Comparative Reproduction Schemes for Evolving Gathering Collectives

A.E. Eiben, G.S. Nitschke, M.C. Schut

Computational Intelligence Group
Department of Computer Science, Faculty of Sciences
De Boelelaan 1081a, 1081 HV Amsterdam
The Netherlands
gusz@cs.vu.nl, nitschke@cs.vu.nl, schut@cs.vu.nl

Abstract. This research investigates an evolutionary approach to engineering agent collectives that accomplish tasks cooperatively. In general, reproduction and selection form the two cornerstones of evolution and in this paper we study various reproduction schemes in an evolving population of agents. We classify reproduction schemes in temporal and spatial terms, that is, by distinguishing when and where agents reproduce. In terms of the temporal dimension, we tested schemes where agents reproduce only at the end of their lifetime or multiple times during their lifetime. In terms of the spatial dimension we distinguished locally restricted reproduction (agents reproduce only with agents in adjacent positions) and panmictic reproduction (when an agent can reproduce with any other in the environment). This classification leads to four different reproduction schemes, which we compare, via their overall impact upon collective performance. Results using two completely different types of evolvable controllers (hand-coded or neural-net based) indicate that utilizing single reproduction at the end of an agent's lifetime and locally restricted reproduction afforded the agent collective a significantly higher level of performance in its cooperative task.

1 Introduction

The research theme of this paper is described by the term: *Emergent Collective Intelligence* (ECI). The end goal of ECI research is to combine and exceed achievements in multi-agent systems [1], swarm intelligence [2], and evolutionary computation [14] research via developing synthetic methodologies such that groups of computationally complex agents produce desired emergent collective behaviors resulting from the bottom-up development of certain individual properties and social interactions. This paper investigates certain technical aspects of artificial evolution as means of achieving adaptability at the local level and desired emergent behavior at the global level.

In many multi-agent tasks, such as those common to multi-robot systems, the correct input-output mappings for the agents' controllers are not known in advance so it is not possible to program or train them with supervised learning methods [3]. To solve this problem many researchers have used neuro-evolution [4] as a generalized methodology for adaptability in agent behavior. Many researchers have highlighted that neuro-evolution is most appropriately applied to complex problems that are neither effectively addressed via pure artificial evolution nor neural processing approaches [5], [6], [7]. For example, Gomez [8] devised the enforced sub-populations (ESP) neuro-evolution methodology that was used for deriving the correct input-output mappings for the agents' controllers in the learning of multi-agent control tasks [9] with small numbers of agents. ESP has been shown to work well for various discrete-state applications such as the game *Go*, as well as the classical pole-balancing task [10]. Neuro-evolution approaches investigated in collective robotic systems such as RoboCup soccer [11], simulated pursuit-evasion tasks [12], and multi-agent computer games [13] were only able to derive limited forms of cooperative behavior, and the behavior did not scale with the number of agents.

Our application domain is the gathering of renewable resources from an environment. This gathering task is divided into *locating*, *retrieving*, and *transporting* the resources in question. It is an essential assumption that this task is interfaced to the population of agents via fitness rewards that are given after delivering the resources to a given 'home area'. Additionally, we distinguish resources with different values and postulate that gathering of higher value (more complex) resources necessitates a higher degree of cooperative behavior (more agents). The performance evaluation criterion for the agent collective as a whole is then the *total value gathered cooperatively*, measured at the final generation of the simulation. Clearly there are many particular applications fitting into this general description. One could think of collecting some renewable resources, for example: harvesting farming produce, or minesweeping. We use the minesweeping example throughout this paper. That is, we consider the task of the agent collective to be the location, and extraction of different types of mines, and their transportation to the home area within a simulated mine field. The successful delivery of a mine to the home area is equated with a fitness reward and fitness rewards are proportional to the type and amount of mines gathered.

Our approach to developing successful agents for this task is evolutionary; in particular, we evolved agent controller parameter values. Hence, we studied two different types of agent controllers, one heuristic controller with evolvable parameters, and a neural net controller with the same set of evolvable parameters and evolvable connection weights. The technical research goal of this paper was to compare the efficacy of different agent reproduction scheme settings for accomplishing the minesweeping task. We classified reproduction schemes in temporal and spatial terms, that is, by distinguishing when, with which agents a given agent reproduces. For the temporal dimension, the agent reproduction schemes we tested were termed: *Single Reproduction at the End of the Agent's Lifetime* (SREL) and *Multiple Reproductions During an Agent's Lifetime* (MRDL). For the spatial dimension, we distinguished locally restricted reproduction (agents reproduce only with agents in adjacent positions) and panmictic reproduction (when agents reproduce with other agents located anywhere in the environment). This classification led to four different reproduction schemes, which we compared experimentally, using the collective performance of the population accomplishing its task as the basic measure.

