

ТЕХНИЧЕСКИЕ НАУКИ

Лагун Дмитрий Александрович

инженер-программист

Белорусский государственный университет информатики

и радиоэлектроники, ООО «АйтиГраунд»

Гродель Юлия Валентиновна

инженер-программист

ЗАО «Оксаджайл»

г. Минск, Республика Беларусь

ЭВОЛЮЦИОННЫЙ ПОДХОД

К ИГРОВОМУ ИСКУССТВЕННОМУ ИНТЕЛЛЕКТУ

***Аннотация:** в статье кратко рассматриваются подходы к разработке игрового искусственного интеллекта; выявляются и анализируются проблемы проектирования игрового искусственного интеллекта; дается краткий анализ использования существующих технологий и решений для моделирования поведения объектов в виртуальной среде. Как результат, представлено решение, описывающее недетерминированное поведение объекта при наименьших затратах вычислительной мощности.*

***Ключевые слова:** искусственный интеллект, конечные автоматы, генетические алгоритмы, оптимизация, компьютерные игры.*

Компьютерные игры относятся к числу актуальных проблем современной науки, так как представляют собой уникальный продукт развития техники и современной личности. Игры помогают нам смоделировать различные жизненные ситуации, проблемы и выдают некоторые возможные пути их решения. Игра содержит в себе все необходимые предпосылки для естественного развития личности и культуры общества. С совершенствованием компьютеров совершенствовались и игры, привлекая все больше и больше людей. На сегодняшний день ком-

пьютерная техника достигла такого уровня развития, что позволяет программистам разрабатывать очень реалистичные игры с хорошим графическим и звуковым оформлением.

Очень сложно найти компьютерную игру, которая обходится без AI (сокр. от англ. Artificial intelligence – «искусственный интеллект»). Игровой искусственный интеллект – это попытка смоделировать поведение реальных объектов в виртуальной среде. Как ни странно, несмотря на значительное развитие отрасли со времени появления первых игр, основные схемы функционирования искусственного интеллекта остались принципиально неизменными. Наиболее простая, классическая схема функционирования логики компьютерного объекта – это система ветвлений. Если разработчик предусмотрел множество подобных ветвлений на все случаи жизни, то компьютерный объект будет вести себя реалистично или, по меньшей мере, сколько-нибудь разумно. Такой подход прост в реализации, но жесткая шаблонная логика поведения компьютерных оппонентов даже при кропотливой и подробной реализации вариантов рано или поздно исследуется игроками буквально до последней инструкции – и в ней непременно находятся изъяны [1].

Для создания более реалистичного и непредсказуемого сценария поведения объектов лучше использовать недетерминированные подходы к реализации искусственного интеллекта, с вероятностными результатами. Рациональное использование классических парадигм самообучения ИИ (прим. нейронные сети, генетические алгоритмы) приведет к желаемому результату, правда, здесь акценты несколько смещены. Если основная цель классического искусственного интеллекта – найти правильное решение поставленной задачи, то от игрового требуют высокой производительности и «интересности». Большинство недетерминированных методов являются излишне сложными для игровых приложений в части реализации и отладки, а получаемый на выходе результат без соответ-

ствующей оптимизации может быть хуже, чем у классических шаблонных решений [2, 5]. Комбинирование конечных автоматов с генетическим алгоритмом является одним из путей решения данной проблемы.

Генетический алгоритм представляет собой метод, отражающий естественную эволюцию методов решения проблем, и в первую очередь задач оптимизации. Генетические алгоритмы – это процедуры поиска, основанные на механизмах естественного отбора и наследования. В них используется эволюционный принцип выживания наиболее приспособленных особей. Они отличаются от традиционных методов оптимизации несколькими базовыми элементами. В частности, генетические алгоритмы:

- обрабатывают не значения параметров самой задачи, а их закодированную форму;
- осуществляют поиск решения исходя не из единственной точки, а из их некоторой популяции;
- используют только целевую функцию, а не ее производные либо иную дополнительную информацию;
- применяют вероятностные, а не детерминированные правила выбора.

