

УДК 004.4*242

ПРИМЕНЕНИЕ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ ДЛЯ ПОСТРОЕНИЯ АВТОМАТОВ МУРА И СИСТЕМ ВЗАИМОДЕЙСТВУЮЩИХ АВТОМАТОВ МИЛИ НА ПРИМЕРЕ ЗАДАЧИ ОБ «УМНОМ МУРАВЬЕ»

А. А. Давыдов, Д. О. Соколов, Ф. Н. Царев

В работе предлагается генетический алгоритм, осуществляющий построение конечных автоматов Мура. Аналогичный алгоритм предлагается применять и для построения систем взаимодействующих автоматов Мили. Применение этих алгоритмов иллюстрируется на примере задачи об «Умном муравье».

Введение

В последнее время все чаще применяется автоматное программирование [1], в рамках которого поведение программ описывается с помощью конечных детерминированных автоматов. В ряде задач автомат удается построить эвристическими методами, однако часто такое построение требует больших затрат времени. В ряде случаев автоматы эвристически и вовсе не построить. Примером такой задачи является задача об «Умном муравье» [2]. Даже для этой относительно простой задачи полный перебор крайне трудоемок. Поэтому для построения автоматов в задачах такого рода целесообразно применять генетические алгоритмы [2–9].

Все известные авторам работы в этой области [10] посвящены построению одного автомата Мили. Однако в автоматном программировании допускается использование автоматов Мура и систем взаимодействующих автоматов [1]. При этом отметим, что одним из основных способов взаимодействия автоматов является их вложенность.

Целью настоящей работы является построение с помощью генетического программирования [7] пары вложенных автоматов Мили и автомата Мура для задачи об «Умном муравье».

Постановка задачи

Ниже приведено описание рассматриваемой задачи [2]. Игра происходит на поверхности тора размером 32×32 клетки (рис. 1).

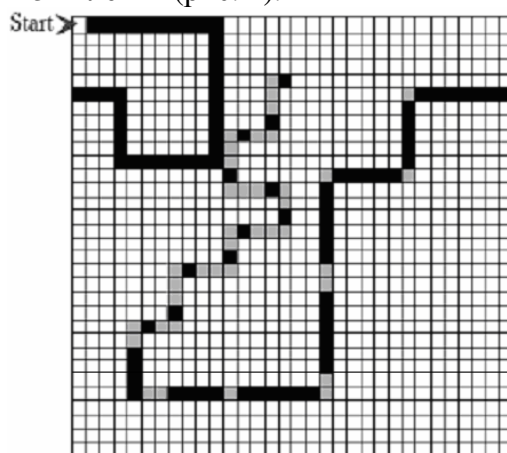


Рис. 1. Игровое поле

В некоторых клетках находится еда (на рисунке эти клетки обозначены черным цветом). Муравей начинает движение из клетки, помеченной меткой *Start*.

За ход муравей может выполнить следующие действия:

- повернуть налево;
- повернуть направо;
- сделать шаг вперед, и если в новой клетке есть еда, то съесть ее;
- ничего не делать.

Игра длится 200 шагов. Цель игры – создать муравья «с минимальным числом состояний», который за минимальное число шагов съест как можно больше яблок.

Ранее эта задача была решена [3] при помощи автомата Мили без использования взаимодействующих автоматов. Для построения таких автоматов в работе [8] применялись генетические алгоритмы.

В настоящей работе рассматриваемая задача решается сначала с помощью системы из двух автоматов Мили, взаимодействующих за счет вложенности, а затем – с помощью автомата Мура.

Алгоритм генетического программирования

Для генерации пары вложенных автоматов Мили и автомата Мура предлагается использовать один и тот же *островной генетический алгоритм* [5, 9], но с различными структурами хромосом, операторами мутации и скрещивания и функциями приспособленности. Отметим, что при необходимости этот алгоритм может быть адаптирован и для других типов систем автоматов.

