

Санкт-Петербургский национальный исследовательский
университет информационных технологий, механики и оптики
Факультет информационных технологий и программирования
Кафедра «Компьютерные технологии»

А. С. Касаткин

Отчет по лабораторной работе
«Применение генетических алгоритмов для построения
автоматов в задаче об умном муравье»

Вариант № 25

Санкт-Петербург
2011

Введение

В данной лабораторной работе изучается влияние процента элитных особей в поколении на скорость работы генетического алгоритма построения конечных автоматов Мили. В качестве примера взята задача об умном муравье. Результатом лабораторной работы являются автоматы Мили, построенные при помощи генетического алгоритма.

При выполнении работы были написаны восемь классов на языке программирования Java. Использовались только стандартные библиотеки и библиотека JFreeChart для построения графиков.

1. Постановка задачи

Задача лабораторной работы — оценить зависимость эффективности генетического алгоритма при решении задачи об умном муравье при различных процентах элитных особей в поколении.

Для решения задачи используется генетический алгоритм. Способ представления особи — конечный автомат Мили с его представлением в виде графа переходов и начального состояния.

1.1. Задача об умном муравье

В задаче об умном муравье рассматривается поле, состоящее из клеток (рис. 1).

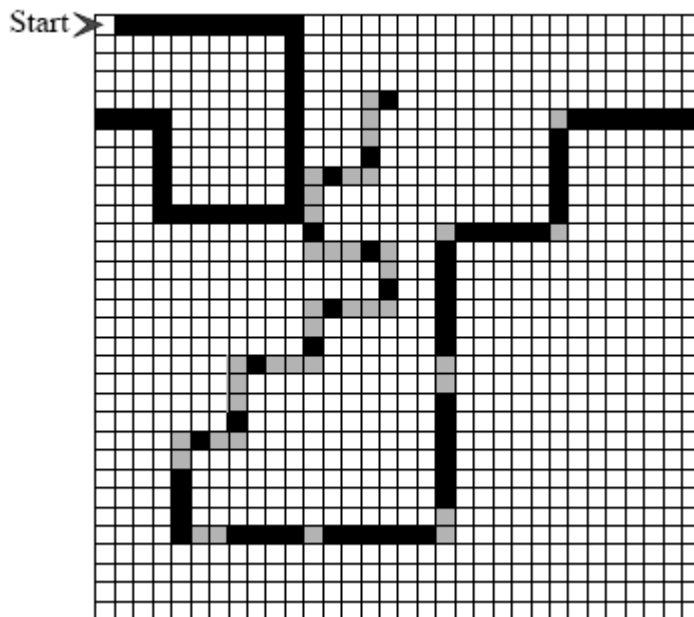


Рис. 1 Поле

Поле имеет размеры 32x32 клетки и располагается на поверхности тора. Некоторые клетки поля пусты, некоторые — содержат по одному яблоку. Всего на поле 89 яблок.

Муравей начинает свое движение из клетки, помеченной как «Start». За один ход муравей может определить, есть ли в клетке перед ним яблоко, и выполнить одно из следующих действий:

- повернуть налево;
- повернуть направо;
- сделать шаг вперед, и если в новой клетке есть яблоко, то съесть его;
- ничего не делать.

Максимальное число шагов — 200. Цель — создать автомат Мили с фиксированным числом состояний, при помощи которого можно съесть все яблоки.

1.2. Автомат Мили

Автомат, у которого выходное воздействие формируется на основании входного воздействия и состояния автомата, называется автоматом Мили.

В данной задаче возможны всего два варианта входного воздействия:

- *True* — перед муравьем есть еда;
- *False* — перед муравьем нет еды.

Так же бывает четыре выходных воздействия:

- *MoveForward* — сделать шаг вперед;
- *TurnLeft* — повернуть налево;
- *TurnRight* — повернуть направо;
- *Nothing* — ничего не делать.

2. Генетический алгоритм

Для задачи поиска оптимального автомата, управляющего муравьем, применяется генетический алгоритм. Его работа описывается несколькими простыми фазами.

