

МЕТОД ПОСТРОЕНИЯ КОНЕЧНЫХ АВТОМАТОВ НА ОСНОВЕ МУРАВЬИНОГО АЛГОРИТМА

Д.С. Чивилихин, В.И. Ульянов

Научный руководитель – ассистент Ф.Н. Царев

В парадигме автоматного программирования выделяется система управления, представляемая в виде конечного автомата, и объект управления. В некоторых случаях автомат управления может быть построен эвристически, однако для большинства систем со сложным поведением построение автоматов вручную является весьма трудоемким. Для автоматического построения конечных автоматов применяются такие алгоритмы, как метод имитации отжига, эволюционные стратегии, генетические алгоритмы и обучение с подкреплением.

Конечный автомат задается множеством состояний S , стартовым состоянием s_0 , множеством входных событий Σ и множеством выходных воздействий Δ , а также функцией переходов $\delta: S \times \Sigma \rightarrow S$ и функцией действий $\lambda: S \times \Sigma \rightarrow \Delta$.

При решении задач управления вводится так называемая функция приспособленности автомата, значения которой тем больше, чем ближе поведение автомата управления к желаемому поведению. Следовательно, для получения автомата с желаемым поведением необходимо найти автомат, значение функции, приспособленности которого не меньше некоего наперед заданного числа.

В последние два десятилетия для решения самых разнообразных комбинаторных задач было разработано множество новых алгоритмов из класса алгоритмов роевого интеллекта, работа которых основана на моделировании поведения коллективов живых существ. Одним из подклассов алгоритмов роевого интеллекта являются муравьиные алгоритмы оптимизации, основанные на поведении колоний муравьев в процессе поиска пищи. Авторами был разработан новый алгоритм построения конечных автоматов, основанный на муравьином алгоритме.

Целью работы является экспериментальное исследование разработанного метода построения конечных автоматов, основанного на муравьином алгоритме, на хорошо известных задачах и сравнение характеристик его производительности с другими алгоритмами, такими, как генетические алгоритмы и эволюционные стратегии.

Постановка задачи. Пусть задано число состояний автомата N , множество входных событий Σ , множество выходных воздействий Δ и вещественная функция приспособленности f , заданная на множестве всех автоматов с параметрами (N, Σ, Δ) . Требуется найти автомат x такой, что значение функции приспособленности этого автомата $f(x)$ не меньше заданного барьерного значения b .

Основные положения исследования. Для решения поставленной задачи сведем ее к задаче поиска оптимальной вершины в некотором графе, вершинами в котором 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 случайно выбранного перехода в автомате меняется либо состояние, в которое ведет этот переход, либо действие, которое на нем выполняется.

В муравьиных алгоритмах решения строятся набором агентов-муравьев, называемых муравьиной колонией. Каждый муравей переходит по ребрам графа от вершины к вершине, пользуясь неким стохастическим алгоритмом для выбора следующего ребра.

На каждом ребре графа задаются две величины, эвристическая информация и значение феромона. Эвристическая информация на ребре определяется как разность значений

функции приспособленности вершин, которые связывает ребро, в то время как значения феромона изменяются муравьями в процессе поиска.

Можно выделить следующие основные этапы работы алгоритма, которые повторяются, пока не будет найдено подходящее решение или не выполнится одно из условий останова:

1. Размещение муравьев по вершинам графа.

Каждый муравей помещается в одну из вершин графа, из которой он начнет строить решения. Одной из лучших стратегий является размещение всех муравьев в вершину, соответствующую текущему лучшему решению.

2. Построение решений муравьями.

Каждый муравей, независимо от других, строит решения, пока не израсходует выделенные ему ресурсы. На каждом шаге муравей с некоторой вероятностью выбирает следующее ребро из числа ребер, инцидентных текущей вершине, руководствуясь эвристической информацией и значениями феромона на этих ребрах. Иначе муравей выполняет некоторое число мутаций текущего решения и переходит во вновь построенную вершину, имеющую наибольшее значение функции приспособленности.

3. Обновление значений феромона.

Сначала значения феромона на ребрах графа, посещенных каждым муравьем, увеличиваются пропорционально значению функции приспособленности лучшего из решений, найденных этим муравьем. Затем происходит испарение феромона – значения феромона на всех ребрах графа уменьшаются в одинаковое число раз.

Результаты. Экспериментальное исследование разработанного алгоритма проводилось для широко известной задачи об «Умном муравье». Рассматривалось несколько конфигураций задачи при различных ограничениях на число ходов муравья и для различных размеров целевого автомата. Разработанный алгоритм сравнивался с генетическим алгоритмом, генетическим программированием и эволюционными стратегиями. В результате было установлено, что во всех случаях муравьиному алгоритму требуется меньше вычислений функции приспособленности для получения идеальных решений, что позволяет говорить о высокой эффективности разработанного метода.

УДК 004.4'242

МИНИМАЛЬНАЯ МОДИФИКАЦИЯ АВТОМАТНЫХ ПРОГРАММ ПРИ ИЗМЕНЕНИИ СЦЕНАРИЕВ ИХ РАБОТЫ

А.В. Шестаков

Научный руководитель – ассистент Ф.Н. Царев

Введение. Одним из подходов к программированию объектов со сложным поведением является использование парадигмы автоматного программирования [1] или «программирование с явным выделением состояний» [2]. При использовании этого подхода поведение программы задается в виде управляющих конечных автоматов. Состояния каждого автомата соответствуют состояниям управляемого объекта, а переходы между состояниями осуществляются при выполнении заданных условий.

Автоматное программирование хорошо подходит для создания событийно-ориентированных приложений [3, 4]. По каждому событию автомат в зависимости от состояния генерирует определенное выходное воздействие, а последовательность событий порождает последовательность выходных воздействий. Такая цепочка пар «событие – воздействие» называется сценарием работы автоматной программы.

Изменение сценариев работы требует модификации автоматной программы. Метод, предложенный в данной работе, позволяет изменить программу в соответствии с новыми