Evolutionary Mobile Agents

Assistant Barna Iantovics
"Petru Maior" University of Tg. Mureş

Abstract*. We denote evolutionary agents the agents that can solve problems using methods based on evolutionary computation. The evolutionary mobile agents represent a subclass of the evolutionary agents. In this paper, we propose the endowment of the mobile agents with a novel architecture with evolutionary problem solving capabilities. The mobile agents obtained this way can solve efficiently difficult problems using evolutionary problem solving methods.*
Keywords*: evolutionary computation, agent architecture, mobile agent, evolutionary agent.*

1 Introduction

Mobile agents are computational software processes capable of roaming wide area networks such as WWW, interacting with foreign hosts, gathering information on behalf of their users and coming back having performed the *problems* (*tasks*) solving duties set by their users [12]. Each host offers different services to the mobile agents. The problems solving methods are described in the mobile agents' body. A mobile agent body is composed from a software cod and different data. The cod describes the overtaken problems solving. The data are used in the overtaken problems solving. Denote *specialization* the method that describes the solving of a problem. A mobile agent at a host in collaboration with the host solves the problems. The host executes the specifications described in the mobile agent body. A mobile agent may interrupt a problem solving at a host. The solving of an interrupted problem solving must be continued at another host. In such situations, the mobile agent body must contain the states of the interrupted problem solving. The main advantage of the mobile agents is the low data transmission in the network where they operate. As examples of applications of the mobile agents, we mention [15]: the fields of manufacturing planning, Internet traffic management, and Internet search.

An *agent architecture* is essentially a map of the internals of an agent, its data structures, the operations that may be performed on these data structures, and the control flow between these data structures. In the paper [9] a novel mobile agent architecture is proposed. In this paper, we propose

BDD-A23870 © 2005 Editura Universităţii „Petru Maior"
Provided by Diacronia.ro for IP 216.73.217.142 (2026-11-14 23:03:55 UTC)

the endowment of the mobile agents with the proposed architecture with specializations based on evolutionary problem solving methods. The evolutionary mobile agents that can be obtained this way can solve problems for which, solving methods based on evolutionary algorithms exist. The problems solving using the proposed evolutionary mobile agents eliminate some disadvantages of the traditional methods based on evolutionary computation, which can be used by static agents.

2 Evolutionary algorithms

Evolutionary computation includes *genetic algorithms* [7, 8], *evolutionary programming* [6], *evolution strategies* [1] and *genetic programming* [11]. Many problems can be solved with methods based on *evolutionary algorithms* [2, 15]. Often evolutionary methods are used in searching and optimization problems. Evolutionary search and optimization is a powerful technique of problem solving, and is applicable to a wide variety of practical problems that are nearly intractable with other conventional optimization techniques [2]. Practical evolutionary search schemes do not guarantee convergence to the global optimum in a predetermined finite time, but they are often capable of finding very good approximate solutions. However, they are shown to asymptotically converge under mild conditions [13, 14]. The principles of the methods based on evolutionary computation define a general paradigm that is based on a simulation of natural evolution.

Evolutionary algorithms perform their search by maintaining at any time t a population $P(t)$ of individuals (chromosomes).

$$P(t) = \{P_1(t), P_2(t), \ \ldots \ P_n(t)\}.$$

Each $P_i(t) \in P(t)$, is a potential solution to the original problem. Genetic operators are applied to create the new and desirably more superior population $P(t+1)$. Usually a *mutation* and a *recombination* operator are used. A mutation operator introduces genetic variations in the population by randomly modifying some of the building blocks of individuals. The recombination operator combines more parents (members of the population) obtaining one or more offspring. The problem solution searching process continues until a terminal condition is satisfied. The choice of an appropriate representation of the solutions is often a critical step in a successful application of evolutionary algorithms. Closely linked to the choice of representation of solutions, is the choice of a fitness function f.

$$f: X \to R.$$

The function f assigns credit to candidate solutions. Individuals in a population are assigned fitness values according to some evaluation criterion. Fitness values measure how well individuals represent solutions to the problem. Highly fit individuals are more likely to create offspring by

BDD-A23870 © 2005 Editura Universităţii „Petru Maior”
Provided by Diacronia.ro for IP 216.73.217.142 (2026-11-14 23:03:55 UTC)

recombination and mutation operations. At each evolutionary step, a breadth search of increasingly optimal subregions of the options space is performed.

