

Designing Global Behavior in Multi-Agent Systems Using Evolutionary Computation

Marko Privošnik

*University of Ljubljana, Faculty of Computer and Information Science, Tržaška 25, 1000 Ljubljana, Slovenia
Email: marko.privosnik@fri.uni-lj.si*

Abstract. Designing a multi-agent system by specifying individual agents and their local interaction in a way that will give rise to a desired global behavior is a difficult task. In this paper, an approach is presented to designing a global emergent behavior for the heap formation task in a reactive multi-agent system using genetic algorithms. Instead of building a multi-agent system by defining the agents' reaction rules, the system is designed using evolution of the global behavior by genetically operating on single agents while evaluating fitness of the whole system. The research includes examination of scalability of the evolved solutions relative to the number of the agents by cross-testing evolved solutions in different system configurations. It is shown that the evolved solutions perform properly, but scale well only on limited intervals that do not span over a critical point corresponding to the condition where the number of the agents equals the number of the objects.

Keywords: multi-agent systems, evolutionary computation, emergent properties

Načrtovanje globalnih lastnosti reakcijskih večagentnih sistemov z uporabo evlucijskega računanja

Načrtovanje večagentnih sistemov z neposrednim planiranjem posameznih agentov in njihovega vzajemnega delovanja je težak problem. V članku predstavljamo postopek načrtovanja globalnega emergentnega vedenja večagentnega sistema z uporabo genetskih algoritmov na primeru kopičenja predmetov. Načrtovanje sistema temelji na evlucijskem razvoju globalnega vedenja z genetskim delovanjem na ravni posameznega agenta in vrednotenjem na ravni celotnega večagentnega sistema. Raziskava je vsebovala tudi preučitev skalabilnosti pridobljenih rešitev v razmerju do števila agentov v sistemu. Raziskava je pokazala, da so bile pridobljene rešitve ustrezne, a so pri spremembi števila agentov delovale dobro le na omejenem intervalu, ki se ni raztezal preko točke, ko je bilo število agentov enako številu predmetov.

1 INTRODUCTION

Certain natural multi-agent systems, like ant colonies, prove that decentralized systems based on simple constituent parts can exhibit an advanced global behavior that has important advantages over explicit central control in multi-agent systems. Multi-agent systems are systems composed of multiple interacting computational entities, named agents, and their environment. Multi-agent systems can be used to solve a range of different problems [1] [2] [3], wherein the distinctive quality of the multi-agent systems are the agents' mutual interactions and the emerging overall effects that overcome properties and capabilities of the single agents. The agents that constitute a multi-agent

system can be sophisticated and complex computer programs, or in contrast, a multi-agent system can be composed of very simple software agents.

A special case of multi-agent systems are reactive multi-agent systems that contain a large number of simple reactive agents [4] [5] [6]. The reactive agent behavior consists of simple reactions to its local environment following very basic rules. The reactive agent does not include any communication capabilities other than ability to sense and change its local environment. Even though the reactive agent behavior is elementary, a reactive multi-agent system as a whole is capable of exhibiting a complex global behavior. However, designing a desired global behavior by defining the agents' reaction rules is a difficult task, since the global behavior in a reactive multi-agent system is in general an emergent phenomenon [7]. The emergent phenomena are not linear and therefore it is very difficult to predict the global behavior of the whole system on the basis of the behavior of single agents.

Instead of building a reactive multi-agent system by explicitly defining the agents' reaction rules, the system can be designed using evolution of the global emergent phenomena using genetic algorithms (GA) [8] by genetically operating on single agents, but then evaluating fitness of the whole system. This approach represents a shift from the conventional method where the genetic activity and fitness evaluation operate at the same level of the system. For instance, in the field of artificial life, the development of an ecosystem is

typically performed by evaluating and operating at the same level, i.e., at the level of an organism [9] [10].

A solution resulting from the described optimization process is evolved in a particular configuration of the system parameters (i.e., the number of the agents, the ratio between the number of the agents and the number of the environment cells, etc.). However, an optimized solution that evolves using one set of the system parameters does not necessarily provide a good solution under a different set of the system parameters.

