

МЕТОД ПОСТРОЕНИЯ КОНЕЧНЫХ АВТОМАТОВ НА ОСНОВЕ МУРАВЬИНОГО АЛГОРИТМА

Чивилихин Д.С.

*магистрант кафедры компьютерных технологий НИУ ИТМО,
chivilikhin.daniil@gmail.com*

Ульянцев В.И.

*магистрант кафедры компьютерных технологий НИУ ИТМО,
ulyantsev@rain.ifmo.ru*

Аннотация: Предлагается новый метод построения конечных автоматов, основанный на сведении этой задачи к поиску на графе и применении муравьиного алгоритма нового типа для поиска решений в этом графе. Результаты экспериментального исследования предложенного метода позволяют говорить о его высокой эффективности по сравнению с генетическим алгоритмом.

Введение

В парадигме автоматного программирования [1] выделяется система управления, представляемая в виде конечного автомата, и объект управления. В некоторых случаях автомат управления может быть построен вручную, однако для большинства объектов со сложным поведением этот процесс является весьма трудоемким. Для автоматизированного построения конечных автоматов в основном применяются эволюционные алгоритмы [2–4].

В данной работе предлагается новый алгоритм построения конечных автоматов, основанный на муравьином алгоритме [5]. Приводятся результаты экспериментального исследования эффективности разработанного метода на задаче об «Умном муравье» [6].

Постановка задачи

Конечный автомат — это шестерка $\langle S, s_0, \Sigma, \Delta, \delta, \lambda \rangle$, где S — множество состояний, $s_0 \in S$ — начальное состояние, Σ — множество входных событий, Δ — множество выходных воздействий. Функция $\delta: S \times \Sigma \rightarrow S$ называется функцией переходов, а функция $\lambda: S \times \Sigma \rightarrow \Delta$ — функцией выходов. За стартовое состояние s_0 в настоящей работе всегда принимается первое состояние. Пример диаграммы состояний конечного автомата приведен на рис. 1.

Пусть задано число состояний автомата N , множество входных событий Σ , множество выходных воздействий Δ и вещественная *функция приспособленности* (ФП) f , определенная на множестве всех автоматов с параметрами (N, Σ, Δ) . Значения ФП автомата пропорциональны близости его поведе-

ния к желаемому. Требуется найти автомат x такой, что значение ФП этого автомата $f(x)$ не меньше заданного барьерного значения.

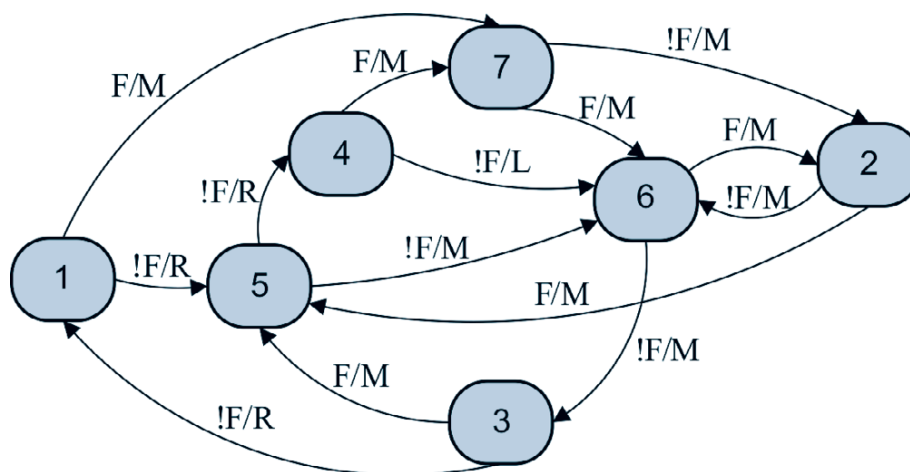


Рис. 1. Пример диаграммы состояний конечного автомата из семи состояний с $\Sigma = \{F, !F\}$, $\Delta = \{L, M, R\}$

Представление пространства поиска в виде графа

Задача поиска оптимального автомата сводится к поиску оптимальной вершины в ориентированном графе G , вершинами которого являются конечные автоматы, а ребрами — *мутации* конечных автоматов. Мутация конечного автомата – небольшое изменение в его структуре. В предлагаемом алгоритме используется два типа мутаций автоматов: для случайного перехода меняется либо состояние, в которое он ведет, либо записанное на нем выходное воздействие.

