

Application of Genetic Algorithms for Construction of Moore Automaton and Systems of Interacting Mealy Automata in “Artificial Ant” Problem

Andrey A. Davydov, Dmitry O. Sokolov, Fedor N. Tsarev.

Scientific advisor: prof. Anatoly A. Shalyto

1

Abstract—a genetic algorithm for construction of Moore finite state machines is described in the paper. This algorithm can be also applied to construct systems of interacting Mealy finite state machines. An example of application of these algorithms for “Artificial ant” problem is also described.

Index terms—Artificial ant, automata based programming, genetic algorithm

I. INTRODUCTION

AUTOMATA based programming technology [1] is widely used at the last time. In the context of this technology behavior of programs is described using deterministic finite state machines.

It is often possible to construct automaton using heuristic methods but in some problems such a method consumes a lot of time or such construction is even impossible. Sample of such a problem is “Artificial ant” problem [2]. Genetic algorithms and genetic programming [2–9] can be applied to construct automata in problems of this kind.

All known to author’s works in this context [10] are devoted to constructing only one Mealy automaton. But Moore automata and systems of interacting automata [1] are also allowed in automata based programming. It is significant

¹Manuscript received March 31, 2008.

Andrey A. Davydov is a third-year student of Computer Technologies Department of St.-Petersburg State University of Information Technology, Mechanic and Optics. e-mail: Andrey.a.Davydov@gmail.ru

Dmitry O. Sokolov is a third-year student of Computer Technologies Department of St.-Petersburg State University of Information Technology, Mechanic and Optics. e-mail: dimoz_88@rambler.ru

Fedor N. Tsarev is a fifth-year student of Computer Technologies Department of St.-Petersburg State University of Information Technology, Mechanic and Optics. e-mail: tsarev@rain.ifmo.ru

Anatoly A. Shalyto is a professor of Computer Technologies Department of St.-Petersburg State University of Information Technology, Mechanic and Optics. e-mail: shalyto@mail.ifmo.ru

Location of St.-Petersburg State University of Information Technology, Mechanic and Optics: Saint-Petersburg, Kronversky av. 49

than one of the most important way of automata interaction is nesting.

Main goal of this paper is to propose an algorithm for constructing using genetic programming [7] a pair of nested Mealy automata and Moore automaton for solving “Artificial ant” problem.

II. PROBLEM DESCRIPTION

Description of “Artificial ant” problem follows [2]. The game takes place on the surface of torus with width and height equaling 32 (fig. 1). There are food in some cells (on the figure these cells are black). Ant starts moving from cell labeled *Start*.

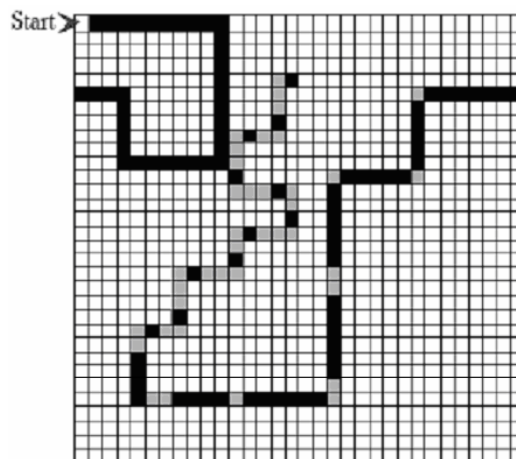


Fig. 1. Field

During one move ant may make one of the following actions:

- turn left;
- turn right;
- step forward and eat food if it is in the next cell;
- do nothing.

The game lasts for 200 moves. Goal of the game is to make ant “with minimal number of states”, which eats maximum

possible quantity of food in minimal number of moves.

Recently this problem was solved in paper [3] using only Mealy automaton with eight states. Minimal known Mealy automaton solving this problem consists of seven states, it was constructed in paper [8].

In this paper this problem is solved firstly using system of two Mealy interacting by nesting automata and then using Moore automaton.

The main advantage of using composition of automata is that it allows use less states than simple automaton for solving some problem.

III. USING OF GENETIC ALGORITHM

The same island genetic algorithm [5, 9] is used for generating pair of two nesting Mealy automata and Moore automaton. The only difference between this two cases is structure of chromosome, fitness function and operators of mutation and crossover. This algorithm can be also applied for generating system of automata in other problems.

In the first case individual is pair of nested Mealy automata, one of them we will call "external", the other we will call "internal". The chromosome of individual consists of description of two automata. Every automaton description contains index of initial state and descriptions of every state.

