

О ПОСТРОЕНИИ АВТОМАТОВ С МИНИМАЛЬНЫМ ЧИСЛОМ СОСТОЯНИЙ ДЛЯ ЗАДАЧИ ОБ «УМНОМ МУРАВЬЕ»

Ф.Н. Царев, А.А. Шалыто

Санкт-Петербургский государственный университет информационных технологий, механики и оптики

Abstract. Problem of construction automata for artificial ant on John Muir trail is studied. Genetic programming approach and brute force approach are used to construct an automaton with minimal states count.

ВВЕДЕНИЕ

В последнее время все шире применяется автоматное программирование, в рамках которого поведение программ описывается с помощью конечных детерминированных автоматов [1].

Для многих задач автоматы удается строить эвристически, однако существуют задачи, для которых такое построение затруднительно. К задачам этого класса относится, в частности, и задача об «Умном муравье» [2–4].

В этих работах генерация автоматов выполнялась с помощью генетических алгоритмов [5,6] на однопроцессорной ЭВМ. Однако построенные в указанных работах автоматы содержат большое число состояний.

Цель настоящей работы – построение (также с помощью однопроцессорной ЭВМ) автомата с минимизированным числом состояний для решения задачи об «Умном муравье».

ПОСТАНОВКА ЗАДАЧИ

Приведем описание задачи об «Умном муравье» [2]. Игра происходит на поверхности тора размером 32 на 32 клетки (рис. 1). В некоторых клетках (обозначены на рис. 1 черным цветом) находится еда. Она расположена вдоль некоторой ломаной, но не во всех ее клетках. Клетки ломаной, в которых нет еды, обозначены серым цветом. Белые клетки не содержат еду и не принадлежат ломаной. Всего на поле 89 клеток с едой.

В клетке, помеченной меткой «Start», в начале игры находится муравей. Он занимает клетку и может смотреть в одном

из четырех направлений (север, юг, запад, восток). В начале игры муравей смотрит на восток.

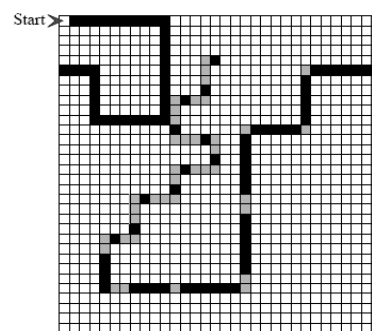


Рис. 1. Игровое поле

Муравей умеет определять находится ли непосредственно перед ним еда. За один ход муравей может совершить одно из четырех действий:

- сделать шаг вперед, съедая еду, если она там находится;
- повернуть налево;
- повернуть направо;
- ничего не делать.

Съеденная муравьем еда не пополняется, муравей жив на протяжении всей игры. Ломаная не случайна, а строго фиксирована. Муравей может ходить по любым клеткам поля.

Игра длится 200 ходов, на каждом из которых муравей совершает одно из четырех действий. По истечении 200 ходов подсчитывается количество еды, съеденной муравьем. Это значение и есть результат игры.

Цель игры – создать муравья, который за 200 ходов съест как можно больше еды.

Один из способов описания поведения муравья – конечный автомат с действиями на переходах (автомат Мили), у которого

О построении автоматов с минимальным числом состояний для задачи об «Умном муравье»

муравьем есть еда» как $P1(i, 1)$, а по значению «Перед муравьем нет еды» – как $P1(i, 0)$. Аналогичный смысл придадим обозначениям $P2(i, 0)$ и $P2(i, 1)$. Тогда для переходов из состояния с номером i в автоматах-потомках $S1$ и $S2$ справедливо:

- либо $S1(i, 0) = P1(i, 0)$,
 $S1(i, 1) = P2(i, 1)$ и $S2(i, 0) = P2(i, 0)$,
 $S2(i, 1) = P1(i, 1)$;
- либо $S1(i, 0) = P2(i, 0)$,
 $S1(i, 1) = P1(i, 1)$ и $S2(i, 0) = P1(i, 0)$,
 $S2(i, 1) = P2(i, 1)$;
- либо $S1(i, 0) = P1(i, 0)$,
 $S1(i, 1) = P1(i, 1)$ и $S2(i, 0) = P2(i, 0)$,
 $S2(i, 1) = P2(i, 1)$;
- либо $S1(i, 0) = P2(i, 0)$,
 $S1(i, 1) = P2(i, 1)$ и $S2(i, 0) = P1(i, 0)$,
 $S2(i, 1) = P1(i, 1)$.

Все четыре варианта равновероятны.

Второй способ скрещивания переходов. В автоматах $P1$ и $P2$ определим переходы, которые они выполняют в течение первых сорока ходов по игровому полю. Обозначим множество таких переходов автоматов $P1$ и $P2$ как $TF(P1)$ и $TF(P2)$ соответственно. Множество переходов некоторого автомата A обозначим $T(A)$. Возможны два равновероятных варианта:

- либо $T(S1) = TF(P1) \cup (T(P2) \setminus TF(P2))$ и
 $T(S2) = TF(P2) \cup (T(P1) \setminus TF(P1))$;
- либо $T(S2) = TF(P1) \cup (T(P2) \setminus TF(P2))$
и $T(S1) = TF(P2) \cup (T(P1) \setminus TF(P1))$.

Формирование следующего поколения. В качестве основной стратегии формирования следующего поколения используется элитизм [8]. При обработке текущего поколения отбрасываются все особи, кроме нескольких наиболее приспособленных. Доля выживающих особей постоянна для каждого поколения и является одним из параметров алгоритма.

