

Заключение

Исследования и эксперименты показали, что предложенный модифицированный метод отжига Коши оказался эффективнее как метода генетического программирования, так и других алгоритмов имитации отжига в рамках игры «Война за ресурсы». Таким образом, результатами работы подтверждена целесообразность применения метода отжига для построения конечных автоматов на примере игры «Война за ресурсы».

Исследование выполнено по Федеральной целевой программе «Научные и научно-педагогические кадры инновационной России на 2009–2013 годы» в рамках государственного контракта П2236 от 11 ноября 2009 года.

Литература

1. Spears W., Gordon D. Evolution of strategies for resource protection problems // Theory and Applications of Evolutionary Computation: Recent Trends. – Springer-Verlag, 2002.
2. Ingber L. Simulated Annealing: Practice Versus Theory // Mathl. Comput. Modelling. – 1993.
3. Лопатин А.С. Метод отжига // Стохастическая оптимизация в информатике. – 2005.
4. Поликарпова Н.И., Шальто А.А. Автоматное программирование. – СПб: Питер, 2009 [Электронный ресурс]. – Режим доступа: http://is.ifmo.ru/books/_book.pdf, своб.

Заикин Александр Константинович – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, студент, zaikin@rain.ifmo.ru

УДК 004.4'242

МЕТОД ПРЕДСТАВЛЕНИЯ АВТОМАТОВ ЛИНЕЙНЫМИ БИНАРНЫМИ ГРАФАМИ ДЛЯ ИСПОЛЬЗОВАНИЯ В ГЕНЕТИЧЕСКОМ ПРОГРАММИРОВАНИИ

В.Р. Данилов, А.А. Шальто

Предлагается метод представления автоматов в виде особой эволюционного алгоритма, основанный на использовании линейных бинарных графов. На примере выполнено сравнение этого метода с известными методами. Предлагаемый метод является более эффективным по сравнению с представлением функции переходов полными таблицами. При некоторых значениях числа состояний он более эффективен, чем метод представления функции переходов деревьями решений.

Ключевые слова: генетическое программирование, конечные автоматы, линейные бинарные графы.

Введение

Генетическое программирование [1] – метод автоматической генерации программ на основе эволюционных алгоритмов [2], использующий представление программ на высоком уровне абстракции. В работах [3, 4] рассматривалось применение генетического программирования для построения автоматов управления системами со сложным поведением. Автоматы такого рода характеризуются сложностью функции переходов из каждого состояния. Таким образом, методы, основанные на представлении функции переходов автоматов полными таблицами, оказываются неприменимыми на практике. Наиболее близким к предлагаемому в этой работе методу является метод представления автоматов, основанный на деревьях решений [4]. Недостатком такого подхода является повторное кодирование повторяющихся поддеревьев. В настоящей работе предлагается метод представления функции переходов автоматов управления, основанный на линейных бинарных графах [5], который лишен указанного недостатка.

Линейные бинарные графы

Разрешающая диаграмма [6] является удобным способом задания булевой функции. Она представляет собой помеченный ациклический ориентированный граф, в котором выделяют вершины двух типов:

- нетерминальные узлы;
- терминальные узлы.

При этом один из нетерминальных узлов является начальным. Все остальные узлы достижимы из начального. Из каждого нетерминального узла выходит по два ребра. Из терминальных узлов ребра не выходят. Метки в графе расставляются по следующим правилам:

- нетерминальные узлы помечаются названиями булевых переменных;
- терминальные узлы помечаются значениями функции;
- ребра помечаются значениями булевых переменных.

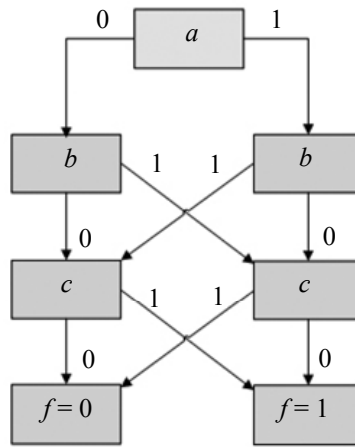


Рис. 1. Пример разрешающей диаграммы

Для определения значения функции по значениям булевых переменных необходимо пройти путь от стартового до терминального узла и определить значение, которым он помечен. При этом из вершины, помеченной булевой переменной x , переход производится по тому ребру, которое помечено тем же значением, что и значение переменной x . На рис. 1 приведен пример разрешающей диаграммы, реализующей булеву функцию $f = a \oplus b \oplus c$.