Перечисленные четыре свойства, которые можно сформулировать также как кодирование параметров, операции на популяциях, использование минимума информации о задаче и рандомизация операций приводят в результате к устойчивости генетических алгоритмов и к их превосходству над другими широко применяемыми технологиями [3, 4].

При описании генетических алгоритмов используются определения, заимствованные из генетики. Например, речь идет о популяции особей, а в качестве базовых понятий применяются ген, хромосома, генотип, фенотип, аллель. Также используются соответствующие этим терминам определения из технического лексикона, в частности, цепь, двоичная последовательность, структура.

Конечный автомат, описывающий поведение объекта можно сгенерировать с помощью генетического алгоритма. Для этого необходимо построить граф переходов конечного автомата, описывающего некоторое поведение объекта, и каждое действие закодировать с помощью двоичных чисел. Полученный результат преобразовывается в строку, которая подается на вход генетического алгоритма.

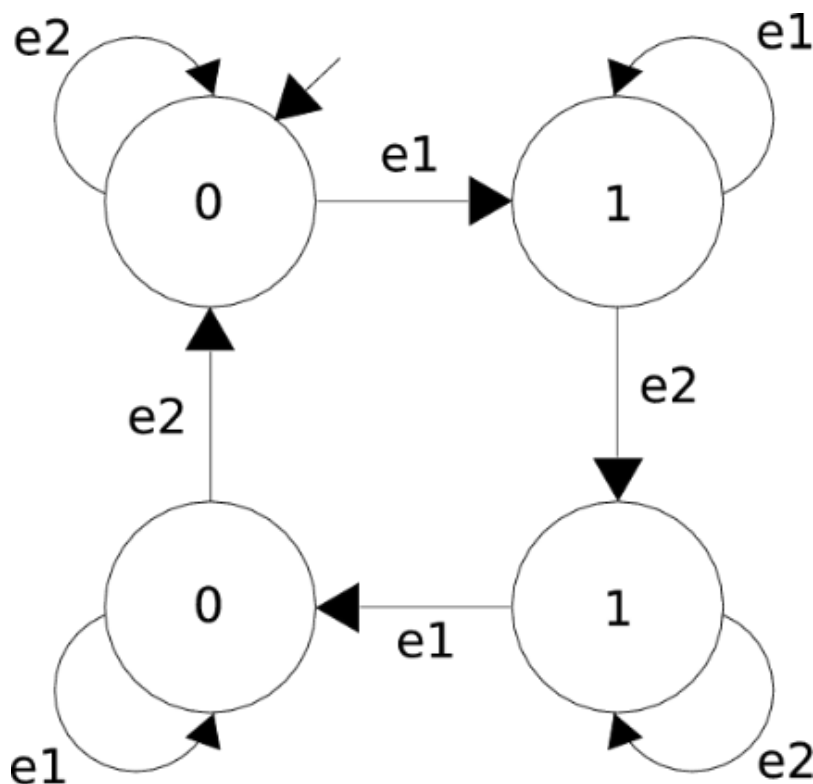


Рис.1 Граф представления переходов конечного автомата [8]

Полученный конечный автомат описывает поведение объекта намного реалистичней, чем первоначальный. Но, тем не менее, он еще далек от идеала.

Hammerman и Goldberg изложили в работе [6] идею, позволяющую улучшить работу генетического алгоритма, особью популяции которого является конечный автомат. Эта идея базируется на следующем наблюдении: существует несколько представлений одного и того же автомата в виде хромосомы (строки фиксированной длины). Например, автоматы на рис.2 изоморфны друг другу (обладают одинаковым поведением), но их записи в виде хромосом различаются. Таким образом, путем перенумерации вершин может быть получено $n!$ изоморфных автоматов, кодируемых различными хромосомами.

