

ПРИМЕНЕНИЕ МУРАВЬИНЫХ АЛГОРИТМОВ ДЛЯ ПОСТРОЕНИЯ КОНЕЧНЫХ АВТОМАТОВ

Д. С. Чивилихин, В. И. Ульянов

Санкт-Петербургский государственный университет
информационных технологий, механики и оптики

Постановка задачи

В рамках парадигмы автоматного программирования ключевыми компонентами программ и программных комплексов являются конечные автоматы. В некоторых случаях автоматы могут быть построены вручную, однако для большинства реальных задач эвристическое построение автоматов является весьма трудоемким. Для автоматической генерации конечных автоматов обычно применяются различные эволюционные алгоритмы.

Управляющий конечный автомат задается множеством состояний, начальным состоянием, множеством входных событий и множеством выходных воздействий, а также функцией перехода, сопоставляющей каждому состоянию автомата и каждому входному событию другое состояние и выходное воздействие.

Для каждой задачи, функция приспособленности — вещественнозначная функция, заданная на множестве всех конечных автоматов с заданными параметрами. В общем случае, задача генерации конечного автомата по заданной функции приспособленности может быть сформулирована следующим образом: по заданному числу состояний, множеству входных событий и множеству выходных воздействий найти конечный автомат с целевым значением заданной функции приспособленности.

Муравьиные алгоритмы — семейство методов решения задач оптимизации, которые могут быть сведены к поиску по графу, таких как, например, задача о коммивояжере. В рамках муравьиных алгоритмов решения строятся набором агентов-муравьев, использующих при выборе пути в графе некоторую стохастическую стратегию. Решения могут быть представлены как путями в графе, так и отдельными его вершинами. В данной работе исследуется возможность применения муравьиных алгоритмов для построения конечных автоматов по заданной функции приспособленности путем сведения этой задачи к поиску целевой вершины в графе.

Цель работы

Целью настоящей работы является разработка метода генерации конечных автоматов для заданной функции приспособленности на основе муравьиных алгоритмов и оценка его эффективности.

Описание предлагаемого подхода

В качестве представления пространства поиска, то есть множества всех конечных автоматов с заданными параметрами, рассматривается граф, вершины которого соответствуют решениям (конечным автоматам) а ребра — *мутациям* конечных автоматов. Под мутацией конечного автомата понимается небольшое изменение его структуры, например, изменение выходного воздействия на переходе или изменение состояния, в которое ведет переход.

Каждому ребру графа ставится в соответствие некоторое вещественное число — значение *феромона*. Значение феромона на ребре может быть увеличено, если по этому ребру прошел муравей. Также, на каждой итерации алгоритма происходит испарение — уменьшения значений феромона на всех ребрах графа в одинаковое число раз. Алгоритм генерации автомата может быть разделен на несколько этапов:

1. Построение решений муравьями-агентами.

Муравей, находясь в определенной вершине графа, выбирает следующую вершину руководствуясь одним из двух правил:

- a) построить новые ребра графа, производя мутации автомата, соответствующего текущей вершине и выбрать лучшее из новых решений;
- b) выбрать следующую вершину исходя из значений феромона на ребрах, ведущих из текущей вершины по классической формуле выбора пути в муравьиных алгоритмах.

2. Обновление значений феромона на ребрах.

3. Проверка условий останова.

Шаги 1–3 повторяются до тех пор, пока не будет найдено решение, удовлетворяющее условиям задачи, или не выполнится другое условие останова, например, истечет заданное максимальное число итераций алгоритма

Результаты

Был разработан и реализован метод генерации конечных автоматов по заданной функции приспособленности, основанный на муравьином алгоритме. Эффективность разработанного метода была оценена путем сравнения с генетическими алгоритмами для задачи об «Умном муравье» и задаче о генерации автоматов на основе тестовых примеров. Эксперименты показали, что среднее время работы предлагаемого алгоритма для построения целевых автоматов в этих задачах в несколько раз меньше времени работы генетического алгоритма.
