

**Заключение**

Таким образом, применение модифицированного генетического алгоритма к задаче об «итерированной дилемме узника» позволило получить особей, способных успешно соревноваться не только со своими собратьями, но и с неизвестными им противниками, что было показано на примере стратегии «Око за око». Исходя из этих результатов, можно сделать вывод о том, что предложенный метод является эффективным решением проблемы, описанной в начале статьи.

**Литература**

1. Поликарпова Н.И., Точилин В.Н., Шальто А.А. Разработка библиотеки для генерации автоматов методом генетического программирования // Сборник докладов X международной конференции по мягким вычислениям и измерениям. – СПб: СПбГЭТУ, 2007. – Т. 2. – С. 84–87.
2. Bäck T. Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms. – Oxford University Press, 1996.
3. Axelrod R., Hamilton W.D. The evolution of cooperation // Science. – 1981. – V. 211. – № 4489. – P. 1390.
4. Axelrod R. The evolution of strategies in the iterated prisoner's dilemma. – Cambridge university press, 1987. – P. 1–16.

*Кулев Владимир Анатольевич*

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, me@lightoze.net

УДК 004.4'242

**ПРИМЕНЕНИЕ ГЕНЕТИЧЕСКОГО ПОДХОДА ДЛЯ ГЕНЕРАЦИИ  
КЛЕТОЧНЫХ АВТОМАТОВ**

**А.В. Тихомиров, А.А. Шальто**

Рассматривается метод генерации произвольных клеточных автоматов на основе тестовых наборов при помощи генетических алгоритмов. Описаны основные проблемы при использовании стандартного генетического алгоритма для решения поставленной задачи. Предложены модифицированные генетические операторы для устранения данных недостатков. Произведена апробация на нескольких обучающих примерах.

**Ключевые слова:** клеточные автоматы, генетические алгоритмы.

**Введение**

В настоящее время широко распространено использование клеточных автоматов для симуляции многих физических процессов, таких как диффузия энергии, разнообразные химические реакции, рост кристаллов и т.д. [1–4]. Однако использование клеточных автоматов затрудняется тогда, когда физическая система известна, а описывающий ее клеточный автомат – нет, или построение его обычными эвристическими методами затруднительно, так как автомат может иметь большое число состояний, переходов, условий и действий на переход [5].

Цель настоящей работы – устранить этот недостаток, используя генетическое программирование с использованием обучающих наборов.

**Постановка задачи**

Задана двумерная плоскость определенной размерности с координатной сеткой, которая делит плоскость на квадраты [6, 7]. При этом задаются данные для различных временных шагов, т.е. состояние системы в начале, несколько промежуточных состояний и конечное состояние системы.

Каждая ячейка плоскости управляется одинаковым автоматом.

Цель поставленной задачи – вырастить клеточный автомат, который наиболее точно описывает заданную физическую систему.

**Строение хромосомы клеточного автомата**

Обычно при использовании генетического подхода используется подход, при котором данные в хромосоме хранятся в виде строки [5]. Однако в рассматриваемой задаче длина хромосомы может варьироваться в зависимости от количества переходов между состояниями. Для более удобного описания данных ген представляется не строкой, а специализированной структурой, что отличается от стандартного метода генетического программирования. При этом некоторые виды генов являются составными, т.е. содержат другие, более простые составляющие.

Количество базовых генов соответствует количеству состояний в клеточном автомате. Таким образом, базовый ген под названием «StateGene» описывает определенное состояние автомата. «StateGene» состоит из генов переходов клеточного автомата, которые называются «TransitionGene» (рис. 1).

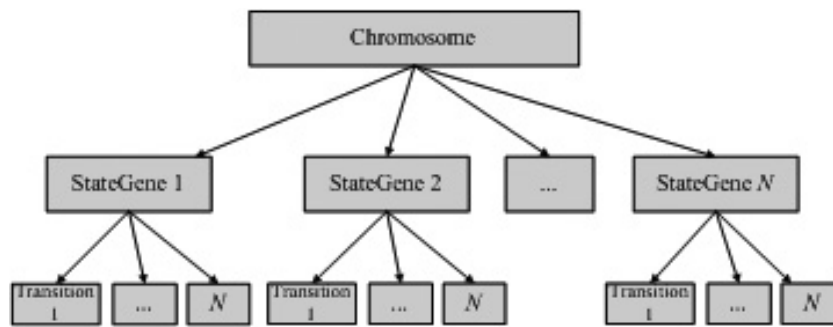


Рис. 1. Структура хромосомы

Каждый ген перехода состоит из списка генов условий на переход и генов действий, которые будут осуществляться в случае этого перехода. В случае отсутствия гена перехода из состояния  $N1$  в  $N2$ , это означает, что в клеточном автомате, описываемом заданным геном нет перехода из состояния  $N1$  в  $N2$  (рис. 2). Иначе говоря, длина хромосомы зависит от количества состояний в клеточном автомате, который описывает данная хромосома.