В первом случае особь представляется собой пару вложенных автоматов Мили, один из которых будем называть внешним, а другой – внутренним. Хромосома особи состоит из описаний двух автоматов, каждое из которых содержит номер начального состояния и описания каждого состояния.

Описание состояния содержит описание двух переходов – перед муравьем есть еда, и ее нет. Для внешнего автомата любой из этих переходов может быть не определен. В такой ситуации переход совершается во вложенном автомате, у которого, в свою очередь, определены все переходы. Описание перехода состоит из номера состояния, в которое он ведет, и действия, выполняемого при выборе этого перехода.

Во втором случае особь представляет собой автомат Мура. Хромосома особи состоит из номера начального состояния и описания каждого состояния. Описание состояния содержит действие, выполняемое при переходе в состояние, и описание двух переходов – перед муравьем есть еда, и ее нет. Описание перехода состоит из номера состояния, в которое он ведет.

Хромосома не кодируется битовой строкой, как это делается традиционно, а представляет собой объект на языке *Java* вида:

```
class Automaton {
    int initialState;
    Transition[][] transition;
    char[] stateAction[]; // только для автоматов Мура
    Automaton nestedAutomaton;
}
```

Островной алгоритм традиционно состоит из следующих этапов:

- создание начального поколения;
- мутация;
- скрещивание (кроссовер);

- вычисление функции приспособленности (фитнес-функции);
- обмен особями между островами;
- отбор особей для формирования следующего поколения.

Опишем этапы этого алгоритма применительно к рассматриваемой задаче. При этом сначала опишем этапы, которые являются общими для обеих моделей описания поведения муравья.

Создание начального поколения

Все острова заполняются случайно сгенерированными автоматами. Все автоматы имеют заранее заданное число состояний.

Мутация (малая или мутация особи)

Каждое поколение особей с заданной вероятностью может мутировать.

Для пары вложенных автоматов Мили это означает:

- случайное изменение стартового состояния особи;
- выбор случайного перехода и случайное изменение состояния, в который он ведет;
- удаление (добавление) случайного перехода для внешнего автомата;
- изменение действия на случайном переходе;
- мутация вложенного автомата.

Для автоматов Мура применяется аналогичный оператор мутации:

- случайное изменение стартового состояния особи;
- выбор случайного перехода и случайное изменение состояния, в который он ведет;
- изменение действия в случайном состоянии.

Каждый из перечисленных действий выполняется с некоторой априори заданной вероятностью.

Мутация (большая или мутация острова)

Через заранее заданное число поколений фиксированная доля островов заменяется островами со случайными особями. Проведение мутации в момент, когда функция приспособленности (имеются в виду только элитные особи) изменяется незначительно, невозможно, так как за счет миграции особей между островами фитнес-функция постоянно изменяет свое значение.

Скрещивание

Оператор скрещивания получает на вход две особи и выдает две особи (потомки входных особей). Оператор скрещивания пары вложенных автоматов аналогичен описанному в работе [8]. Вложенные автоматы равновероятно либо скрещиваются, либо один ребенок наследует вложенный автомат одного родителя, а другой – другого.

Оператор скрещивания автоматов Мура также аналогичен описанному в работе [8], но имеет особенность в связи с наличием действия в состоянии. Обозначим родительские особи – $P1$ и $P2$, а детей – $S1$ и $S2$. Обозначим действие в k -ом состоянии автомата A , как $A.a[k]$. Тогда для любого k будет верно одно из следующих утверждений:

- $S1.a[k] = P1.a[k]$, $S2.a[k] = P2.a[k]$;
- $S1.a[k] = P2.a[k]$, $S2.a[k] = P1.a[k]$;
-

Оба этих варианта равновероятны.