Опишем граф G , в котором производится поиск оптимальной вершины. Две вершины в полностью построенном графе G соединены ребром, если соответствующие вершинам автоматы могут быть получены друг из друга с помощью одной операции мутации. Таким образом, применяя определенные мутации, можно получить из любого автомата любой другой автомат. Кроме того, на каждом ребре (u, v) графа задается значение эвристической информации, вычисляемое по формуле:

$$\eta_{uv} = \max(\eta_{\min}, f(v) - f(u)),$$

где η_{\min} — небольшая положительная константа, например, 0,001.

Муравьиные алгоритмы

В муравьиных алгоритмах [5] решения строятся набором агентов-муравьев, называемых *муравьиной колонией*. Каждый муравей перехо-

дит по ребрам графа от вершины к вершине, пользуясь неким вероятностным алгоритмом. При этом вершины графа обычно соответствуют компонентам решения задачи, а полное решение представляется последовательностью или множеством вершин. На каждом ребре графа задаются две величины, *эвристическая информация* и *значение феромона*. Значения эвристической информации фиксируются перед началом работы алгоритма и не изменяются во время его работы. Значения феромона модифицируются муравьями в процессе работы алгоритма и реализуют долговременную память колонии. Можно выделить следующие основные этапы работы любого муравьиного алгоритма, которые повторяются, пока не будет найдено подходящее решение или не выполнится одно из условий останова.

1. **Построение решений муравьями.** Муравьи переходят от одной вершины графа к другой, руководствуясь при выборе пути эвристической информацией и значениями феромона на ребрах графа.
2. **Обновление значений феромона.** Значение феромона на каждом ребре графа может увеличиться за счет откладывания феромона муравьями или уменьшиться за счет испарения — равномерного уменьшения значений феромона.

Предлагаемый алгоритм построения конечных автоматов

Общая схема предлагаемого муравьиного алгоритма в целом совпадает с изложенной выше схемой работы классических муравьиных алгоритмов. Ключевой особенностью нового муравьиного алгоритма является то, что вершинами графа в новом алгоритме являются не компоненты решений, а полные решения — конечные автоматы.

В начале работы алгоритма строится случайный конечный автомат с заданным числом состояний. Далее, к случайному автомату применяется $(1+1)$ -эволюционная стратегия в течение небольшого числа вычислений ФП. Эволюционная стратегия хранит одно решение и выполняет одну случайную мутацию этого решения на каждом шаге, заменяя старое решение новым в случае увеличения или сохранения значения ФП. Автомат, построенный с помощью эволюционной стратегии, добавляется в граф и становится его первой вершиной.

Построение решений муравьями

Все муравьи начинают поиск из вершины, соответствующей лучшему из найденных решений. На каждом шаге муравей, находясь в вершине u , соответствующей автомату A , выполняет одно из следующих действий.

1. **Построение новых решений.** Муравей выполняет N_{mut} случайных мутаций автомата A . Результатом выполнения каждой мутации является некий автомат A_{mut} . Далее, если в графе G нет вершины t , ассоциированной с автоматом A_{mut} , такая вершина создается и добавляется в граф. Наконец, в граф добавляется ребро (u, t) . После выполнения всех мутаций муравей переходит в вершину, соответствующую лучшему из построенных на этом шаге автоматов.
2. **Выбор из существующих решений.** Следующая вершина выбирается из множества вершин N_u , инцидентных текущей вершине u . Вершина $v \in N_u$ выбирается с вероятностью:

$$p_{uv} = \frac{\tau_{uv}^\alpha \cdot \tau_{uv}^\beta}{\sum_{w \in N_u} \tau_{uw}^\alpha \cdot \tau_{uw}^\beta},$$

где $\alpha, \beta \in [0, 1]$ — параметры, определяющие значимость феромона и эвристической информации при выборе пути.