В начале происходит генерация стартового поколения особей. Каждая особь в этом поколении генерируется случайным образом. Затем алгоритм начинает выполнение итеративного процесса — на каждой итерации алгоритм считает функцию приспособленности для каждой особи в поколении и сортирует по ее значению поколение. После этого некий процент особей с максимальной функцией приспособленности переходит в следующее поколение, а к оставшимся применяется оператор кроссовера, после него каждая получившаяся особь мутирует с некоей вероятностью. Так же, если на протяжении большого числа поколений максимальное значение функции приспособленности не изменяется, то происходит большая мутация: все особи кроме элитных заменяются на случайно сгенерированные. Большие мутации необходимы из-за достаточно быстрой сходимости алгоритма.

2.1. Представление особи

Особями в данном алгоритме являются конечные автоматы Мили. Автомат Мили представляется в виде графа переходов, который реализуется в виде массива вершин, в которых записаны действия и переходы по каждому из входных воздействий.

2.2. Мутация

Оператор мутации применяется к одной особи. Он совершает одно из четырех равновероятных действий:

- меняет стартовое состояние в автомате;
- меняет в случайном состоянии и случайном входном воздействии выходное воздействие на случайное;
- меняет в случайном состоянии и случайном входном воздействии состояние в которое переходит автомат на случайное;
- меняет в случайном состоянии выходные воздействия и состояния в которые переходит автомат между собой местами.

2.3. Функция приспособленности

Функция приспособленности вычисляется по формуле $Fitness = Apples - LastStep/200$, где *Apples* — число яблок, съедаемых муравьем за 200 шагов, *LastStep* — номер шага, на котором муравей съедает последнее яблоко.

Таким образом, автомат, позволяющий съесть большее число яблок, будет иметь большее значение функции приспособленности. Среди автоматов, съедающих одинаковое число яблок, большее значение функции приспособленности будет иметь тот, который раньше съедает последнее яблоко.

2.4. Оператор кроссовера

Оператор кроссовера, используемый в данной реализации генетического алгоритма, применяется только к одной не элитной особи. Парой для скрещивания будет случайно выбранная элитная особь, причем элитная особь не изменяется, а только служит донором.

Сам оператор кроссовера выглядит просто. В начале выбирается случайная элитная особь. Она будет выступать донором. Далее для всех состояний особи, к которой применяется кроссовер, выполняется одно из четырех равновероятных действий:

- скопировать из элитной особи выходное воздействие и состояние в которое переходит автомат при входном воздействии *True*;
- скопировать из элитной особи выходное воздействие и состояние в которое переходит автомат при входном воздействии *False*;
- скопировать из элитной особи выходное воздействие и состояние в которое переходит автомат по всем входным воздействиям;
- ничего не делать.

3. Построенные графики и их анализ

В результате работы описанного выше алгоритма были получены графики зависимости максимума функции приспособленности в поколении в зависимости от номера поколения.

На рис. 2 изображен график зависимости функции приспособленности от номера поколений, построенный для 30000 поколений, состоящих из 300 особей с 7 состояниями, и усредненный по результатам 400 запусков. Максимальное значение функции приспособленности — 87,035.