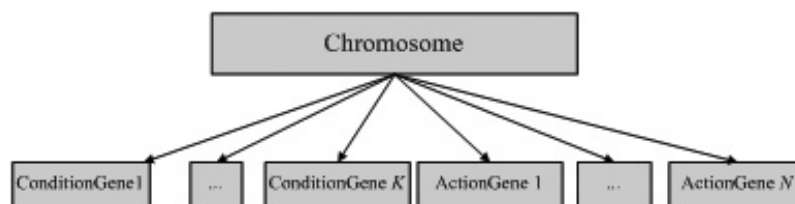


Рис. 2. Структура гена переходов

Условия переходов бывают двух видов:

1. Условия наличия рядом с текущей клеткой определенного числа  $N$  клеток с определенными состояниями. Этот вид условия задается в виде специальной строки;
2. Условия соотношения данных между текущей клеткой и определенной из окружающих клеток. Этот вид условия задается при помощи специального дерева, в верхнем узле которого находится операция сравнения, а ниже по дереву – операции вычисления условия.

В случае отсутствия генов условий подразумевается, что условие на данный переход всегда является верным.

Начальное поколение состоит из фиксированного числа случайно генерированных автоматов. Все автоматы в поколении имеют одинаковое наперед заданное количество состояний, но количество переходов между состояниями может варьироваться.

### Проблема вырождения популяции

Одной из проблем в применении генетического подхода для генерации клеточных автоматов стало вырождение популяции, т.е. такой ситуации, когда выделяется один-единственный генотип, который представляет собой локальный максимум, а затем все элементы популяции проигрывают ему отбор, и вся популяция «забывается» копиями этой особи.

Из-за схожего строения хромосом операция скрещивания становится малоэффективной, и вероятность получить хромосому, которая лучше решает поставленную задачу, сильно падает. Следовательно, для сокращения времени работы алгоритма надо как-то бороться с этой проблемой. В данной работе рассматриваются несколько способов преодолеть этот недостаток:

- модифицированный алгоритм скрещивания хромосом;
- алгоритм каскадной фитнес-функции;
- специальный алгоритм отбора нового поколения.

Все эти модификации позволяют улучшить «сходимость» генетического подхода при генерации клеточных автоматов.

### Операции скрещивания

В данной работе операция скрещивания была реализована в двух различных вариантах, чтобы избежать вырождения популяции.

Для решения этой проблемы при генерации начального поколения создается отдельная пустая популяция «Graveyard», которая не участвует в генетических операциях. В эту популяцию добавляются

особи, которые не попали в следующее поколение эволюции, при этом размер контролируется путем выбора произвольных  $n$  особей. Такой подход позволяет всегда иметь разнообразие в особях.

Операции скрещивания выполняется в двух вариантах (рис. 3):

- стандартная операция скрещивания, действие которой заключается в обмене генами между особями в текущей популяции;
- операция скрещивания между особью из текущей популяции и из популяции «Graveyard», причем право на участие в отборе имеет особь, основа которой была в популяции, участвующей в эволюции.

Каждая операция скрещивания состоит из двух вариантов:

1. Обмен несколькими «State» генами между особями, т.е. обмен состояниями;
2. Обмен одним из составных генов между особями, т.е. обмен переходами, условиями и действиями на переход. Причем возможен обмен генами, которые находятся в разных «State» генах.

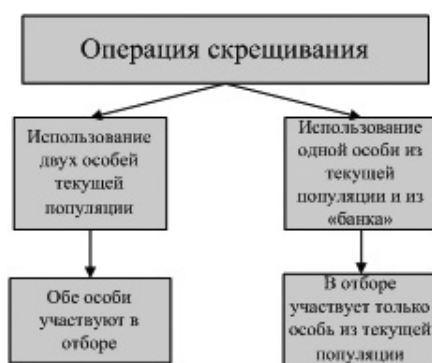


Рис. 3. Схема операции скрещивания

Если гены для скрещивания у особей являются идентичными, то операция скрещивания не совершается.

Для оптимизации операции скрещивания применяется анализ хромосом на совпадающие гены, т.е. из каждой пары хромосом, участвующих в скрещивании, выделяется список уникальных генов. Именно среди этих генов и производится поиск для обмена. Такая операция позволяет совершать меньшее число мутаций и заранее исключать мутации, которые приведут к хромосомам, которые уже есть в популяции.