3 The proposed evolutionary mobile agents

In the paper [9] a novel intelligent mobile agent architecture is proposed. The mobile agents endowed with the proposed architecture can solve many difficult problems [9]. In the proposed agent architecture, elements from the papers [3, 4, 5, 16] have been used. A proposed mobile agent *MOBILE AGENT* is composed from two parts: a static part *STATIC PART* and a mobile part *MOBILE PART*.

$$MOBILE\ AGENT = STATIC\ PART + MOBILE\ PART.$$

The static part of a mobile agent is a static subagent. The notion of *subagent* is defined in the paper [9]. The static subagent is responsible for overtaking of the problems from the user. The static subagent creates mobile subagents. The mobile subagents represent the mobile part of the mobile agent. The mobile subagents have all the proprieties of the known mobile agents. The mobile subagents are responsible for the problems solving. The body of a mobile subagent contains the description of the problems solving methods. Figure 1 illustrates the proposed mobile agent architecture. In Figure 1, the following notations are used: *S* represents the static part of the mobile agent; M_1, M_2, … M_n represents the mobile part of the mobile agent.

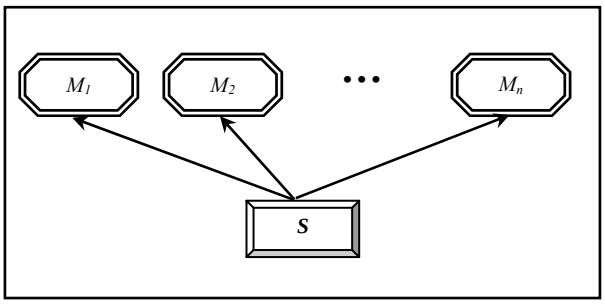


Fig. 1. *The proposed mobile agent architecture*

In the following, we propose the endowment of the mobile agents, with the architecture described in the paper [9], with evolutionary problem solving capabilities. A proposed evolutionary mobile agent can use more problem solving methods based on evolutionary algorithms. The mobile agent can solve a set $C = \{c_1, c_2, ... c_m\}$ of *classes* (*types*) of problems. Each problem *p* member of a class of problems c_i can be solved using an evolutionary algorithm.

The knowledge bases of a proposed mobile agent are detained by the static subagent. The static subagent has three knowledge bases denoted: *specializations*, *rules* and *network*.

The specializations knowledge base contains a set *S* of specializations.

BDD-A23870 © 2005 Editura Universităţii „Petru Maior”
Provided by Diacronia.ro for IP 216.73.217.142 (2026-11-14 23:03:55 UTC)

$$S = \{s_1, s_2, \ldots s_m\}.$$

Each specialization s_k is an evolutionary problem solving method, which can be used for solving problems members of a class c_k of problems.

The rules knowledge base is formed by a set of rules of the following form:

$$type \rightarrow specialization.$$

Where: *type* represents the description of a class of problems; *specialization* represents the necessary specialization for the problems solving members of the class *type*.

The network knowledge base contains different knowledge about the network where the mobile agent operates. As examples of knowledge that can be detained in the network knowledge base, we mention: the informations about some hosts from the network, the informations about the network. In the case of a host, the network knowledge base may contain informations like: the problems that can be solved at the host; the services offered by the host; the intervals of time when the host is usually overloaded etc. About the network it may contains informations like: the topology of a part of the network, latency, bandwidth etc. If the mobile agent operates in a WAN, then the network knowledge base does not contain all the necessary knowledge. Different difficulties in the information collections from a WAN are analyzed in the paper [16].

In the following, we describe how a proposed evolutionary mobile agent operates. The *evolutionary mobile agent algorithm* describes how a proposed mobile agent solves the overtaken problems at a problem solving cycle. A problem solving cycle consists in the overtaking and solving of a set of problems.

Algorithm - Evolutionary Mobile Agent

Step 1

The static subagent overtakes the problems $PR = \{pr_1, pr_2, .. pr_k\}$ transmitted by the user.

Step 2

The static subagent establishes the necessary mobile subagents $MS = \{ms_1, ms_2, .. ms_j\}$ for the overtaken problems solving.

While (not all the established mobile subagents are created) *do*

{

The static subagent selects an uncreated mobile subagent *ms* from the set *MS*.