The description of state contains descriptions of two transitions. First of them describes case if there is food front of the ant, the second describes other case. For external automaton some of transitions may be not described (in other words, transition function of external automaton can be partly defined). In this case transition is performed in internal automaton, every transition of which is defined.

The description of transition contains index of ending state and action which is performed when this transition is chosen.

In the second case individual is Moore automaton. Chromosome consists of index of initial state and descriptions of every state. The description of state contains action which is performed when there was transition to the state, and the description of two transitions.

The description of transitions consists of index of ending state.

Chromosome is not coded into bit string, but is represented using Java object of following structure:

```
class Automaton {
    int initialState;
    Transition[][] transition;
    char[] stateAction[]; // only for
Moore automata
    Automaton nestedAutomaton;
}
```

Island genetic algorithm consists of following phases:

- initial generation creation;
- mutation;
- crossover;

- calculating of fitness function;
- exchange individuals between islands;
- selection of individuals for forming next generation.

Description of all phases of the algorithm with reference to "Artificial ant" problem follows.

IV. INITIAL GENERATION CREATION

Every island is filled randomly generated automata. Every automaton has the same predetermined number of states.

V. MUTATION (SMALL OR INDIVIDUAL'S MUTATION)

Every individual can mutate with predefined probability.

For pair of nesting Mealy automata it means:

- random modification of initial state;
- random selection of a transition and random modification of ending state;
- erasing (adding) of a randomly chosen transition for external automaton;
- modification of action on a randomly chosen transition;
- mutation of internal automaton.

For Moore automaton is applied similar operator of mutation:

- random modification of initial state;
- random selection transition and random modification of ending state;
- modification of action on a randomly chosen state;

Each of the actions listed above is performed with some predefined probability.

VI. MUTATION (BIG OR ISLAND MUTATION)

After predefined number of generations fixed part of islands is changed to randomly islands. It is impossible to execute this mutation when the fitness function lightly changes. Fitness function always changes her value because of exchange individuals between islands.

VII. CROSSOVER

Crossover operator takes two parent individuals and returns two children individuals.

Crossover operator of pair nested automata is similar to operator described in paper [8]. Nested automaton equiprobable cross or the first child inherits one nested automaton and the second inherits another nested automaton.

Crossover operator of Moore automaton is also similar to the operator described in paper [8], but it has some features. Denote parents $P1$ and $P2$ and children $S1$ and $S2$.

Denote action in k -th state in automaton A as $A.a[k]$. For each k on the following will be true:

- $S1.a[k] = P1.a[k], S2.a[k] = P2.a[k];$
- $S1.a[k] = P2.a[k], S2.a[k] = P1.a[k].$

These cases are equiprobable.

VIII. CALCULATING OF FITNESS FUNCTION

In this work for generating Moore automata fitness function of such kind is used:

$$F - \frac{T}{200}, F \text{ is number of ate food, } T \text{ is index of last move,}$$

on which ant ate food.

Experiments showed that such a function does not suit for generating system of Mealy automata because if we will use it we will get as the result external automaton in which nearly all transitions are not defined. Its action practically concludes in passing control to internal automaton. So, for generating a pair of nested Mealy automata fitness function of following kind is used:

$$F - \frac{T}{200} + C \cdot Z, F \text{ is number of food ate, } T \text{ is index of}$$

last effective move, Z is number of visited states in external automaton, and C is coefficient of influence. Selection of coefficient C is the most important for generating system of nested automata using genetic programming.

IX. NEXT GENERATION CREATION

The main strategy of next generation creation is elitism [11]. Every individual except some part of the most fit (elite) is thrown off. These individuals move to the next generation. After that the generation is added in predefined proportion by randomly individual, individuals from current generation which mutated, and results of crossover of individuals from current generation. Individuals, which “have the right” to cross, take part “in a duel”. Two individuals are selected, the more fit of them becomes one of the parents. Evolution performs on every island independently.

Every island exchanges with some other by some predefined number of randomly selected elite individuals in some predefined number of generations.

The big mutation (the island mutation) is performed also in some predefined number of generations.

In process of development of the algorithm one more instrument of selection was being investigated. We called it “war between islands”. Winner, for example, island with the greatest sum of fitness function values, destroys population of loser (may be, except elite) and populate by own inhabitants. But it does not speed up evolution because variety of individuals decreases.