### Операция мутации

Операция мутации состоит в том, что происходит изменение в произвольном гене копии текущей особи.

При мутации случайно выбирается один из четырех равновероятных вариантов:

1. Добавление нового перехода или полное замещение уже существующего;
2. Удаление уже существующего перехода;
3. Мутация существующего перехода

Мутация действий

- изменение действия на переходе;
- добавление случайного действия;
- удаление уже существующего действия.

Мутация условий

- изменения условия на переходе;
- добавление случайного условия;
- удаление уже существующего условия;

4. Обмен двух переходов местами.

### Фитнесс-функция

Функция приспособленности особи, т.е. клеточного автомата, состоит из нескольких значений. Первое значение характеризует автомат с точки зрения решения поставленной задачи, при моделировании сравниваются эталонные и полученные результаты для каждого шага моделирования. Второе значение характеризует автомат с точки зрения избыточности, в нем хранится «длина автомата» и количество неиспользуемых переходов и условий. Функция приспособленности особи вычисляется при моделировании и запоминается, таким образом, для каждой особи она вычисляется один раз. Для борьбы с вырождением популяции была применена каскадная фитнес-функция. Основная идея заключается в том, что

на начальных этапах выращивания клеточных автоматов функция приспособленности учитывает только несколько первых временных срезов входных данных. Когда в популяции есть клеточные автоматы, которые в какой-то степени удовлетворяют текущей фитнес-функции, то происходит смена на функцию с учетом большего количества слоев.

Ниже рассмотрены плюсы и минусы при использовании каскадной фитнес-функции:

Минусы:

- дополнительные затраты при смене функции;
- сложность сравнения эффективности особей (особенно это проявляется при сравнении особей из разных «островов»).

Плюсы:

- прирост производительности (на ранних этапах уменьшается количество этапов проверки клеточного автомата);
- после каждого этапа полученный автомат частично решает поставленную задачу.

Прирост скорости генерации (количества поколений, необходимых для получения решения) сильно зависит от начальных настроек при этом подходе.

### **Отбор нового поколения**

В результате операций скрещивания и мутации возможно получение уже содержащихся хромосом, что приводит к вырождению популяции. Для решения этой проблемы авторами предложена дополнительная операция, которая удаляет из претендентов на попадание в новый этап эволюции определенные клеточные автоматы.

В данной работе были рассмотрены два варианта:

1. Одинаковые клеточные автоматы. Для этого производится приведение всех клеточных автоматов к единой форме;
2. Клеточные автоматы с одинаковым значением фитнес-функции.

Чтобы избежать локальных минимумов функции приспособленности и вырождения популяции, применяются два алгоритма отбора в новое поколение.

1. Происходит отбор первых  $N$  особей с лучшей функцией приспособленностью. Все они попадают в новое поколение.
2. Если на протяжении достаточно большого числа поколений не происходит улучшения значения функции приспособленности, то при обработке текущего поколения отбрасываются все особи, кроме нескольких, которые имеют наилучшее значение, т.е. наиболее приспособленных. Оставшееся место в новом поколении занимают особи, полученные из особей текущего поколения путем скрещивания и мутаций. Также изменяется механизм мутации поколения. Мутация может происходить не в единственном гене хромосомы, как происходит при обычных условиях, а во всех генах хромосомы с некоторой вероятностью.

Таким образом, в случае успешного процесса генерации используется первый алгоритм, а в случае долгих «простоев», когда генетический алгоритм не может сгенерировать хромосому, которая лучше решает поставленную задачу, в работу вступает второй алгоритм отбора. Он благодаря большому числу мутаций и скрещиваний имеет больше шансов выйти из «простоя».

Также стоит отметить, что при отборе первых  $N$  особей при одинаковой функции приспособленности выбираются более молодые особи.

### **Полученные результаты**

Для тестового примера была поставлена задача: вырастить клеточный автомат, зная только состояния клеток на каждом конкретном шаге расчетов. Эталонный клеточный автомат, сделанный для проверки, содержит 5 состояний, 8 переходов и 8 условий на переходах. Система в среднем выращивает автомат, который является эквивалентным искомому за 700–1000 поколений, при этом среднее количество перебранных клеточных автоматов составляет 30000–45000.

Для ускорения процесса генерации было сделано допущение, что автоматы, у которых из какого-либо состояния истинно условие у нескольких переходов, являются корректными. На них было наложено условие: при нескольких вариантах перехода переход осуществляется в состояние с меньшим индексом. После упрощения полученного клеточного автомата и сравнения его с эталоном было установлено, что эти автоматы являются идентичными.