Выбор следующей вершины на каждом шаге муравья осуществляется с помощью построения новых решений с вероятностью p_{new} и путем выбора из существующих решений с вероятностью $1 - p_{\text{new}}$. Муравьи запускаются параллельно — каждый муравей делает один шаг и передает ход следующему. На одной итерации муравьиной колонии каждый муравей строит решения до тех пор, пока не исчерпает максимальное число ходов n_{stag} , на которых не произошло увеличения лучшего найденного им значения ФП. Колония может совершить не более N_{stag} итераций без увеличения лучшего значения ФП. После этого алгоритм перезапускается.

Обновление феромона

На каждом ребре графа, кроме значения феромона τ_{uv} , хранится число τ_{uv}^{\max} — наибольшее из всех значений феромона, когда-либо отложенных на ребре (u, v) . Из пути каждого муравья выделяется участок, ведущий от начала к лучшей из вершин пути, и на ребрах этого участка обновляются значения τ_{uv}^{\max} :

$$\tau_{uv}^{\max} = \max(\tau_{uv}^{\max}, f(\bar{x})),$$

где \bar{x} — вершина пути муравья с наибольшим на пути значением ФП. Затем значения феромона на каждом ребре графа обновляются по формуле:

$$\tau_{uv} = \max(\tau_{\min}, \rho \tau_{uv} + \tau_{uv}^{\max}),$$

где $\rho \in [0, 1]$ — скорость испарения феромона, а τ_{\min} — минимальное значение феромона.

Эксперименты: задача об «Умном муравье»

В задаче об «Умном муравье» [6] задано тороидальное поле размером 32×32 клетки. На поле вдоль некоторой ломаной распределены «яблоки». В данной работе рассматриваются две конфигурации задачи – поле Джона Мура и поле Санта Фе (рис. 2). Оба поля содержат по 89 клеток с яблоками. Черные клетки содержат яблоки, серые клетки обозначают пустые клетки оптимального пути, а белые клетки пусты. Муравей изначально располагается в левой верхней клетке и «смотрит» на восток. Находясь в некоторой клетке, он может определить, есть ли в следующей клетке яблоко. На каждом шаге муравей может повернуть налево, повернуть направо или перейти вперед на одну клетку. Если в клетке, в которую переходит муравей, находится яблоко, муравей его «съедает».

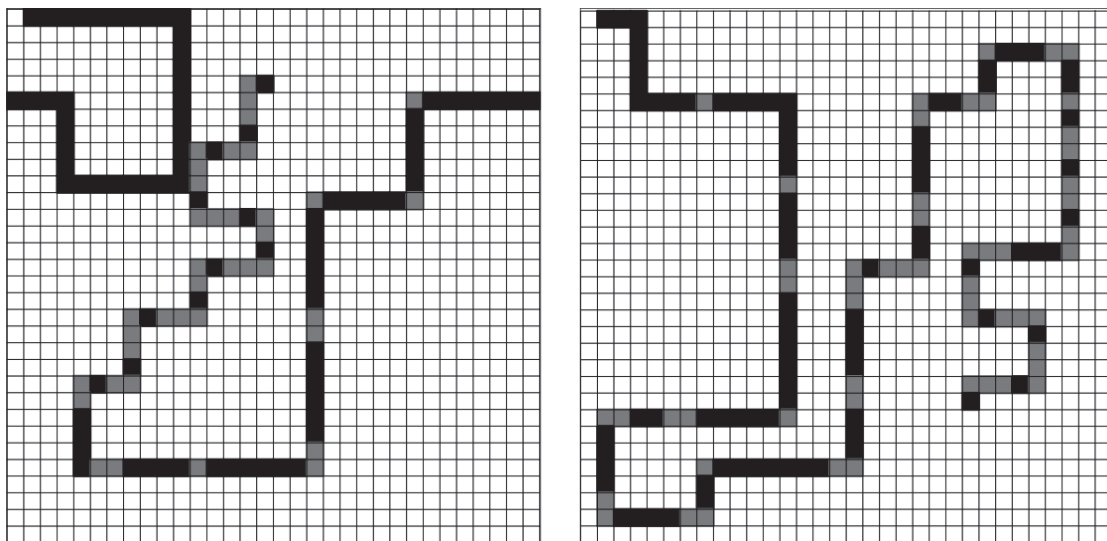


Рис. 2. Поле Джона Мура (слева) и поле Санта Фе (справа)