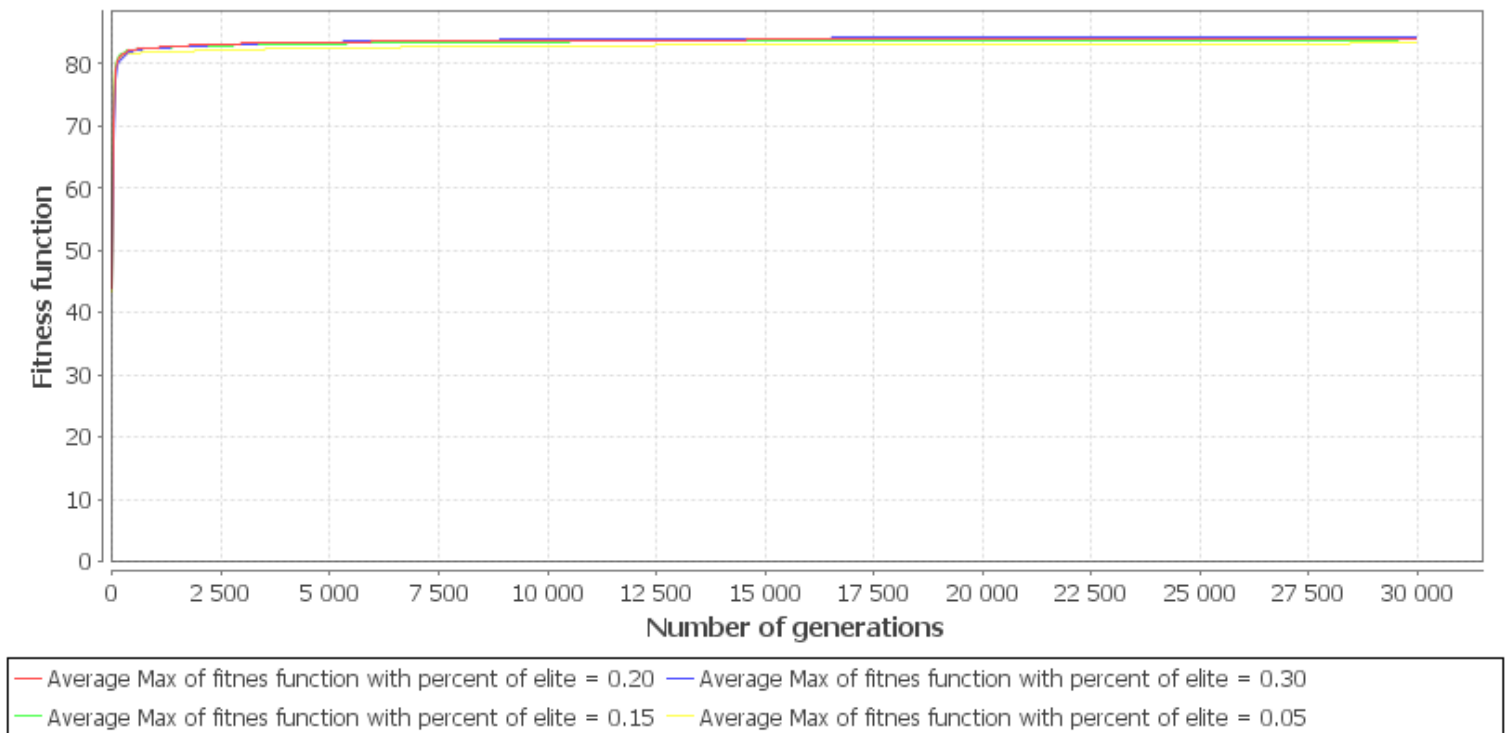


Рис. 2 Усредненные графики значения функции приспособленности от номера поколения

Из графика видно, что значение функции приспособленности довольно быстро достигает 83, а дальше растет очень медленно. Дальнейший рост связан с тем, что за счет больших мутаций в поколении появляются новые, отличные от элитных, особи.

Так же этот график показывает, что процент элитных особей хоть и влияет на рост значения функции приспособленности, но не играет критической роли, а при большом номере поколения вообще не важен. Это рассуждение верно только для малых процентов элитных особей, т. к. при больших процентах элитных особей в поколении слишком мало разнообразия. Это видно из графиков для большого набора вариантов процентов элитных особей, изображенных на рис. 3.