Важным частным случаем разрешающих диаграмм являются линейные разрешающие диаграммы или линейные бинарные графы. Узлы линейного бинарного графа могут быть пронумерованы таким образом, что из каждого нетерминального узла одно из ребер ведет в узел со следующим номером. При этом число элементов, необходимых для реализации булевой формулы бинарным линейным графом, линейно зависит от числа букв в формуле [5].

Пример линейного бинарного графа, реализующего булеву функцию $f = ab \vee c$, приведен на рис. 2.

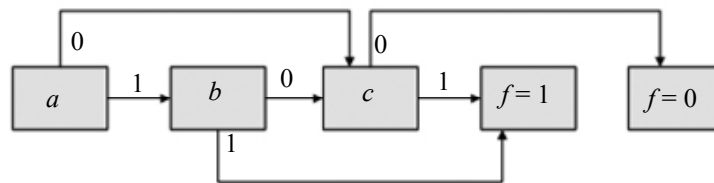


Рис. 2. Пример линейного бинарного графа

Представление автоматов бинарными линейными графами

Опишем предлагаемый метод представления автомата с помощью линейных бинарных графов. Для задания автомата необходимо выразить его функции переходов и действия. Осуществим следующее преобразование автомата: вместо задания функций переходов и действия для автомата в целом определим эти функции в каждом состоянии.

Более формально это выглядит следующим образом: зададим для каждого состояния $q \in Q$ функцию $\sigma_q : X \rightarrow Q \times Y$, такую, что $\sigma_q(x) = (\delta(q,x), \lambda(q,x))$ для $\forall x \in X$.

Функции σ_q соответствуют функциям переходов и действия в состоянии q . Каждая из таких функций может быть выражена с помощью некоторого линейного бинарного графа. Переменными этих графов являются входные переменные автомата, а множеством значений – все возможные пары (Новое Состояние, Действие). Таким образом, автомат в целом задается упорядоченным набором линейных бинарных графов.

Для использования представления автоматов в виде набора линейных бинарных графов в генетическом программировании необходимо определить генетические операции. Это может быть сделано с помощью генетических операций над линейными бинарными графами:

- случайное порождение автомата – в каждом состоянии создается случайный линейный бинарный граф;
- скрещивание автоматов – линейные бинарные графы в соответствующих состояниях скрещиваются;
- мутация автомата – в линейном бинарном графе, соответствующему случайному состоянию, производится мутация.

При этом считается, что число состояний в автомате фиксировано. Поэтому противоречий при выполнении определенных таким образом операций не возникнет.

Теперь определим представление линейных бинарных графов в виде хромосомы и соответствующие генетические операции. Линейный бинарный граф может быть представлен в виде хромосомы в следующей форме.

- Граф представляет собой упорядоченный список узлов. При этом считается, что узлы идут в таком порядке, что из каждого узла одно из выходящих ребер ведет в следующий узел.
- Для каждого нетерминального узла хранится переменная, которой помечен данный узел, значение переменной расщепления, соответствующее переходу в следующий узел, и номер узла, в который производится переход при ином значении переменной расщепления.
- Для каждого из терминальных узлов хранится значение функции, которым помечен соответствующий узел.

При этом терминальные узлы являются последними узлами линейного бинарного графа. Поэтому хромосому можно представить в виде строки, отдельно выделяя части, кодирующие терминальные и нетерминальные узлы.

Закодируем для примера в виде строки граф, приведенный на рис. 2. Заметим, что узлы этого графа могут быть пронумерованы в требуемом порядке слева направо. Тогда этому графу будет соответствовать следующая строка:

a13b04c15 10

Для скрещивания линейных бинарных графов можно применить любой из методов скрещивания строк, отдельно скрещивая части, описывающие нетерминальные и терминальные узлы. При этом часть, соответствующая терминальным узлам, должна быть скопирована полностью из одной родительской хромосомы.

Операция мутации может быть определена следующим способом. Выбирается произвольный нетерминальный узел. После этого в него вносится одно из следующих изменений:

- замена переменной, которой помечен модифицируемый узел;
- замена значения переменной, которым помечен переход в следующий в порядке нумерации узел;
- замена номера узла, в который производится переход при несовпадении значения переменной.

Заключение

Разработанный подход был протестирован на задаче «Умный муравей-2» [7]. Приведем описание задачи. Муравей находится на случайном игровом поле. Поле представляет собой тор размером 32×32 клетки. Муравей видит перед собой некоторую область (рис. 3).

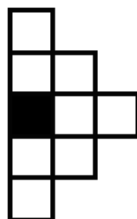


Рис. 3. Видимая муравью область

Еда в каждой клетке располагается с некоторой вероятностью μ . Значение μ является параметром задачи. Игра длится 200 ходов, за каждый из которых муравей может сделать одно из трех действий: поворот налево или направо, шаг вперед. Если после хода муравей попадает на клетку, где есть еда, то еда съедается. Целью задачи является построение стратегии поведения муравья, при которой математическое ожидание съеденной еды максимально.