2 Collective Behavior Design

The experiments utilized a simulated minefield and an initial population of 1000 agents, placed at random positions on a grid-cell environment with a 50 x 50 resolution. A maximum of four agents could occupy any given grid-cell within the environment. A home area spanning 4 x 4 grid-cells was randomly placed within the environment. This home area was where gathered mines were taken. Gathering was the term applied to the process of locating, extracting, transporting, and delivering a mine to the home area. Within the simulated minefield there were three types of mines: *type A*, *type B* and *type C*. The different types of mines had differing *values* to reflect the difficulty (degree of cooperation) associated with gathering it. The cost of gathering mines comprised two sub-costs: the cost of extracting a mine from its location in the environment, and the cost of transporting a mine to the home area. The costs of extracting and transporting one unit of each of the three mine types are presented in *table 1*. The transport cost was applied per unit being transported, and per grid-cell traversed. Initially, a quantity of between 0 and 3 mines of each type were randomly initialized and placed within each grid-cell. It is assumed that a long-term process of gathering and replenishment in a minefield is being simulated, where mines are considered a renewable resource, and each mine type is renewed at a rate of 3 per simulation iteration. That is, the simulation is of a long-term process of collective gathering behavior being evolved, whilst an unseen competitor renews gathered mines. Additionally, it is assumed that an agent never triggered a mine to detonate.

In order to gather the different mine types a degree of cooperative behavior was necessitated. Cooperation was necessary when at least one agent was attempting to extract a given mine type, and the value of the prevalent agent controller parameter was too low for the agent to individually gather the mine. These prevalent agent controller parameters were termed: *Mine type A capacity*, *Mine type B capacity*, *Mine type C capacity* and *transport capacity*, and provided an indication of the capability of an agent for gathering a particular mine type. Specifically, to gather *one unit* of a particular mine type, the sum of the values of the *capacity* parameter for that mine type (for all agents simultaneously attempting to extract the mine) must exceed a given *capacity threshold*. These capacity thresholds are presented for each mine type in *table 1*. The task of each agent was to gather the highest possible value of mines during the course of its lifetime. This task was interfaced to the agent collective by providing fitness rewards for gathered mines.

The fitness rewards for gathering one unit of the different mine types are presented in *table 1*. The total value of mines that all agents gathered in cooperation with at least one other agent during the course of its lifetime was termed the *value gathered cooperatively*. Further to playing its conventional role in survivor selection, fitness was also used as a metaphor of energy (actions cost fitness). An agent was able to move one grid-cell in any direction per simulation iteration at a cost of one unit of fitness.

	Capacity Threshold	Extraction Cost	Transport Cost	Fitness Reward
Mine type A	**300**	**8**	**0.04**	**20**
Mine type B	**150**	**4**	**0.02**	**10**
Mine type C	**75**	**2**	**0.01**	**5**

Table 1. The capacity thresholds, and the costs for extracting and transporting mines, as well as the fitness reward for gathering one unit of the different mine types.

Value Ranges	Initial	Minimum to Maximum
Parameters: Not Evolvable		
Sight	1	1
Death_Age	[20..100]	[20..100]
Min_Fit_Reproduction	50	50
Parameters: Evolvable		
Mine Type A Capacity (CA)	[0..100]	[0..Infinity]
Mine Type B Capacity (CB)	[0..100]	[0..Infinity]
Mine Type C Capacity (CC)	[0..100]	[0..Infinity]
Transport Capacity (CT)	[0..300]	[0..Infinity]

1

```
IF AmA < CA THEN
  IF ( Holding + AmA ) < CT THEN extract AmA
 ELSE IF AmB < CB THEN
       IF ( Holding + AmB ) < CT THEN extract AmB
   ELSE IF AmC < CC THEN
         IF ( Holding + AmC ) < CT THEN extract AmC
     ELSE Look-Ahead

Look-Ahead:
IF end of life and SREL active THEN reproduce
IF at home THEN unload mines transported
  IF MRDL active THEN reproduce
IF transporting a quantity of mines THEN move to home
ELSE IF mine type A detected THEN move to mine type A
 ELSE IF mine type B detected THEN move to mine type B
   ELSE IF mine type C detected THEN move to mine type C
    ELSE move to a random cell
```

2

Fig. 1. Evolvable and non-evolvable agent controller parameters. **Fig. 2.** Heuristics utilized by agents operating under the pure-evolution approach. *AmA*, *AmB*, and *AmC* denote the amount of mine type *A*, *B* and *C*, respectively, on a given grid-cell. *Holding* denotes the current amount of all mine types a given agent is currently transporting. *CA*, *CB*, *CC* and *CT*, denote the gathering capacities for mine types *A*, *B*, and *C*, and the transport capacity, respectively.