Вычисление функции приспособленности

В настоящей работе для генерации автоматов Мура применяется функция приспособленности вида

$$F - \frac{T}{200},$$

где F – количество съеденной еды, T – номер последнего хода, на котором муравей съел еду. Как показали вычислительные эксперименты, такая функция не подходит для генерации пары взаимодействующих автоматов Мили, так как при ее использовании генерируются слабо определенные внешние автоматы, функциональность которых, в основном, состоит в передаче управления вложенному автомату. Поэтому для пары вложенных автоматов Мили применялась функция приспособленности вида

$$F - \frac{T}{200} + C \cdot Z,$$

где F – количество съеденной еды, T – номер последнего хода, на котором муравей съел еду, Z – число посещенных состояний у внешнего автомата, C – некоторый коэффициент. Подбор коэффициента C является ключевым при генерации средствами генетического программирования систем автоматов, взаимодействующих за счет вложенности автоматов.

Формирование следующего поколения

В качестве основной стратегии формирования следующего поколения используется элитизм [11]. При рассмотрении текущего поколения отбрасываются все особи, кроме некоторой доли наиболее приспособленных – элиты. Эти особи переходят в следующее поколение. После этого оно дополняется в определенной пропорции случайными особями, мутировавшими особями из текущего поколения и результатами скрещивания особей из текущего поколения. Особи, «имеющие право» давать потомство, определяются «в поединке»: выбираются две случайные пары особей, и более приспособленная в каждой паре становится одним из родителей. Эта эволюция происходит независимо на каждом из островов.

Через фиксированное число поколений каждый остров меняется с другим случайным числом выбранных случайно элитных особей. Также через некоторое число поколений происходит большая мутация.

В процессе разработки алгоритма исследовалась также возможность использования еще одного средства отбора – «войны между островами», в которой победитель (например, остров с большим суммарным значением функции приспособленности) истребляет население проигравшего (возможно, кроме элиты) и заселяет остров своими «жителями». Однако это не ускоряет общую эволюцию, приводя к уменьшению разнообразия живущих особей.

Настраиваемые параметры генетического алгоритма

Алгоритм имеет следующие настраиваемые параметры:

- число островов;
- размер поколения на одном острове;
- число поколений между большими мутациями;
- доля уничтожаемых островов во время большой мутации;

- процент элиты;
- число поколений между обменом особями между островами;
- число обмениваемых особей;
- параметры малой мутации;
- параметры скрещивания;
- отношение «мутантов», случайных особей и детей особей из текущего поколения при формировании следующего поколения;
- C – коэффициент влияния внешнего автомата.

Соотношение этих параметров, при котором достигается оптимальный результат, неизвестно, однако при генерации систем автоматов существенное значение имеет коэффициент влияния внешнего автомата C в функции приспособленности.

Результаты

С помощью разработанного алгоритма построен автомат Мура с 10 состояниями, который позволяет муравью съесть всю еду за 198 шагов (рис. 2).