В описанной задаче целью является построение конечного автомата для управления муравьем, который позволит ему съесть все яблоки за отведенное число шагов s_{\max} . Используемая ФП имеет вид:

$$f = n_{\text{food}} + \frac{s_{\max} - s_{\text{last}} - 1}{s_{\max}},$$

где n_{food} — число съеденных яблок, а s_{last} — номер шага, на котором было съедено последнее яблоко. Переходы автоматов в этой задаче могут быть

помечены двумя входными событиями: F (следующая клетка содержит яблоко) и $!F$ (следующая клетка пуста), а также выходными воздействиями L (повернуть налево), R (повернуть направо) и M (сделать шаг вперед).

На рис. 3 приведены графики, позволяющие сравнить предложенный алгоритм с генетическим алгоритмом [4] для поля Джона Мура при $s_{\max} = 200$. Из рис. 3 видно, что для рассмотренной задачи предложенный алгоритм эффективнее генетического алгоритма. Более того, для построения автоматов из семи состояний, решающих поставленную задачу, предложенному алгоритму требуется в среднем 31×10^6 вычислений ФП, в то время как генетическому алгоритму требуется приблизительно в 60 раз больше — 1800×10^6 вычислений ФП. Диаграмма состояний одного из построенных автоматов изображена на рис. 1. Этот автомат позволяет муравью съесть всю еду за 189 шагов.

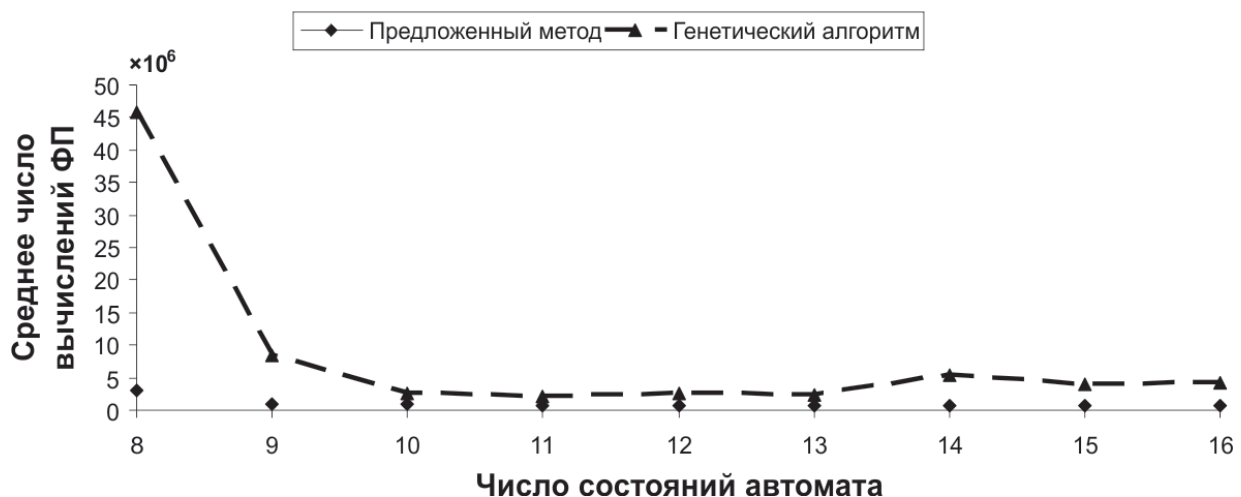


Рис. 3. Поле Джона Мура: зависимость среднего числа вычислений ФП от числа состояний автомата при $s_{\max} = 200$

Пометки об использовании переходов автомата

При разработке алгоритма и проведении экспериментальных исследований был предложен подход, не имеющий прямого отношения к муравьиному алгоритму, однако позволяющий увеличить его производительность. Предлагается при вычислении значения ФП автомата пометать переходы, которые при этом совершались. Если переход автомата не использовался при вычислении ФП, то поведение автомата при изменении такого перехода не может измениться, следовательно, значение ФП можно не вычислять заново.