Эти особи переходят в следующее поколение. После этого оно дополняется до требуемого размера следующим образом: пока оно не заполнено, выбираются две особи из текущего поколения, и они с некоторой вероятностью скрещиваются или мутируют. Обе особи, полученные в

результате мутации или скрещивания, добавляются в новое поколение.

Кроме этого, если на протяжении достаточно большого числа поколений не происходит увеличения приспособленности, то применяются «малая» и «большая» мутации поколения. При «малой» мутации поколения ко всем особям, кроме 10% лучших, применяется оператор мутации. При «большой» мутации каждая особь либо мутирует, либо заменяется на случайно сгенерированную.

Количество поколений до «малой» и «большой» мутации постоянно во время работы алгоритма.

Вычисление функции приспособленности. Функция приспособленности особи (автомата)

равна $F + \frac{200 - T}{200}$, где F – количество еды,

которое съедает за 200 ходов муравей, поведение которого задается этим автоматом, а T – номер хода, на котором муравей съедает последнюю единицу еды.

РЕЗУЛЬТАТЫ ПРИМЕНЕНИЯ ГЕНЕТИЧЕСКОГО ПРОГРАММИРОВАНИЯ

На рис. 3 изображен граф переходов, построенный описанным алгоритмом после генерации 160 млн. автоматов. Этот граф переходов описывает поведение автомата с семьёю состояниями, который позволяет муравью съесть всю еду за 190 ходов.

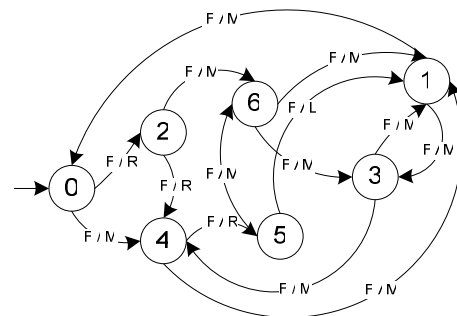


Рис. 3. Автомат, позволяющий муравью съесть всю еду

На рис. 3 используются те же обозначения, что и на рис. 2.

Отметим, что при другом запуске алгоритма после генерации 250 млн. автоматов был построен еще один автомат с семьёю состояниями.

Из изложенного следует, что применение генетического программирования для автоматов с семью состояниями позволяет сократить перебор в 10^{10} раз, так как при полном переборе пришлось бы рассмотреть $3 \cdot 10^{18}$ автоматов.

С помощью описанного алгоритма генетического программирования также были построены автоматы из пяти (рис. 4) и шести состояний, позволяющие съесть 83 и 85 единиц еды соответственно.

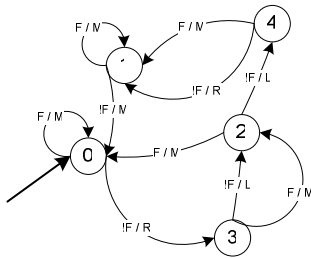


Рис. 4. Автомат из пяти состояний, построенный алгоритмом генетического программирования

ПРИМЕНЕНИЕ ПЕРЕБОРА ДЛЯ ПОСТРОЕНИЯ АВТОМАТОВ

Каждый автомат является ориентированным графом с пятью вершинами, одна из которых выделена. При этом каждой вершине инцидентны две дуги, одна из которых соответствует тому, что перед муравьем есть еда, а другая – тому, что перед муравьем нет еды. При этом каждая дуга помечена действием, которое муравей выполняет при выборе соответствующего перехода.

Заметим, что любой автомат, решающий задачу об «Умном муравье» можно преобразовать в автомат, который не содержит действий «Ничего не делать». Эти действия можно устранить алгоритмом, подобным алгоритму ϵ -замыкания [9].

Таким образом, количество автоматов с n состояниями можно оценить сверху как $(3n)^{2n}$, что при $n=5$ составляет 576 650 390 625. Так как помимо перебора такого количества автоматов, для каждого из них необходимо вычислить функцию приспособленности, то на выполнение этих вычислений на однопроцессорной ЭВМ требуется порядка 200 суток.

Анализ автоматов, построенных с помощью генетического программирования, показывает, что все они действуют по

принципу «вижу еду – иду вперед». Количество таких автоматов с n состояниями равно $n^n(3n)^n$. При $n=5$ это число составляет 2 373 046 875. Такое количество автоматов можно перебрать за время порядка двух суток.

В результате такого перебора был найден автомат с пятью состояниями, позволяющий муравью съесть 83 единицы еды за 182 хода – на шесть ходов быстрее, чем автомат, построенный методом генетического программирования.

ЗАКЛЮЧЕНИЕ

Генетическое программирование позволяет строить автоматы, близкие к оптимальным. При этом построение с помощью генетического программирования требует значительно меньше времени по сравнению с полным перебором. Вопрос о минимальном количестве состояний, необходимом для решения задачи об «Умном муравье», пока остается открытым.

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. J., Pollack J. Evolutionary Module Acquisition /Proceedings of the Second Annual Conference on Evolutionary Programming. 1993. <http://www.demon.cs.brandeis.edu/papers/ep93.pdf>
4. Chambers L. Practical Handbook of Genetic Algorithms. Complex Coding Systems. Volume III. CRC Press, 1999.
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. MIT Press, 1992.
8. De Jong K. An analysis of the behavior of a class of genetic adaptive systems. PhD thesis. Univ. Michigan. Ann Arbor, 1975.
9. Хопкрофт Д., Мотвани Р., Ульман Д. Введение в теорию автоматов, языков и вычислений. М.: Вильямс, 2002.