2.1 Pure-Evolution Approach

For the evolution of agent controller parameter values, a standard evolutionary algorithm was used [14]. When an agent initiated reproduction, the fittest partner (with the highest energy) of *m* potential partner agents was selected for reproduction. The population initially contained 1000 individuals (agents), and the genotype of each agent was its set of gathering and transport capacities (evolvable parameters illustrated in *figure 1*). These parameter values directly influenced the heuristic agent lifetime behavior, though the behavioral heuristics (*figure 2*) remained static over the course of the evolutionary process. That is, once an agent had gathered as many mines as it could transport, it would begin transporting the mines back to the home area. During reproduction, agent controller heuristics (*figure 2*) were copied from parent to child, and the fitness inherited by a child was the average fitness of the two parent agents. Ninety percent of the inherited fitness was then subtracted from each parent's fitness.

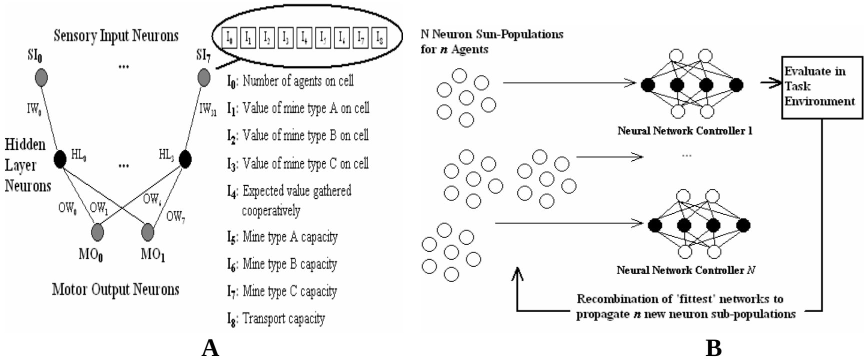

Fig. 3A. Neural network agent controller (neuro-evolution approach). Note that, sensory input neurons SI_1 to SI_6 , hidden layer neurons HL_1 and HL_2 , input weights IW_1 through to IW_{31} , and output weights OW_2 through to OW_5 are not presented. **B.** Neuro-evolution approach.

2.2 Neuro-Evolution Approach

Figure 3A presents the feed-forward neural network agent controller operating under the neuro-evolution approach. Input-output weights connecting the hidden layer neurons from sensory input neurons to motor output neurons were evolved over successive generations under the neuro-evolution process. Agent controller parameter values (the evolvable parameters illustrated in *figure 1*) were evolved over successive generations using a standard evolutionary algorithm [14]. Evolved parameter values were then used as part of the sensory input (*figure 3A*) of the next generation of agents. Thus, as with the pure-evolution approach, the initial population contained 1000 individuals (agents), where the genotype of each agent was the set of input-output weights for the hidden layer of the neural network controller (*figure 3A*), and the set of gathering and transport parameters (*figure 1*). F*igure 3A* presents the neural network controller as having 8 sensory input nodes (SI_0 through to SI_7) to account for 8 surrounding grid-cells, 4 hidden layer nodes (HL_0 through to HL_3), and 2 motor output nodes (MO_0 and MO_1) to account for the *x, y* position that the agent moves to. As illustrated in *figure 3A*, each sensory input neuron (SI_0 through to SI_7) was comprised of a 9 value input array. The first four inputs of the array (I_0, I_1, I_2, I_3) correspond to: the *number of agents* observed on the given grid-cell, the value of *mine type A*, *mine type B*, and *mine type C* observed on the given grid-cell, respectively. The fifth value of the sensory input neuron (I_4) was the expected value to be gathered cooperatively. The neural network operated via attempting to select actions that minimized error. Error was the difference between expected value to be gathered cooperatively at simulation time *t*, and actual value gathered cooperatively at simulation time $t+1$. The final four values (I_5, I_6, I_7, I_8) were the *mine types A, B, C* and *transport* capacities of this agent. The evolvable aspects were the 40 input-output weights connecting hidden layer neurons, and the gathering and transport capacities of the agent.

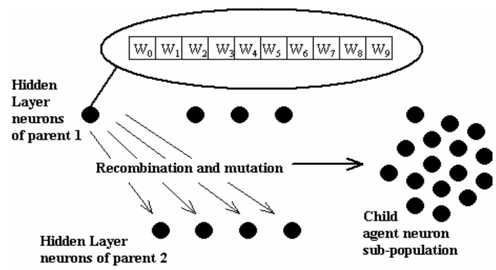

Fig. 4. Neuron reproduction to produce a new child agent sub-population. Note that, only the input-weights of the first hidden layer neuron of parent 1 are presented.