The objective of the presented research was to develop optimization methods for constructing the desired global behavior of a homogeneous reactive multi-agent system entirely by evolution of the agents' local behavior. The required task was to produce a heap formation behavior [11]. The resulting multi-agent systems were tested for performance scalability relative to the change in the number of the agents while keeping the rest of the system parameters invariable.

The heap formation task examined in this research is well covered in the scientific literature. Deneubourg et al. [12] presented a computer simulation of various tasks inspired by ants' behavior. They showed that simple agents can perform coordinated clustering of scattered objects using only simple local rules and without any explicit communication between the agents. Beckers, Holland and Deneubourg [1] achieved clustering by using small, puck-piling robots with an even simpler algorithm. A similar research was done by Chantemargue and Hirsbrunner [13] who achieved regrouping of objects in the environment by virtue of self-organization and emergence, and by Barfoot and D'Eleuterio [14] who used a cellular automaton to arbitrate between a number of fixed basis behaviors to drive agents. Another simulation-based research exploring emergent cooperation was done by Dagaëff et al. [15]. They used an environment composed of a discrete torus grid, a set of objects, and mobile agents charged to gather all the objects on a single cell. Arkin and Balch dealt with the heap formation task using behavior-based robotics [16].

To date, there has not been much work done concerning the problem of scalability of the evolved solutions in multi-agent systems. Bahceci and Sahin [17] studied how different parameters of evolutionary methods affect the performance and scalability in the swarm robotic systems. Some other relevant research on scalability in multi-agent systems was done by Fontan and Mataric [18] that studied the issue of the critical mass in a multi-agent retrieval task and found that effectiveness of a group of robots depends significantly on the group size. Further, Rosenfeld et al. [19] presented a study on productivity of the robotic groups during scaling-up. They introduced an interference metric that measures the total time the robots take to deal with resolving the team conflicts and reducing the group productivity.

2 METHODS

2.1 Reactive multi-agent system

The experimental multi-agent system consists of a set of identical reactive agents, a set of passive objects that the agents can manipulate and a multicellular environment.

The core of each agent is a finite state machine (FSM) which transforms the agent's input perceptions into its output actions. Perceptions sensing and actions execution are implemented by the perception and execution function (see Fig. 1).

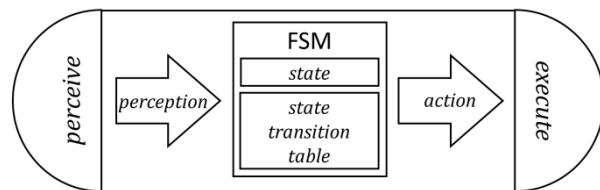


Figure 1. Reactive agent connected with its environment by converting perceptions into actions.

The FSM transition function is represented by a state transition table. This type of the lookup table mapping between perceptions and actions is called reactive control. The agent with a reactive control is called reactive agent. The agent's FSM assigns action $o \in O$ to each perception $p \in P$ and internal state $s \in S$:

$$fsm: P \times S \rightarrow O \quad (1)$$

where P is a set of all possible agent perceptions, S is a set of FSM internal states, and O is a set of all possible actions that the agent can perform. Other than the FSM internal state, reactive agents have no presentation of the universe in which they operate, thus they merely react to their local environment.

The perception function assigns perception $p \in P$ to each local environment state $\in \Sigma$:

$$perception: \Sigma \rightarrow P \quad (2)$$

where Σ is a set of all possible states of the agent's environment. The agents are orientated in the environment and their local environment is defined as a cell in front of the agent. The perception function distinguishes between the following elementary perceptions: an empty cell, a cell with an agent, a cell with a small pile of objects, and a cell with a large pile of objects. The size of the pile is determined probabilistically according to the number of the objects in the pile and the total number of the objects in the system. The elementary perception is combined with a random bit to form a combined final perception.

The execution function puts into effect a new local environment state $\sigma' \in \Sigma$ as a result of execution of action $o \in O$ on local environment state $\sigma \in \Sigma$

$$execution: O \times \Sigma \rightarrow \Sigma \quad (3)$$

The execution function performs the following actions: doing nothing, turning left, turning right, stepping forward, picking up an object, and putting down the object.