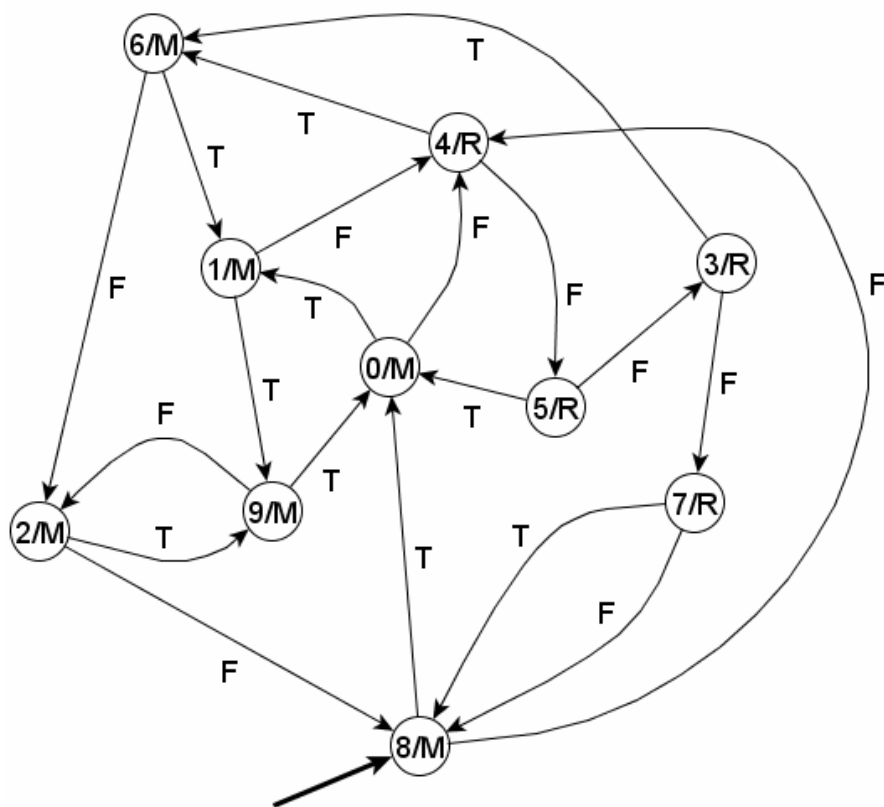


Рис. 2. Автомат, позволяющий муравью съесть всю еду за 198 шагов

Поясним условные обозначения на рис. 2. Пометки на вершинах имеют вид: номер / действие. Действия обозначаются следующим образом:

- L – поворот налево;
- R – поворот направо;
- M – сделать шаг, и если в новой клетке есть еда, то съесть ее.

Условия на переходах обозначаются следующим образом:

- T – перед муравьем есть еда;
- F – перед муравьем нет еды.

Начальное состояние отмечено внешней стрелкой.

Лучшим достигнутым авторами результатом для *девяти состояний* является автомат, который позволяет муравью съесть 86 единиц еды также за 198 ходов.

Для взаимодействующих автоматов Мили не удалось построить пару, которая позволяет муравью съесть всю еду, и каждый автомат из которой имеет меньше семи состояний, как это имеет место в случае использования одного автомата Мили [8].

Лучшей построенной авторами системой является система автоматов (4, 6) (четыре состояния во внешнем автомате, шесть – во внутреннем), позволяющая муравью съесть 87 единиц еды за 185 ходов (рис. 3). Данный результат был получен при значении параметра $C = 0.01$. Из изложенного можно сделать вывод о том, что автомат, получаемый в результате решения задачи о муравье, не декомпозируем либо плохо декомпозируем.

Поясним условные обозначения на рис. 3. Пометки на вершинах имеют вид: номер, а на переходах: условие / действие. Условия, действия и стартовые состояния обозначаем так же, как и на рис. 2.

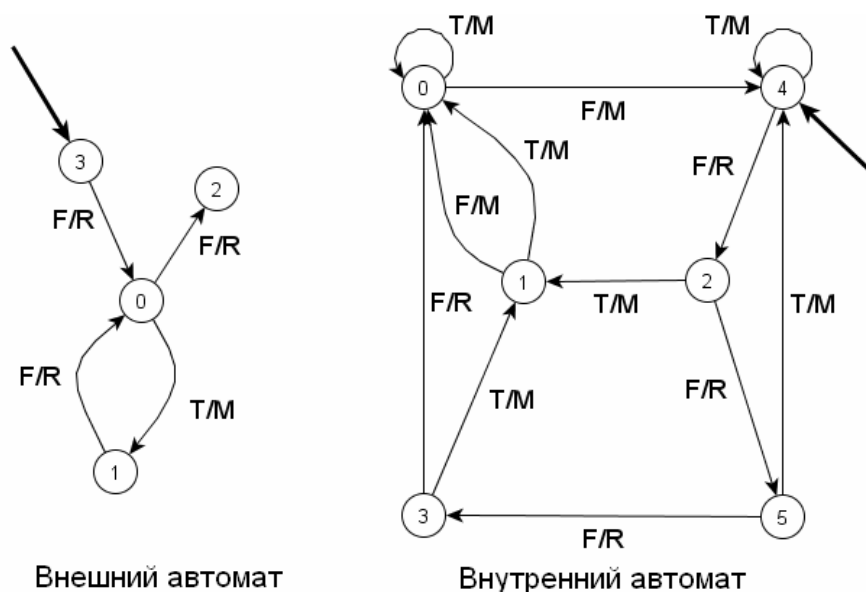


Рис. 3. Пример системы взаимодействующих автоматов Мили

Заключение

В работе предложен островной генетический алгоритм, осуществляющий построение конечных автоматов Мура и систем из двух взаимодействующих за счет вложенности автоматов Мили. Эти алгоритмы применены для задачи об «Умном муравье», и построен решающий эту задачу автомат Мура с десятью состояниями.

