью

графов. Данный метод впервые был предложен исследователем Marco Dorigo [4].

Суть метода заключается в имитации алгоритма поведения муравьев при поиске кратчайшего пути от колонии до еды или другого объекта. В основе алгоритма лежит маркировка удачных дорог специальными феромонами, чтобы другие муравьи могли найти путь, сделанный другими. Работа начинается с размещения муравьев в вершинах графа, после чего происходит движение муравьев, где вероятность какого-либо направления определяется на основании формулы:

$$P_i = \frac{l_i^q \cdot f_i^p}{\sum_{k=0}^N l_k^q \cdot f_k^p} \quad (3)$$

Где P_i – вероятность перехода дорогой i ;

l_i – длина i -ого перехода;

f_i – количество феромонов на i -ом переходе;

q – величина, определяющая «жадность» алгоритма;

p – величина, которая определяет «стадность» алгоритма i

$q + p = 1$.

Следовательно, при выборе пути муравей ориентируется не только на свою цель – пройти кратчайший путь, но и на опыт других агентов-муравьев. При этом уровень феромона на каком-либо определенном пути определяет желание особи выбрать этот или другой путь.

Очевидна неизбежность попадания в локальный максимум или минимум ввиду случайности выбора пути на начальных этапах, однако данная проблема решается путем уменьшения концентрации (или испарения) феромонов на отдельном пути с течением времени. Это является некоторой обратной связью в общении агентов между собой. Такое «испарение» феромонов происходит равномерно по всем путям, таким образом, на более короткий путь тратится меньше времени, что сохраняет концентрацию феромона на нем в течение более длительного времени [6].

Алгоритм капель воды

Алгоритм интеллектуальных капель воды – очередной алгоритм, основанный на роевой оптимизации, использующий также естественные методы нахождения оптимальных путей к месту назначения, которые используют природные реки.

Этот алгоритм находит оптимальные или близкие к ним пути, вытекающие из взаимодействия между каплями воды в процессе тока воды в русле реки. В алгоритме искусственные капли воды находят оптимальный маршрут на пути наименьшего сопротивления, таким образом, завися друг от друга и меняя свое окружение реактивно [5].

Заключение

1. Разработанная методология и анализ различных подходов моделирования, основанных на агент-ориентированном подходе, может способствовать решению оптимизационных задач в области экономического прогнозирования, а также управления рисками в стохастических децентрализованных динамических системах, к которым относятся практически все социально-экономические процессы. Также данный обзор должен помочь при построении архитектуры агент-ориентированной системы при определении типа агентов и способов его обучения для достижения поставленной цели в кратчайшие сроки.

2. С появлением современных технологий высокоскоростных параллельных вычислений методы агент-ориентированной оптимизации, основанные на численных вычислениях, постепенно заменяют аналитические методы оптимизации. Такие численные методы оптимизации, как роевые алгоритмы, нейронные сети и др., могут помочь найти оптимальное решение некоторой задачи в кратчайшие сроки. Также следует отметить, что аналитическое решение невозможно при решении задач, связанных с динамическими системами с высокой стохастичностью. Появление агент-ориентированной архитектуры моделирования можно считать следующим этапом развития компьютерного имитационного моделирования.

3. Резюмируя вышесказанное, нужно отметить, что в связи с активным развитием высокоинтеллектуальных систем и компьютерных технологий, необходимо постоянно анализировать новые появляющиеся подходы с целью их эффективного применения при моделировании социально-экономических процессов. В следующих работах авторов планируется применение описанных в данном исследовании подходов

при разработке комплекса агент-ориентированных моделей, имитирующих социально-экономическую систему с акцентом на формализацию поведения агентов на уровне отдельных индивидуумов и выполнении прогностических расчетов с применением суперкомпьютеров.

ИСТОЧНИКИ:

1. Xie X. L., Beni G. A validity measure for fuzzy clustering // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 1991. — Vol. 13. — № 8. — P. 841-847. — doi: 10.1109/34.85677
2. Parsopoulos K.E., Vrahatis M.N. Recent approaches to global optimization problems through Particle Swarm Optimization // Natural Computing. — 2002. — Vol. 1. — № 2. — P. 235-306. — doi: 10.1023/A:1016568309421
3. Kennedy J., Eberhart R. Particle Swarm Optimization // Proceedings of IEEE International Conference on Neural Networks. — 1994. — Vol. 4. — P. 1942-1948. — doi: 10.1109/ICNN.1995.488968
4. Dorigo M., Gambardella L. M. Ant colony system: a cooperative learning approach to the traveling salesman problem // IEEE Transactions on Evolutionary Computation. — 1997. — Vol. 1. — № 1. — P. 53–66. — doi: 10.1109/4235.585892
5. Shah-Hosseini H. The intelligent water drops algorithm: a nature-inspired swarm-based optimization algorithm // International Journal of Bio-Inspired Computation. — 2009. — Vol. 1. — № 1/2. — P. 71–79. — doi: 10.1504/IJBIC.2009.022775
6. Кажаров А.А., Курейчик В.М. Муравьиные алгоритмы для решения транспортных задач // Известия Российской академии наук. Теория и системы управления. — 2010. — № 1. — С. 32-45.
7. Franklin S., Graesser A. Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents // Intelligent Agents III Agent Theories, Architectures, and Languages. — Berlin: Springer Verlag, 1997. — P. 21-35. — (Lecture Notes in Computer Science, Vol. 1193).
8. Henning P. A. Computational evolution // Complexity and Artificial Markets. — Berlin, Heidelberg: Springer-Verlag, 2008. — P. 175–193. — (Lecture Notes in Economics and Mathematical Systems, Vol. 614).
9. Gardner M. Mathematical Games — The fantastic combinations of John Conway's new solitaire game "life" // Scientific American. — 1970. — № 223. — P. 120–123.
10. Галушкин А.И., Зак Л.С., Тюхов Б.П. К сравнению критериев оптимизации адаптивных систем распознавания образов // Кибернетика. — 1970. — № 6. — С. 122-130.
11. Werbos P. J. Beyond regression: New tools for prediction and analysis in the behavioral sciences: Ph.D. thesis. — Cambridge: Harvard University, 1974. — 234 p.
12. Rumelhart D. E., Hinton G. E., Williams R. J. Learning Internal Representations by Error Propagation // Parallel Distributed Processing. Vol. 1. — Cambridge: MIT Press, 1986. — 567 p. — P. 318-362.

13. Сотников И.Ю., Григорьева И.В. Адаптивное поведение программных агентов в мультиагентной компьютерной игре // Вестник Кемеровского государственного университета. — 2014. — № 4-2. — С. 65-71.
14. Маслобоев А.В. Гибридная архитектура интеллектуального агента с имитационным аппаратом // Вестник Мурманского государственного технического университета. — 2009. — Т. 12. — № 1. — С. 113-124.
15. Bratman M. E. Intentions, Plans, and Practical Reason. — Cambridge: Harvard University Press, 1987. — 146 p.
16. Хайкин С. Нейронные сети: полный курс. — 2-е изд. — М.: ИД «Вильямс», 2006. — 1104 с.
17. Романов В.П., Бадрина М.В. Информационные технологии моделирования финансовых рынков. — М.: Финансы и статистика, 2010. — 288 с.
18. Social Self-Organization. — Berlin, Heidelberg: Springer-Verlag, 2012. — 454 p. — (Understanding Complex Systems).
19. Васенков Д.В. Методы обучения искусственных нейронных сетей // Компьютерные инструменты в образовании. — 2007. — № 1. — С. 20-29.
20. Ахмадеев Б.А. Моделирование эффективного взаимодействия «стартапов», инвесторов и корпораций // Креативная экономика. — 2015. — Т. 9. — № 6. — С. 659-682.
21. Селяков И.С. Определение структуры многоагентной системы для моделирования популяции животных // Труды СПИИРАН. — 2007. — № 4. — С. 437-444.

Marina A. Burilina, Researcher, Central Economics and Mathematics Institute of the Russian Academy of Sciences

Bulat A. Akhmadeev, Researcher, Central Economics and Mathematics Institute of the Russian Academy of Sciences

Analysis of the variety of architectures and methods for modeling of decentralized systems based on the agent-oriented approach

ABSTRACT

The article gives the methodology for building of the artificial intelligence, agents and hybrid agent-oriented models using neuron networks that allow to negotiate disadvantages of known methods for mathematical formalization of behavior of micro-level agents. The authors have developed the detailed review of learning and non-learning agents, have mathematically described approaches to design of interrelations between agents in the network and have suggested classifiers for artificial intelligence.

This work may be useful for development of optimization models based on methods for simulating computer modeling, neuron networks as well as for building of decentralized agent-oriented models of social systems.

KEYWORDS: agent-oriented models, neuron networks, logical algorithms, artificial society, behavior of agents, architecture of agent, mathematical modeling of agents' behavior