The static subagent creates and launches the mobile subagent *ms*, which consist in:

- establishing the problems from the set *PR* that have to be solved by the mobile subagent *ms*. The static subagent establishes the necessary specializations in the overtaken problems solving. Endows the mobile subagent *ms* with the established specializations (evolutionary problem solving methods);

BDD-A23870 © 2005 Editura Universităţii „Petru Maior”
Provided by Diacronia.ro for IP 216.73.217.142 (2026-11-14 23:03:55 UTC)

- establishing the itinerary *I* that have to be traversed by the mobile subagent *ms*. The itinerary contains: the hosts that have to be visited; the order in which the hosts have to be visited; the problems that have to be solved at each host;
- placing the itinerary to the mobile subagent *ms*;
- locating the first host specified in the itinerary *I*. The mobile subagent is transmitted to the located host.

}

Step 3

Each mobile subagent executes the following actions:

- visits all the hosts specified in its itinerary. The hosts are visited in the order established in the itinerary.
- at each visited host the problems are solved. The results obtained after the problems solving are transmitted to the static subagent.

Step 4

The static subagent collects the results transmitted by the mobile subagents. If it is necessary the static subagent can process the results, the solutions obtained this way are transmitted to the user.

End

The static subagent is responsible for the problems overtaking from the user. After the problems overtaking, the static subagent creates the mobile subagents. Each mobile subagent is responsible for the solving of one or more problems. The mobile subagents operate as the known mobile agents. Each mobile subagent is endowed with one or more evolutionary problem solving methods, which describes the overtaken problems solving. Each overtaken problem by a mobile subagent is solved at a host (the host executes the problem solving description contained in the mobile agent's body). The problems solutions are transmitted by the mobile subagents to the static subagent after they are obtained.

The mobile subagents can communicate with the static subagent during their operation [10]. At the endowment of a mobile subagent with specializations, some specializations may be absent [3]. The absent specializations are transmitted to the mobile subagent just when and where they are necessary. If a mobile subagent needs a specialization at a host, and the specialization is not present in his body, then the mobile subagent must query the specialization from the static subagent (the subagents can communicate). This operating manner is useful when the mobile subagent must visit numerous hosts, the specializations size is large and the mobile subagent needs the specialization at few hosts.

BDD-A23870 © 2005 Editura Universităţii „Petru Maior”
Provided by Diacronia.ro for IP 216.73.217.142 (2026-11-14 23:03:55 UTC)

4 Simulation results

We have realized simulations of a mobile agent endowed with the proposed mobile agent architecture [9] using a developed simulation environment. The simulations' purposes were to compare the performance of the traditionally used mobile agents with the performance of the mobile agents endowed with the proposed mobile agent architecture. The performance evaluation consists in the necessary time for solving all the overtaken problems at a problems solving cycle. The simulation environment parameters consist in the simulated hosts and the specializations of each host. Each host is endowed with a specializations set.

A proposed mobile agent may use more mobile subagents at each problems solving cycle in the overtaken problems solving. For each generated mobile subagent the nearly optimal itinerary in the overtaken problems solving time using an evolutionary algorithm is determined. A mobile subagent itinerary consists in the hosts that must be visited by the subagent and problems that must be solved at each host. Using the evolutionary algorithm different itineraries are generated in the case of each created mobile subagent. The best mobile subagents' itineraries obtained during the searching process represent the solution.

Figure 2 presents simulation results of a mobile agent endowed with the proposed architecture realized in the simulation environment containing 30 hosts. In the simulation environment are used a set of 100 specializations distributed between the hosts.