The agents are entirely defined by FSM, the perception and the execution function, together with the corresponding sets of perceptions and actions. All the agents in the same experimental multi-agent system have an identical FSM and share the same perception and execution function, hence they are identical.

The agents' environment is a discrete two-dimensional toroidal grid of cells. A single cell can be empty or can hold a pile of objects, a single agent, or an agent transporting an object. The time advances in discrete steps. In every time step, for all the agents in the environment, a percept–react–execute sequence is performed. The order in which the agents are processed is not explicitly determined.

2.2 Evolutionary optimization

Construction of the desired global behavior of a reactive multi-agent system is formulated as an optimization process using a genetic algorithm. In order to use a genetic algorithm to solve an optimization problem, a corresponding optimization task has to be defined. In our case, the problem of the global system behavior design is determined as the following optimization task: "Find the agents' FSM transition function that minimizes the distance between the resulting global behavior and the desired global behavior of the multi-agent system." The corresponding objective function $f_o: S_p \rightarrow \mathbb{R}$ mapping the search space to the set of real numbers, cannot be expressed analytically. The objective function reflects some global property of the whole system that is in general an emergent property. An emergent property is by definition computationally irreducible and its outcome can essentially be found only by explicit simulation.

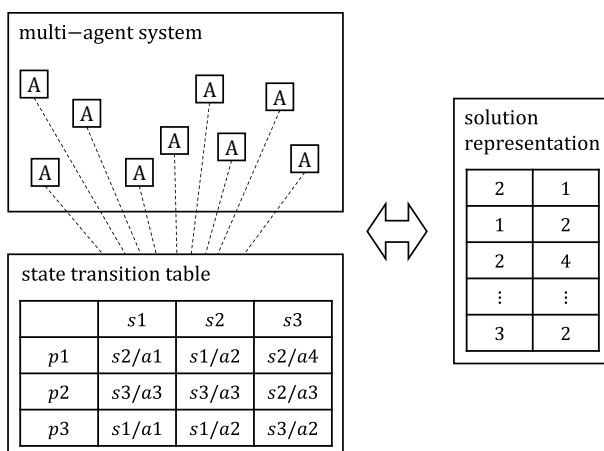


Figure 2. Representation of a particular solution as a string of integers representing the state transition table, being the same for all the agents.

In an experimental homogeneous reactive multi-agent system, a single reactive behavior pattern defined by FSM applies to all the agents and therefore every solution in the search space can be represented by a single state transition table. Throughout the evolutionary process, the number of the FSM internal states is fixed. Consequently, the size of the search space is equivalent to the number of all the possible different transition tables of the same size determined by the number of the FSM internal states, the number of the possible perceptions and the number of the possible actions.

The transition table represented in a chromosome appears as a string of integers that correspond to the internal states and actions indices (see Fig. 2).

3 EXPERIMENTS AND RESULTS

The goal of our experiments was to study an evolutionary construction of a desired global behavior of a homogeneous multi-agent system by genetically operating on the agents' local reactive behavior. The global task was to collect the objects into piles.

Our experiments involved a multi-agent system with a discrete two dimensional toroidal grid of the size 64×64 cells. The agents and objects were initially randomly positioned. The experiments varied in the number of the agents and objects involved. The fitness function measured the relative number of the objects included in the biggest pile at the end of the evolutionary optimization process.

The experiments were performed in two stages. In the first stage, the evolutionary optimization process was tested on different configurations varying in the number of the FSM internal states. These preliminary experiments were used to examine validity of the proposed methods for evolutionary construction of the multi-agent global behavior. In the second stage, experiments were conducted to evaluate scalability of the evolved solutions.

3.1 Preliminary optimization examination

The experimental multi-agent system contained 64 randomly positioned objects and 32 randomly positioned reactive agents. All the agents had the same FSM. The fitness function measured the fraction of the objects included in the biggest pile averaged over the last 27 simulation steps. Simulations were left to run until stalling for the 27000 agents' steps.