Как видно из примера, получившийся клеточный автомат содержит избыточные переходы, по которым никогда не происходит перехода. Однако при исследовании было установлено, что если задавать условие минимизации в генетических операторах и при выборе нового поколения, то время работы системы сильно увеличивается.

Быстродействие предложенного способа сильно зависит от начальных настроек, таких как порции между генетическими операциями, величина начальной популяции и многое другое.

**Заключение**

В работе предложен метод генерации клеточных автоматов произвольного количества состояний. Он позволяет автоматически получать клеточные автоматы, которые довольно точно описывают данные о физической системе на входе. Также этот метод позволяет получать клеточные автоматы, которые получить эвристическими методами затруднительно.

**Литература**

1. Тоффли Т., Марголус Н. Машины клеточных автоматов. – М.: Мир, 1991.
2. Frish U. Lattice gas hydrodynamics in two and three dimensions // Complex Systems. – 1987. – V. 1. – P. 649–707.
3. Wolfram S. Cellular automation Fluids // J.Stat.Phys. – 1986. – V. 45. – P. 471–526.
4. Фон Нейман Дж. Теория самовоспроизводящихся автоматов: Пер. с англ. – М.: Мир, 1971.
5. Царев Ф.Н., Шалыто А.А. Применение генетического программирования для генерации автомата в задаче об «Умном муравье» // Сборник трудов IV-ой Международной научно-практической конференции. – М.: Физматлит, 2007. – Т. 2. – С. 590–597.
6. Наумов Л.А. Метод введения обобщенных координат и инструментальное средство для автоматизации проектирования программного обеспечения вычислительных экспериментов с использованием клеточных автоматов. Дис...канд. техн. наук: 05.13.12. – СПб, 2007. – 283 с.
7. Скаков П.С. Классификация поведения одномерных клеточных автоматов [Электронный ресурс]. – Режим доступа: [http://is.ifmo.ru/papers/\\_skakov\\_master.pdf](http://is.ifmo.ru/papers/_skakov_master.pdf), своб.
8. Wolfram S. A New Kind of Science. – Wolfram Media, Inc., 2002.

- Тихомиров Андрей Владимирович* – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, студент, [and.tikhomirov@gmail.com](mailto:and.tikhomirov@gmail.com)
- Шалыто Анатолий Абрамович* – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, доктор технических наук, профессор, зав. кафедрой, [shalyto@mail.ifmo.ru](mailto:shalyto@mail.ifmo.ru)

УДК 004.4\*242

**ПРИМЕНЕНИЕ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ К ГЕНЕРАЦИИ ТЕСТОВ  
ДЛЯ АВТОМАТНЫХ ПРОГРАММ**

**А.Ю. Законов, А.А. Шалыто**

Описан подход к автоматизации тестирования автоматных программ. Для формализации требований спецификации к модели и объектам управления предлагается использовать контракты. Тест описывается как последовательность переходов в модели. Для автоматизации процесса создания кода теста предложен генетический алгоритм, который позволяет находить значения переменных, удовлетворяющие условиям на переходах.

**Ключевые слова:** тестирование, автоматное программирование, контракты, генетические алгоритмы.

**Введение**

Автоматная программа состоит из конечных автоматов и набора объектов управления, с которыми взаимодействуют автоматы [1]. Наиболее распространенным способом проверки автоматных программ является Model Checking [2], так как для этого класса программ уровень автоматизации процесса верификации может быть повышен по сравнению с традиционными программами. Это объясняется тем, что в автоматных программах поведенческая модель задается априори, а не строится по программе, как это делается обычно. Однако проверка моделей позволяет верифицировать только автоматы, но не систему в целом. Поведение объектов управления и их взаимодействие с автоматами не проверяются при указанном подходе. Таким образом, в автоматной программе могут остаться невыявленные ошибки, даже если система автоматов была успешно верифицирована относительно данной спецификации.

В данной работе используется тестирование для проверки автоматных программ в целом. Тестирование является трудоемкой и ресурсоемкой задачей. Около половины всего времени разработки проекта часто тратится на тестирование [3]. В последнее время многие исследования посвящены теме автоматизации процессов создания и запуска тестов [4]. Несмотря на то, что успешное тестирование не гарантирует отсутствие ошибок в программе, большой набор тестов может существенно помочь в обнаружении ошибок в логике работы программы и в ее реализации. Это может обеспечить повышение вероятности того, что программа будет успешно решать поставленные задачи.

В данной работе предлагается подход к тестированию автоматных программ и способ автоматизации процесса создания тестов на основе использования генетических алгоритмов.