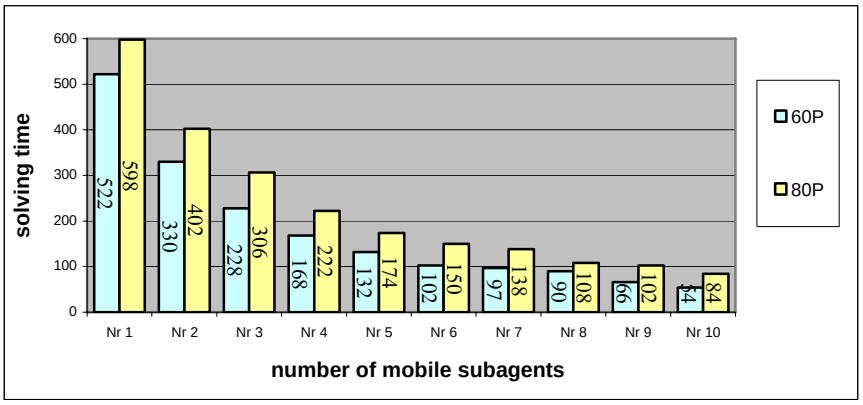


Fig. 2. Simulation results of 60 and 80 problems solving

The simulations were realized for different sets of overtaken problems with different necessary solving time, different overloading degree of the hosts. Figure 2 illustrates how the average problems solving time measured in seconds, of 60 and 80 overtaken problems is changing using from 1 to 10 mobile subagents. The simulation results presented in Figure 2 illustrates that the problem solving time with the use of 2, 3, 4 mobile subagents decrease substantially as opposed to the use of a single mobile subagent, the use of more subagents decrease the problems solving time

BDD-A23870 © 2005 Editura Universităţii „Petru Maior"
Provided by Diacronia.ro for IP 216.73.217.142 (2026-11-14 23:03:55 UTC)

in a smaller degree than the previous number of mobile subagents. The results obtained during the simulations illustrate that more mobile subagents can solve faster a large number of overtaken problems than a single mobile subagent. How many mobile subagents are necessary in solving efficiently a set of overtaken problems depends on the network and hosts resources overloading degree and the parameters of the problems (type, number, difficulty etc.) which must be solved.

We have realized simulations to compare the performance of a mobile agent with a mobile agent endowed with the proposed agent architecture. In the simulation environment are used a set of 100 specializations distributed between 30 hosts. The simulations were realized for different overloading degree of the hosts and different sets of overtaken problems.

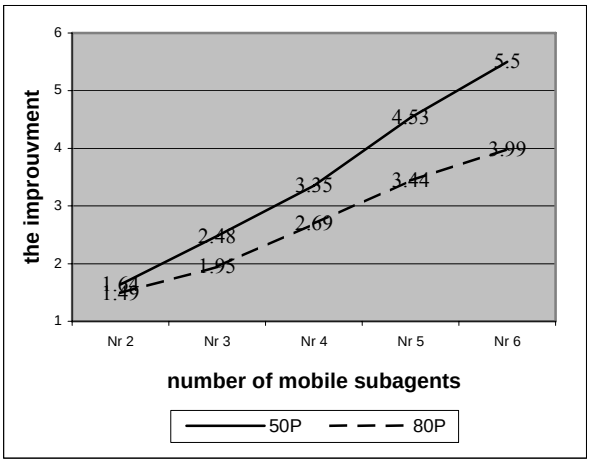


Fig. 3. A mobile agent versus a proposed mobile agent

Figure 3 illustrates how many times the problems solving time is improved in average using a mobile agent endowed with the proposed architecture that uses from 2 to 6 mobile subagents in the solving of 50 and 80 problems as opposite to a traditionally used mobile agent. The simulations show that a mobile agent endowed with the proposed architecture that uses more subagents can solve faster a large number of problems that have to be solved at more hosts than a traditionally used mobile agent.

6 Conclusions

The mobile agents can be considered a relatively new paradigm in the area of distributed programming and a useful supplement of traditional techniques like the Client/Server architecture. In this paper, we have proposed the endowment of the mobile agents, with the architecture proposed in the paper [9], with evolutionary problem solving capabilities. The evolutionary mobile agents that can be obtained in this way can solve efficiently problems that can be solved using methods based on evolutionary computation. This new evolutionary problem solving is not a new problem

BDD-A23870 © 2005 Editura Universităţii „Petru Maior”
Provided by Diacronia.ro for IP 216.73.217.142 (2026-11-14 23:03:55 UTC)

solving method, just different in the way that existent evolutionary methods are used in the problems solving.

The paper [10] describes some improvements of the mobile agent architecture. There is also analyzed the communication, cooperation and learning capabilities of the proposed mobile agents. Mobile agents endowed with the proposed architecture can form intelligent cooperative multiagent systems.

In the following, we mention some of the advantages of the problems solving using a proposed evolutionary mobile agent versus the traditionally used problems solving methods based on evolutionary computation.

The mobile agent can solve a large variety of problems autonomously. Each problem type has an evolutionary problem solving method associated. The mobile agent can picks up autonomously the corresponding evolutionary solving method to a problem (the problem type define the necessary problem solving method).