The GA population of 100 representations of the FSM transition table was used. The number of generations in the evolution was 250. In the course of the same experiment, the FSM complexity (the number of internal states) was fixed and only the FSM state transition table was evolved. The crossover probability used in the evolution process was 0.04 and the mutation probability was 0.01.

In the experiments, different configurations were tested. They differed in the number of the FSM internal states. Fig. 3 shows the fitness values of the best multi-agent systems in the first 250 generations, where 1, 2, 4, and 8 states FSM were used. The experiments showed a clear progress of the evolved systems. Also noted was that the higher FSM complexity resulted in an improved outcome of the evolutionary process.

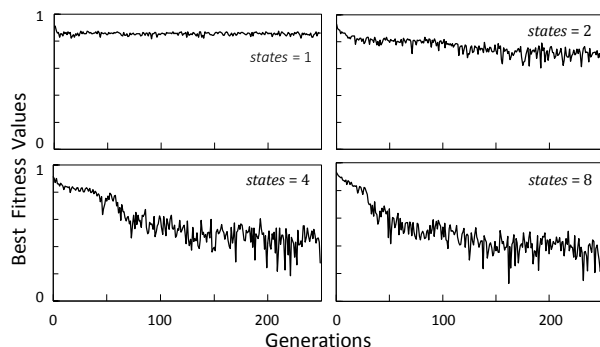


Figure 3. Best fitness values in 250 evolution generations for multi-agent configurations using FSM with 1, 2, 4 and 8 internal states.

3.2 Scalability of the evolved solutions

Scalability of an evolved multi-agent system refers to the ability of the system to respond to the changes of its particular parameter. For example, the scalability variable can be the size of the problem. The scalability variable considered in our research was the number of the constituent agents, while keeping the rest of the system parameters invariable.

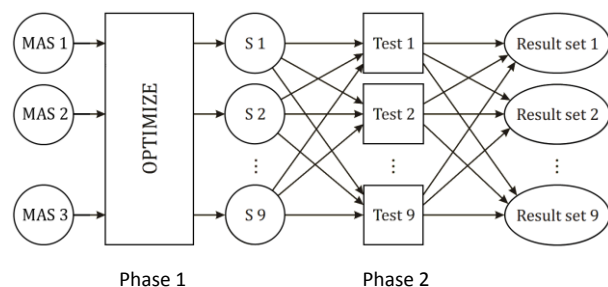


Figure 4. Scalability evaluation of the evolved multi-agent systems. For every system configuration, optimized solutions are sought in the first phase. In the second phase, these solutions are tested under all system configurations including configurations different from those used in the evolution.

Scalability was evaluated in two phases (see Fig. 4). In the first phase, best solutions for different system configurations were found using GA with the same optimization process as described in Section 3.1. In the second phase, the best evolved solutions were cross-tested in all the system configurations.

3.2.1 Optimization process

The first phase consisted of the optimization process where best solutions to the task were searched by using GA. The experimental setup was a variation of the setup used in the basic optimization process. In order to study the impact of the ratio between the agents and objects in the optimization phase on scalability of the optimized solution, nine system configurations with a different agent/object ratio were examined as shown in Table 1.

Table 1: System configurations used in the scalability-evaluation optimization step and the scalability-evaluation test step

Number of Agents	Number of Objects	Agents/Objects Ratio
1	54	0.02
13	54	0.24
27	54	0.50
40	54	0.74
54	54	1.00
68	54	1.26
81	54	1.50
95	54	1.76
108	54	2.00

A GA population of 108 representations of the FSM transition table was used. The number of the generations in evolution was 324. The number of the internal states was 4. The crossover fraction used in the evolution process was 0.5 and the mutation probability was 0.02.

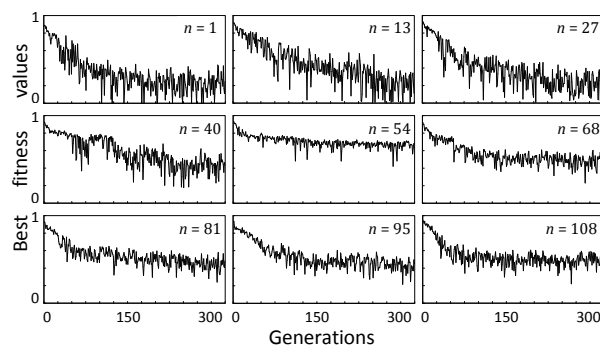


Figure 5. Best fitness values in the evolutionary process for nine different experimental system configurations. Parameter n indicates the number of the agents in the system. The number of the objects is 54 for any of the system configurations.