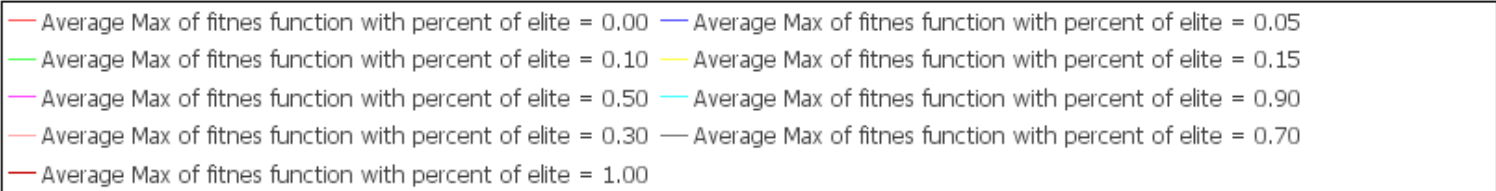
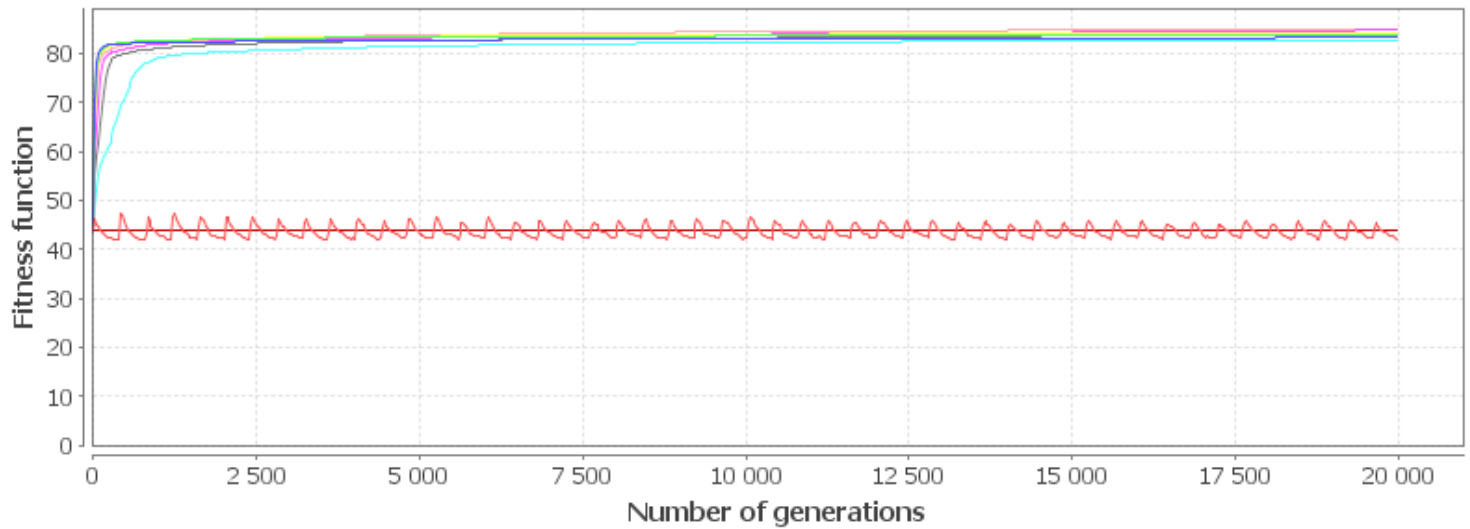


Рис. 3 Усредненные графики значения функции приспособленности от номера поколения

Заключение

Результаты лабораторной работы показали, что при создании генетического алгоритма можно использовать практически любой процент элитных особей от 5 до 30 процентов и эффективность алгоритма в большей степени зависит не от процента элитных особей, а от стратегии отбора и оператора кроссовера.

При используемых в работе операторах кроссовера и мутации наилучший результат был достигнут при 15 процентах элитных особей.

Исходные коды алгоритма выложены в свободный доступ по адресу <https://bitbucket.org/Kasetkin/cleverantproject>

Источники

1. [Н. Поликарпова, А. Шалыто «Автоматное программирование»;](#)
2. [Ф. Царев, А. Шалыто «Применение генетического программирования для генерации автомата в задаче об «Умном муравье».](#)