The mobile agent can solve a large number of problems efficiently. If a host is overloaded, some mobile subagent can migrate to another host where they can solve the problems. The mobile subagents can solve problems in the same time at different hosts.

The evolutionary mobile agent can solve distributed problems in the network. Each distributed problem in the network can be solved by a mobile subagent of the mobile agent using a method based on evolutionary computation.

The mobile agent can solve efficiently composed problems. A composed problem can be decomposed in simple subproblems. Each subproblem can be solved by a mobile subagent at a host. This way, more mobile subagents solve simultaneously parts of the composed problem.

Bibliography

[1]. Bäck, T.; *Evolutionary Algorithms in Theory and Practice*, Oxford University Press, New York, 1996.

[2]. Dumitrescu, D.; *Algoritmi Genetici şi Strategii Evolutive - aplicaţii în Inteligenţa Artificială şi în domenii conexe*, Albastră, Cluj Napoca, 2000.

[3]. Erfurth, C., Braun, P., Rossak W.R.; *Migration Intelligence for Mobile Agents*, Artificial Intelligence and the Simulation of Behaviour Symposium on Software mobility and adaptive behaviour, AISB'01, University of York, United Kingdom, March 2001, pp. 81-88.

[4]. Erfurth, C., Rossak, W.R.; *Characterization and management of dynamical behaviour in a system with mobile agents*, LNCS 2346, Proceedings of the Innovative Internet Computing

BDD-A23870 © 2005 Editura Universităţii „Petru Maior"
Provided by Diacronia.ro for IP 216.73.217.142 (2026-11-14 23:03:55 UTC)

System - Second International Workshop, IICS'02, Kühlungsborn, Germany, June 2002, pp. 109-119.

[5]. Erfurth, C., Rossak, W.R.; *Autonomous Itinerary Planning for Mobile Agents*, Proceedings of the Third Symposium on Adaptive Agents and Multi-Agent Systems AISB'03, University of Wales, Aberystwyth, Great Britain, April 2003, pp. 120-125.

[6]. Fogel, L.J., Owens, A.J., Walsh, M.J.; *Artificial Intelligence Through Simulated Evolution*, John Wiley, New York, 1966.

[7]. Goldberg, D.E.; *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Massachusetts, 1989.

[8]. Holland, J.H.; *Adaptation in Natural and Artificial Systems: an introductory analysis with applications to biology, control, and artificial intelligence*, Cambridge, Massachusetts, MIT Press, third edition, University of Michigan, 1994.

[9]. Iantovics, B.; *Intelligent Mobile Agents*, Proceedings of the Symposium "Zilele Academice Clujene", Babeş-Bolyai University of Cluj-Napoca, 2004, pp. 67-74.

[10]. Iantovics, B.; *A new intelligent mobile multiagent system*; IEEE International Workshop on Soft Computing Applications, IEEE – SOFA 2005, Szeged-Hungary and Arad-Romania 27-30 August, 2005 (accepted paper).

[11]. Koza, J.; *Genetic programming: On the programming of computers by means of natural selection*, Cambridge, Massachusetts, MIT Press, 1992.

[12]. Nwana, H.S.; *Software Agents: An Overview*, Knowledge Engineering Review, 11(3), October / November 1996, pp. 205-240.

[13]. Subbu, R.V.; *Network Decision Support based on Distributed Coevolutionary Algorithms*, Ph.D. Dissertation, New York, Rensselaer Polytechnic Institute, Troy, 2000.

[14]. Subbu, R., Sanderson, A.C.; *Modeling and convergence analysis of distributed coevolutionary algorithms*, Proceedings of the IEEE Congress on Evolutionary Computation, 2000.

[15]. Subbu, R., Sanderson, A.C.; *Distributed Search and Decision-Making Using Cooperative Coevolutionary Agents*, Proceeding of the AAAI Spring Symposium on Information Refinement and Revision for Decision Making: Modeling for Diagnostics, Prognostics and Prediction, Stanford, CA, March 2002.

[16]. Xu, H.; *A model-based approach for development of multi-agent software systems*, PhD Thesis, the Graduate College of the University of Illinois at Chicago, 2003.

BDD-A23870 © 2005 Editura Universităţii „Petru Maior"
Provided by Diacronia.ro for IP 216.73.217.142 (2026-11-14 23:03:55 UTC)