Автомат управления муравьем в этой задаче имеет восемь (число видимых клеток) булевых входных переменных, каждая из которых определяет, есть ли еда в клетке, соответствующей переменной.

Эксперимент состоял в сравнении полученных значений фитнес-функции (объема съеденной еды) за фиксированное число шагов. Запуск производился со следующими настройками: стратегия отбора – элитизм, для размножения отбирались 25% популяции, имеющих наибольшее значение фитнес-функции; частота мутации – 2%; размер популяции – 200 особей; число популяций – 100; фитнес-функция – среднее значение съеденной еды на 200 случайных картах полей; карты внутри одной популяции совпадают; карты различных популяций различны. Последнее измерение фитнес-функции осуществлялось на случайном наборе из 2000 карт.

Результаты эксперимента приведены в таблице.

μ	0,04			
	Функция приспособленности			
Число состояний	2	4	8	16
Полные таблицы	17,18	15,94	15,03	13,68
Деревья решений	18,28	20,28	18,60	20,18
Предложенный метод	18,82	16,44	19,75	15,46

Таблица. Результаты экспериментов

Анализ результатов показывает, что предложенный метод является более эффективным по сравнению с представлением функции переходов полными таблицами. При некоторых значениях числа состояний предложенный метод более эффективен по сравнению с методом представления функции переходов деревьями решений.

Исследование выполнено по Федеральной целевой программе «Научные и научно-педагогические кадры инновационной России на 2009–2013 годы» в рамках государственного контракта П1188 от 27 августа 2009 года.

Литература

1. Koza J.R. Genetic programming: On the Programming of Computers by Means of Natural Selection. – Cambridge: MIT Press, 1992.
2. Гладков Л.А., Курейчик В.В., Курейчик В.М. Генетические алгоритмы. – М.: Физматлит, 2006.
3. Поликарпова Н.И., Точилин В.Н., Шальто А.А. Применение генетического программирования для генерации автоматов с большим числом входных переменных // Научно-технический вестник СПбГУ ИТМО. – 2008. – № 53. – С. 24–42.
4. Данилов В.Р. Метод представления автоматов деревьями решений для использования в генетическом программировании // Научно-технический вестник СПбГУ ИТМО. – 2008. – № 53. – С. 103–108.
5. Шальто А.А. Логическое управление. Методы аппаратной и программной реализации. – СПб: Наука, 2000. – 780 с.
6. Bryant R.E. Graph-based algorithms for boolean function manipulation // IEEE Transactions on Computers. – 1986. – № 8. – P. 677–691.
7. Бедный Ю.Д., Шальто А.А. Применение генетических алгоритмов для построения автоматов в задаче «Умный муравей». – СПбГУ ИТМО, 2007 [Электронный ресурс]. – Режим доступа: <http://is.ifmo.ru/works/ant>, свободный. Яз. рус. (дата обращения 09.02.2011).

Данилов Владимир Рюрикович

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, Vladimir.Daniloff@gmail.com

Шальто Анатолий Абрамович

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, доктор технических наук, профессор, зав. кафедрой, shalyto@mail.ifmo.ru

УДК 004.4'242

АВТОМАТИЧЕСКИЙ ПОДБОР ПАРАМЕТРОВ ВНЕШНЕЙ СРЕДЫ ПРИ ГЕНЕРАЦИИ АВТОМАТНЫХ ПРОГРАММ С ПОМОЩЬЮ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ

В.А. Кулев

Предлагается новый метод генетического программирования, позволяющий улучшить способ построения функции приспособленности для генетических алгоритмов, проводящих тестирование особей путем симуляции окружающей среды. Метод позволяет автоматически подбирать параметры внешней среды, что приводит к улучшению качества генерируемых особей. В ходе исследований на примере задачи об «Итерированной дилемме узника» было показано, что разработанный подход применим для генерации автоматных программ в задачах, где поведение окружающей среды может быть описано с помощью конечного автомата.

Ключевые слова: генетическое программирование, конечный автомат, автоматное программирование.

Введение

Вычисление функции приспособленности на основе симуляции внешней среды получило наибольшее распространение благодаря своей очевидности и простоте. В некоторых ситуациях этот подход также представляется единственным возможным, при этом множество параметров среды, которое используется для вычисления функции приспособленности, намного меньше множества возможных параметров среды. Это множество, как правило, задается априори, исходя из эвристических соображений.