In a set of experiments, an evolution process was executed on different configurations varying in the agent/object ratio. Fig. 5 shows the fitness values of the best evolution solutions. As seen, different configurations produced significantly different outcomes of the evolutionary process.

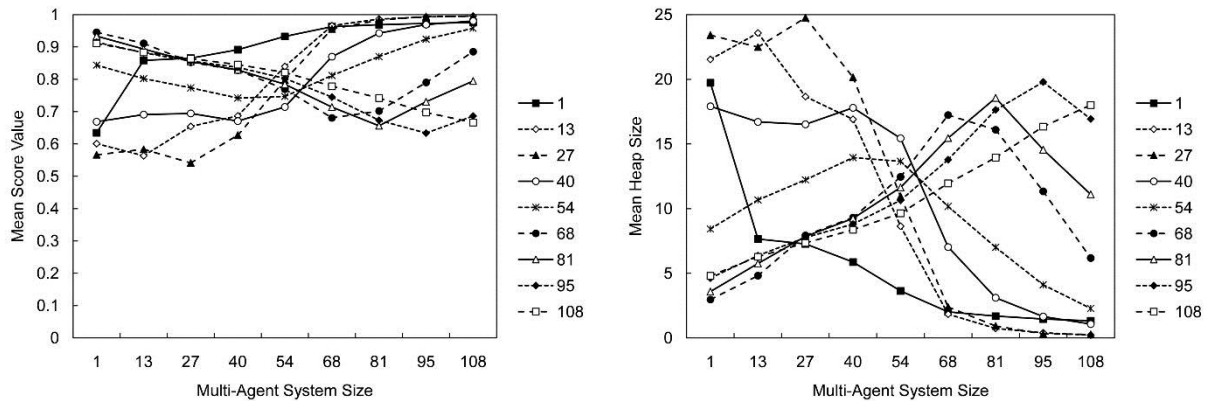


Figure 6. Results of the scalability tests expressed as mean score values (left graph) and as mean pile sizes (right graph). The horizontal axis represents the multi-agent system size used for cross-testing scalability of the obtained solutions. The different lines represent the optimized solutions for a particular multi-agent system size.

3.2.2 Scalability Test

The results of the first phase were nine FSM sets with their corresponding fitness values. The best three FSMs for each examined configuration passed to the second experimental phase where the scalability tests were performed. The scalability tests measured the performance of the best FSMs from the first phase in each of the multi-agent system configurations. The rest of the system parameters were invariable.

A single scalability test consisted of nine sets of simulations, each testing one of the system configurations. Each set consisted of 108 random runs for each of the best three solutions from the first experimental phase. The simulations in the scalability test were the same as the ones used in the optimization process. The averaged results of the scalability tests are shown in Fig. 6.

In addition to cross-testing the evolved solutions with the same simulations as used in the optimization phase, the solutions were further cross-tested with another type of simulations. For these simulations, the agents had to form a pile of the size equal to 7. The measured value was the number of the agents' steps for the task to be completed (see Fig. 7). The agents' steps were limited to 1,000,000. Note that the number of the agents' steps and not the number of the simulation steps was considered in further scalability tests while comparing the different evolved solutions from the first phase. In this way, it was the effective work that was compared rather than the time taken by the system to complete the task.

4 DISCUSSION AND CONCLUSION

Allowing global coordination to emerge from collection of simple components has important advantages over explicit central control in both the natural and the multi-agent systems (i.e., robustness, flexibility, and

scalability). On the other hand, complex emergent systems are highly unpredictable and difficult to design.