X. PARAMETERS OF GENETIC ALGORITHM

Algorithm has several parameters, which can be tuned:

- number of islands;
- generation size on one island;
- number of generation between big mutations;

- part of effaceable islands during big mutation;
- percent of elite;
- number of generation between exchange of individual between of islands;
- number of exchangeable individual;
- parameters of small mutation;
- parameters of crossover;
- part of mutant in next generation;
- C – coefficient of external automaton influence in fitness function formula for systems of interacting Mealy state machines.

Values of these parameters which give optimal result is unknown. Only one thing is evident – coefficient C is the most important parameter for generation pair of nested Mealy automaton.

XI. MOORE AUTOMATON

Moore automaton with 10 states was created with the help of developed algorithm. It was created on 247-th generation. This automaton allows ant to eat all food in 198 steps (fig. 2).

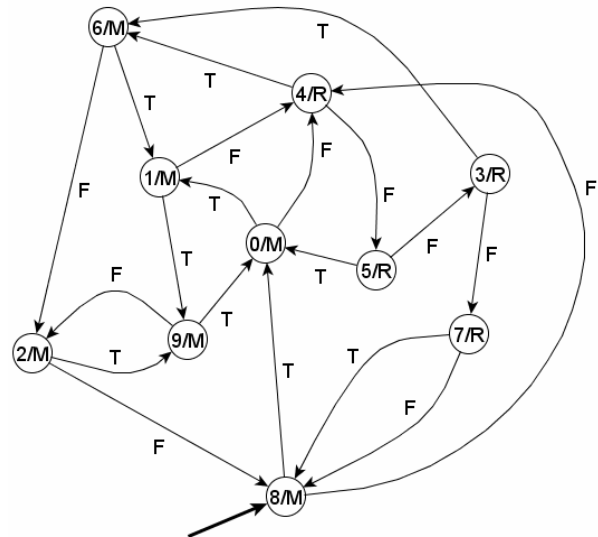


Fig. 2. Automaton which allows eating all food for 198 steps

Let us explain the notation used on this figure. Initial state is marked by an arrow. Vertices are marked by *number/action*. Action is denoted as:

- L – turn left;
- R – turn right;
- M – step forward and eat food if it is in the next cell;

One interesting property of this automaton is that it does not contain action “turn left”.

Conditions on transitions are given by:

- T – there is food in front of ant;
- F – there is no food in front of ant.

Moore automaton with nine states, which generated by this method, allows ant to eat only 86 units of food in 198 steps. The system was created on 268-th generation.

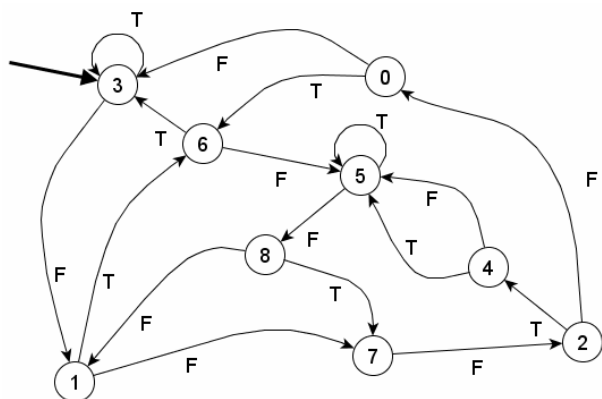


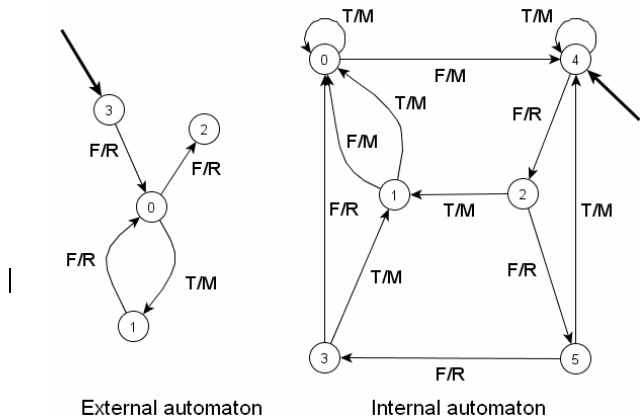
Fig. 3. Automaton which allows eating 86 units of food for 198 steps

XII. SYSTEM OF TWO INTERACTING MEALY AUTOMATA

Well-known Mealy automaton with seven states which solve viewing problem [8]. Authors decide that it makes sense to generate system of two automata, which have less than seven states in each automaton.