На рис. 5 приведены графики зависимости среднего числа вычислений ФП от числа состояний автомата с использованием и без использования по-

меток на переходах для поля Санта Фе при $s_{\max} = 600$. Как видно из графиков, выигрыш от использования пометок увеличивается с ростом числа состояний целевого автомата. Это объясняется тем, что вероятность совершить мутацию, не меняющую поведение автомата, также увеличивается с ростом числа его состояний.

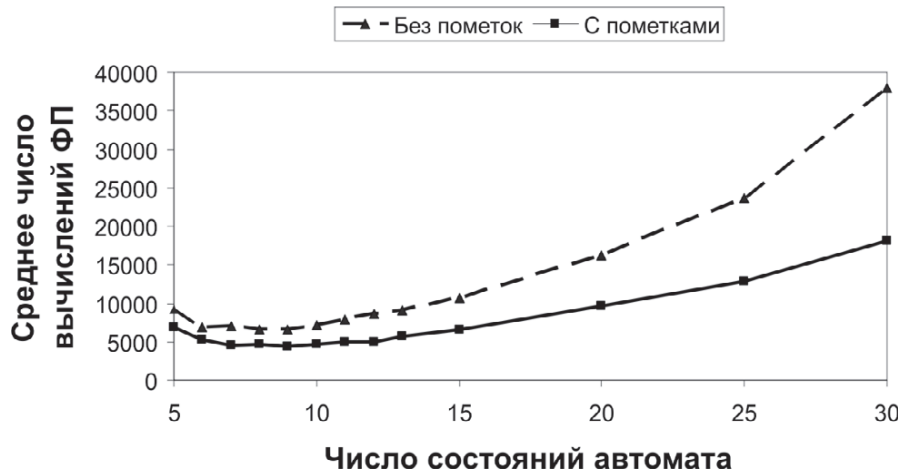


Рис. 5. Влияние пометок об использовании переходов на производительность предложенного алгоритма

Заключение

Был разработан метод построения конечных автоматов, основанный на муравьином алгоритме. Новый метод сравнивался с генетическим алгоритмом на примере задачи об «Умном муравье». В результате было установлено, что предложенному методу требуется меньше вычислений функции приспособленности для получения идеальных решений, что позволяет говорить о его высокой эффективности. Также был предложен подход к уменьшению числа вычислений ФП путем введения пометок об использовании переходов автомата при вычислении ФП. Этот подход может также быть применен для увеличения эффективности эволюционных алгоритмов построения конечных автоматов.

Литература

1. Полицарпова Н.И., Шалыто А.А. Автоматное программирование. — 2011. — СПб: Питер. — 176 с.
2. Lucas S., Reynolds J. Learning Finite State Transducers: Evolution versus Heuristic State Merging // IEEE Transactions on Evolutionary Computation. — 2007. — Vol. 11. — №3. — P. 308–325.
3. Chellapilla K., Czarnecki D. A preliminary investigation into evolving modular finite state machines // Proceedings of the 1999 Congress on Evolutionary Computation. — 1999. — Vol. 2. — P. 1349–1356.

4. *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». — 2007. — Т. 2. — С. 88–91.
 5. *Dorigo M.* Optimization, Learning and Natural Algorithms. PhD thesis. Polytechnico di Milano, Italy. — 1992.
 6. *Jefferson D., Collins R., Cooper C., Dyer M., Flowers M., Korf R., Taylor C., Wang A.* Evolution as a theme in artificial life: The Genesys/Tracker system. Technical report. — 1990.
-