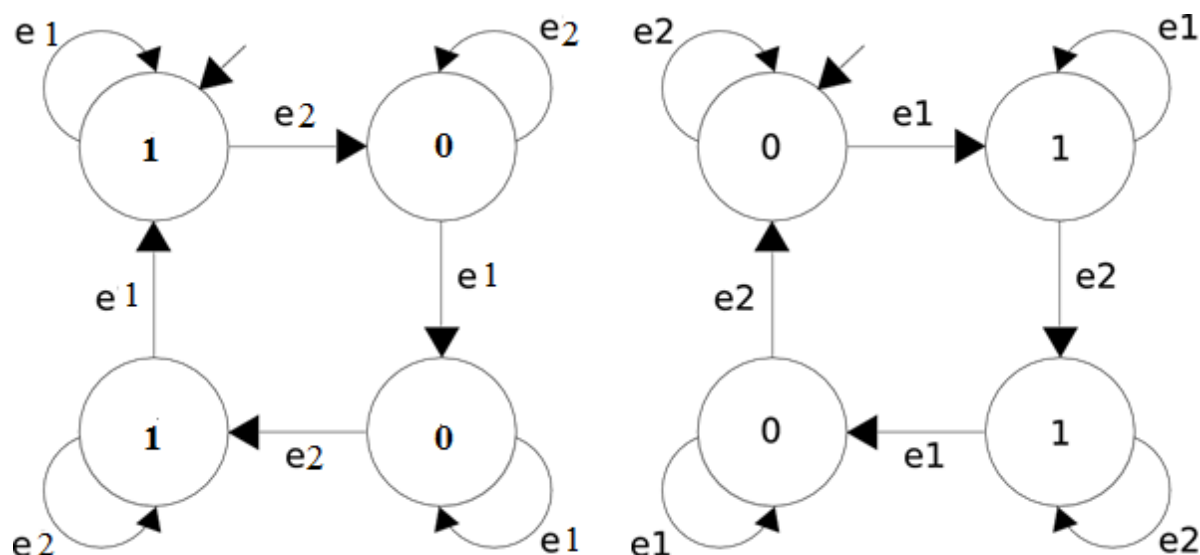


Рис. 2 Изоморфные автоматы

Опираясь на «теорему о схемах» (англ. – schemata theorem) [7] (поясняет почему генетические алгоритмы являются эффективным методом решения оптимизационных задач), авторы [6] утверждают, что различные схемы изоморфных автоматов «соревнуются» друг с другом, замедляя работу генетического алгоритма. Для того, чтобы устранить этот эффект, необходимо приводить автоматы к некоторому каноническому виду (единому для всех изоморфных автоматов), а уже затем кодировать его в виде хромосом. Для реализации этой идеи существует два алгоритма – MTF и SFS.

Таким образом, использование конечных автоматов в виде особой генетического алгоритма представленных графами переходов даст наилучшие результаты. Добавление вероятностного описания поведения объекта – возможное улучшение метода. Количество входных воздействий останется прежним, но потребуются затраты памяти для того, чтобы переоценить по формуле Байеса вероятности выбора действия.

Список литературы

1. Люгер, Дж. Ф. Искусственный интеллект: стратегии и методы решения сложных проблем / Люгер Дж. – М.: Издательский дом «Вильямс», 2003. – 863с.
2. Шампандар, А. Дж. Искусственный интеллект в компьютерных играх / Шарпандэр А. – М.: Издательский дом «Вильямс», 2007. – 768с.

3. Курейчик, В. М. Генетические алгоритмы и их применение / Курейчик В. – Таганрог: Издательство ТРТУ, 2002. – 242с.
4. Панченко, Т.В. Генетические алгоритмы [Текст]: учебно-методическое пособие / Панченко Т. – Астрахань: Издательский дом «Астраханский университет», 2007. – 87с.
5. Яхьева, Г.Э. Нечеткие множества и нейронные сети: Учебное пособие / Яхьева Г.Э. – М.: Интернет-университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2006. – 316с.
6. Chambers L. D. Practical Handbook of Genetic Algorithms, Volume 3, and Chapter 6 – Algorithms to Improve the Convergence of a Genetic Algorithm with a Finite State Machine Genome. CRC Press, 1999.
7. Mitchell M. Introduction to Genetic Algorithms. MA: MIT, 1999.
8. Поликарпова, Н. И. Автоматное программирование / Поликарпова Н. И. – М.: С-Пб.: Питер, 2009.