Our preliminary experiments confirm that evolutionary optimization using GA can be used to design global emergent properties of a reactive multi-agent system. We show that the performance of the evolutionary approach to designing reactive multi-agent systems is much dependent on the complexity of the agents' FSM, where the FSM complexity is determined by the number of the FSM internal states. The simplest agents do not develop a successful behavior, while more complex agents do develop a useful behavior.

As expected, our scalability experiments show that the evolved solutions perform well if the agent/object ratio of the multi-agent system used in testing is similar to the agent/object ratio of the multi-agent system used in the evolution. On the basis of the measured data, a critical point can be identified, at which performances of different solutions differ minimally, while far from the critical point some solutions perform significantly better than others. Typically, the solutions performing well on one side of the critical point perform badly on the other side, suggesting that the solutions performances are significantly dependent on the agent/object ratio. The critical point corresponds to the boundary of the critical condition that occurs where the number of the agents is equal or greater than the number of the objects.

Additional scalability tests showed substantial changes in the solutions performances in the critical condition interval with rapid changes at the critical point. Manifestation of the critical condition can be explained by the fact that when the number of the agents surpasses the number of the objects, the agents will probably at some time carry all the objects so that none remains on the ground as a seed for the pile. This situation is difficult to be dealt with, especially for the solutions that are evolved in the systems where a critical condition cannot occur, i.e., in systems with the agent/object ratio lower than 1. On the other hand, it is not evident why the solutions that evolve in large multi-

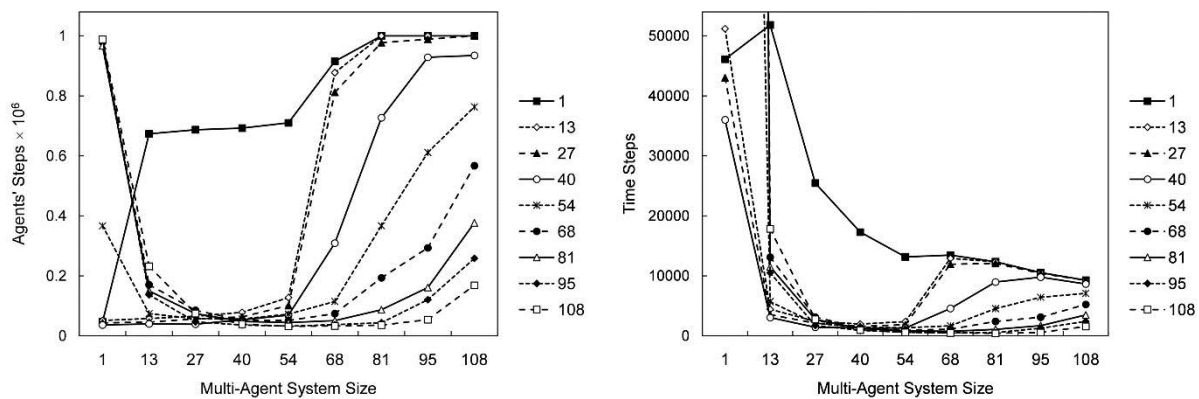


Figure 7. Results of the additional scalability tests measured in the agents' steps (left graph) and in the simulation time steps (right graph). The horizontal axe represents the multi-agent system size used for cross-testing the solutions scalability. The different lines represent the optimized solutions for a particular multi-agent system size.

agent systems perform badly in small multi-agent systems. This question needs to be examined in further research.

In the presented study, we show that evolutionary computation can be used to build a reactive multi-agent system with a required global behavior. Unfortunately, the study demonstrated that such evolved solutions may scale poor under the conditions that differ from those used in evolution. Further research is needed on evolutionary optimization techniques that will produce solutions that will scale well under the system configuration changes.