Authors successes in generating system of two Mealy automata (4, 6) (four states in external automaton and six in internal), while allow to eat only 87 units of food for 185 steps (fig. 3). This result is provided by C equals 0.001.

In this system internal automaton is put in each state of external and runs if the transition in external automaton does not described. Attention this ant, as in previous result, never turns left.



External automaton Internal automaton
Fig. 4 Example of system interacting Mealy automata

Explain the notation on fig.4. Vertex's label is given by *number*. Transition's labels are given by *condition/action*.

In the end note that this method can not decomposite automata, which solve describing problem.

XIII. CONCLUSION

In this work island genetic algorithm is proposed for generating system of Mealy automata and Moore automaton.

This algorithm was used for "Artificial Ant" problem and the Moore automaton with 10 states which solve this problem was created.

Even this method was used for solving rather easy problem, also it is used for solving more complex problems. Good example of problem of such kind is problem about «Flying Plates» [12].

REFERENCES

- [1] Shalyto A.A.. Technology of automata-based programming /Works of first All-Russian scientist confention "Methods and resources of processing information" M.: MSU. 2003.
- [2] Jefferson D., Collins R., Cooper C., Dyer M., Flowers M., Korf R., Taylor C., Wang A. The Genesys System. 1992. www.cs.ucla.edu/~dyer/Papers/AlifeTracker/Alife91Jefferson.html
- [3] Angeline P., Pollack J. Evolutionary Module Acquisition / Proceedings of the Second Annual Conference on Evolutionary Programming. 1993. <http://www.demon.cs.brandeis.edu/papers/ep93.pdf>
- [4] Chambers L. Practical Handbook of Genetic Algorithms. Complex Coding Systems. Volume III. CRC Press, 1999. [http://www.eknigu.com/info/Cs_Computer%20science/CsGn_Genetic.%20neural/Chambers%20D.L.%20\(ed.\)%20Vol.%203.%20Handbook%20of%20genetic%20algorithms.%20Complex%20coding%20systems%20\(CRC.%201999\)\(ISBN%200849325390\)\(T\)\(659s\).djvu](http://www.eknigu.com/info/Cs_Computer%20science/CsGn_Genetic.%20neural/Chambers%20D.L.%20(ed.)%20Vol.%203.%20Handbook%20of%20genetic%20algorithms.%20Complex%20coding%20systems%20(CRC.%201999)(ISBN%200849325390)(T)(659s).djvu)
- [5] Gladkov L. A., Kureichic V. V., Kureichic V. M. Genetic algorithms. M.: Physmathlit, 2006 (in Russian).
- [6] Russel S., Norvig P.. Artificial Intelligence. A Modern Approach. Prentice Hall. 2003.
- [7] Koza J. R. Genetic programming: on the programming of computers by means of natural selection. MA: MIT Press, 1992. [http://www.eknigu.com/info/Cs_Computer%20science/CsGn_Genetic.%20neural/Koza%20J.R.%20Genetic%20programming%20\(MIT.%201998\)\(T\)\(ISBN%200262111705\)\(609s\).djvu](http://www.eknigu.com/info/Cs_Computer%20science/CsGn_Genetic.%20neural/Koza%20J.R.%20Genetic%20programming%20(MIT.%201998)(T)(ISBN%200262111705)(609s).djvu)
- [8] Tsarev F. N., Shalyto A. A. About constructing automata with minimal number of states for "Artificial Ant" problem /Proceedings of X international conference by soft computing and measuring. SPbSU "Eltech". T.2, 2007, c. 88–91. (in Russian) http://is.ifmo.ru/download/ant_ga_min_number_of_state.pdf
- [9] Yaminov B. Genetic algorithms <http://rain.ifmo.ru/cat/view.php/theory/unsorted/genetic-2005>
- [10] Polykarpova N.I. Tochilin V.N., Shalyto A.A. Using of genetic programming for implementation system with hard behavior. /Scientific and technical bulletin. Effort over the range information technology. SPbSU IFMO. 2007. Edt. 39, c. 276–293. (in Russian) http://vestnik.ifmo.ru/ntv/39/ntv_39.3.3.pdf
- [11] De Jong K. An analysis of the behaviour of a class of genetic adaptive systems. PhD thesis. Univ Michigan. Ann Arbor, 1975.
- [12] Paracshenko D. A. Shalyto A. A. Tsarev F. N. The technology of simulation of class of multiagent system on basis of automata-based programming by the example of game "Flying Plate's Competition" (in Russian) <http://is.ifmo.ru/download/plates.pdf>