В дальнейшем возможно применение сокращенных таблиц [10], что должно позволить использовать предложенный алгоритм для решения задач с большим числом входных переменных.

Литература

1. Шалыто А. А. Технология автоматного программирования /Труды первой Всероссийской научной конференции «Методы и средства обработки информации». М.: МГУ, 2003. http://is.ifmo.ru/works/tech_aut_prog/

2. Jefferson D., Collins R., Cooper C., Dyer M., Flowers M., Korf R., Taylor C., Wang A. The Genesys System. 1992.
www.cs.ucla.edu/~dyer/Papers/AlifeTracker/Alife91Jefferson.html
3. Angeline P., Pollack J. Evolutionary Module Acquisition / Proceedings of the Second Annual Conference on Evolutionary Programming, 1993.
<http://www.demo.cs.brandeis.edu/papers/ep93.pdf>
4. Chambers L. Practical Handbook of Genetic Algorithms. Complex Coding Systems. Volume III. CRC Press, 1999.
[http://www.eknigu.com/info/Cs_Computer%20science/CsGn_Genetic,%20neural/Chambers%20D.L.%20\(ed.\)%20Vol.%203.%20Handbook%20of%20genetic%20algorithms.%20Complex%20coding%20systems%20\(CRC,%201999\)\(ISBN%200849325390\)\(T\)\(659s\).djvu](http://www.eknigu.com/info/Cs_Computer%20science/CsGn_Genetic,%20neural/Chambers%20D.L.%20(ed.)%20Vol.%203.%20Handbook%20of%20genetic%20algorithms.%20Complex%20coding%20systems%20(CRC,%201999)(ISBN%200849325390)(T)(659s).djvu)
5. Гладков Л. А., Курейчик В. В., Курейчик В. М. Генетические алгоритмы. – М.: Физматлит, 2006.
6. Рассел С., Норвиг П. Искусственный интеллект: современный подход. – М.: Вильямс, 2006.
7. Koza J. R. Genetic programming: on the programming of computers by means of natural selection. MA: MIT Press, 1992.
[http://www.eknigu.com/info/Cs_Computer%20science/CsGn_Genetic,%20neural/Koza%20J.R.%20Genetic%20programming%20\(MIT,%201998\)\(T\)\(ISBN%2000262111705\)\(609s\).djvu](http://www.eknigu.com/info/Cs_Computer%20science/CsGn_Genetic,%20neural/Koza%20J.R.%20Genetic%20programming%20(MIT,%201998)(T)(ISBN%2000262111705)(609s).djvu)
8. Царев Ф. Н., Шалыто А. А. О построении автоматов с минимальным числом состояний для задачи об «умном муравье» / Сборник докладов X международной конференции по мягким вычислениям и измерениям. СПбГЭТУ "ЛЭТИ". Т.2, 2007, с. 88–91. http://is.ifmo.ru/download/ant_ga_min_number_of_state.pdf
9. Яминов Б. Генетические алгоритмы
<http://rain.ifmo.ru/cat/view.php/theory/unordered/genetic-2005>
10. Поликарпова Н. И., Точилин В. Н., Шалыто А. А. Применение генетического программирования для реализации систем со сложным поведением // Научно-технический вестник СПбГУ ИТМО. 2007. Вып. 39, с. 276–293.
http://vestnik.ifmo.ru/ntv/39/ntv_39.3.3.pdf
11. De Jong K. An analysis of the behaviour of a class of genetic adaptive systems. PhD thesis. Univ Michigan. Ann Arbor, 1975.