REFERENCES

- [1] R. Beckers, O. E. Holland, J. L. Deneubourg, "From local actions to global tasks: Stigmergy and collective robotics", in R. Brooks, P. Maes (eds.), *Proceedings of the Fourth Workshop on Artificial Life*, MIT Press, Cambridge, Massachusetts, USA, pp. 181-189, 1994.
- [2] C. R. Kube, E. Bonabeau, "Cooperative transport by ants and robots", *Robotics and Autonomous Systems*, Vol. 30, No 1-2, Elsevier Science BV, Amsterdam, Netherlands, pp. 85-101, 2000.
- [3] S. T. Kazadi, "Swarm Engineering", *IEEE Connections*, Vol. 2, No. 2, pp. 7, 2004.
- [4] J. Ferber, "Simulating with Reactive Agents", in E. Hillebrand, J. Stender (eds.), *Many Agent Simulation and Artificial Life*, IOS Press, Amsterdam, Netherlands, pp. 8-28, 1994.
- [5] M. Wooldridge, *An Introduction to MultiAgent Systems*, John Wiley & Sons, 2002.
- [6] M. Privošnik, *Evolutionary Computation and Synergism in Massive Multi Agent Systems*, Ph. D. thesis, Faculty of Computer and Information Science, University of Ljubljana, 2002.
- [7] V. Darley, "Emergent Phenomena and Complexity", in R. Brooks, P. Maes (eds.), *Artificial Life IV; Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems*, MIT Press, 1994.
- [8] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. The MIT Press, Cambridge, MA, 1992.
- [9] C. G. Langton (ed.), *Artificial Life*, Santa Fe Institute Studies in the Sciences of Complexity, Addison-Wesley, 1989.
- [10] A. D. Channon, R. I. Damper, "Evolving Novel Behaviors via Natural Selection", in *Artificial Life VI: Proceedings of the Sixth International Conference on Artificial Life*, MIT Press, 1998.
- [11] E. Mayr, *Toward a new Philosophy of Biology: observations of an evolutionist*, Cambridge, MA, USA: Belknap Press, 1988.
- [12] J. C. Deneubourg, S. Gross, N. R. Franks, A. Sendova-Franks, C. Detrain, L. Chretien, "The Dynamics of Collective Sorting: Robot-like Ants and Ant-like Robots", in J. A. Meyer, S. W. Wilson (eds.), *Simulation of Adaptive Behaviour: From Animals to Animats*, Cambridge, Massachusetts, USA: MIT Press, pp. 356-365, 1991.
- [13] F. Chantemargue, B. Hirsbrunner, "A collective robotics application based on emergence and self-organization", *Proceedings of Fifth International Conference for Young Computer Scientists (ICYCS'99)*, 1999.
- [14] T. Barfoot, G. M. T. D'Eleuterio, "An Evolutionary Approach to Multi-agent Heap Formation", in *Proceedings of the Congress on Evolutionary Computation*, 1999.
- [15] T. Dagaëff, F. Chantemargue, B. Hirsbrunner, "Emergence-based Cooperation in a Multi-Agent System", *Proceedings of the Second European Conference on Cognitive Science (ECCS'97)*, pp. 91-96, 1997.
- [16] R. C. Arkin, T. Balch, "Cooperative Multiagent Robotic Systems", in D. Kortenkamp, R. P. Bonasso, R. Murphy (eds.), *Artificial Intelligence and Mobile Robots*, Cambridge, Massachusetts, USA: MIT Press, pp. 278-296, 1998.
- [17] E. Bahgeci, E. Sahin, "Evolving Aggregation Behaviors for Swarm Robotic Systems: a Systematic Case Study", *Swarm Intelligence Symposium, 2005*, Proceedings, IEEE, 2005.
- [18] M. S. Fontan, M. J. Mataric, "A Study of Territoriality: The Role of Critical Mass in Adaptive Task Division", in P. Maes, M. Mataric, J.-A. Meyer, J. Pollack, and S. Wilson (eds.), *From Animals toAnimats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*, MIT Press, pp 553-561, 1996.
- [19] A. Rosenfeld, G. A. Kaminka, S. Kraus, "A Study of Scalability Properties in Robotic Teams", in P. Scerri, R. Vincent, and R. Mailler (eds.), *Coordination of Large-Scale Multiagent Systems*, Springer, pp. 27-51, 2006.

Marko Privošnik received his B.Sc., M.Sc. and Ph.D. degrees in 1992, 1997 and 2002, respectively, all from the Faculty of Computer and Information Science, University of Ljubljana. His research interests include evolutionary computation and emergent properties in complex systems.