In the neuro-evolution approach, as presented in *figure 3B*, individual neurons for neural network controllers were evolved as a result of being evaluated, and recombined in a social context. As illustrated in *figure 3B*, n individual controllers are initially derived by randomly selecting u neurons from each sub-population as the u neurons for the hidden-layers of n controllers. The genetic representation of each sub-population neuron is a string of input and output weights for each hidden-layer neuron.
That is, the approach evolved partial solutions (neurons) that were recombined in novel ways so as to form complete solutions (a group of heterogeneous neural networks). Combinations of hidden layer neurons from two parent agents formed a child sub-population (16 neurons) from which a child network was derived (4 neurons for the hidden layer). *Figure 4* presents the 10 input-output weights of each hidden layer neuron (w_0 to w_9). During reproduction, those in the first parent were recombined (via single-point crossover) and each weight mutated (0.05 probability) with hidden layer neurons in the second parent. This allowed for recombination and mutation of the hidden neuron input-output weights, and produced a new sub-population, from which the fittest 25% of neurons were selected as the hidden layer of a child network.

The key idea of this methodology was that over the course of multiple generations, cooperation occurs within the sub-populations themselves and competition between the n sub-populations so as to produce neural network controllers that operate effectively at addressing both individual and social tasks. The key difference delineating this approach from other neuro-evolution methodologies (most notably: ESP [8]) is that it provides a separate sub-population of neurons for the derivation of each neural network. Also, after each evaluation, fitness is assigned to all neurons within a network, and networks can reproduce with any other network in the population of networks. The approach was split into several phases. In the initial phase, the first time the n neural network controllers are created, the genotype of each neuron (set of input-output weights) is randomly initialized and u neurons are then randomly selected from each of the n sub-populations in order to form the hidden-layer of n neural network controller. The n neural network controllers are then evaluated in the mine sweeping task. Fitness values are awarded to each agent when a mine is delivered to the home area. This fitness is then equally distributed to each hidden-layer neuron participating in the agent's neural network controller.

2.3 Reproduction Schemes and Settings

The experiments compared the four agent reproduction schemes, specifically, SREL and panmictic, SREL and locally restricted, MRDL and panmictic, as well as MRDL and locally restricted reproduction. Each of these schemes was tested and evaluated for a heuristic agent controller with evolvable parameters (operating under a pure-evolution approach), and a neural network controller with evolvable parameters (operating under a neuro-evolution approach).

During the reproduction action, 90% of the fitness of two parent agents was divided amongst and passed onto *p* offspring agents. During reproduction only one partner agent of *m* potential partner agents was selected for reproduction. An agent's fitness could only be replenished when it delivered a mine to the home area. The precondition for locally restricted reproduction setting was that there was at least one potential partner agent in the same grid-cell or an adjacent grid-cell. For either the locally restricted or panmictic settings, reproduction was only possible when both parents current fitness was greater than the value of the *min fit reproduction* parameter.

When *p* offspring agents were produced using the panmictic reproduction setting, each offspring would be placed in a random free grid-cell adjacent to one of the parents. The chance that an offspring an agent was placed in a grid-cell adjacent to parent 1 was 0.5, and the chance that an offspring was placed in a grid-cell adjacent to parent 2 was 0.5. If no adjacent grid cells were free, then the offspring agent died. Using the locally restricted setting offspring agents were always placed in a random free grid-cell adjacent to the parent agent that initiated reproduction. The number of offspring to be produced was determined as *m* = the total amount of fitness to be inherited (*x*) divided by 10. According to the reproduction scheme setting being used, pairs of agents produced *p* offspring using the genetic operations of crossover and mutation [14]. For both the pure evolution and neuro-evolution approaches, the core of reproduction was the application of uniform crossover to 'recombine' the controller parameters: *mine type A, B, C* and *transport capacities* of two parent agents in order to derive the agent controller parameter values of a child agent. The uniform crossover operator selected a parameter value to be inherited from either parent agent with a 0.5 probability. Child controller parameter values were mutated by a value of either plus or minus 10 with a probability of 0.05. If mutation occurred, the probability of adding versus subtracting 10 from the inherited parameter value was 0.5.

3 Experiments, Results, and Discussion

The four agent reproduction schemes were tested and evaluated under the pure-evolution and neuro-evolution approaches. 100 independent runs (each executed for 2000 iterations) were performed. For each of the four reproduction schemes operating under the pure-evolution and neuro-evolution approaches, a control experiment was performed. Each control experiment was non-evolutionary, using static values for the gathering and transport agent controller parameters. The static values utilized were those attained at the end of the evolutionary process (pure-evolution or neuro-evolution) using a given reproduction scheme. The performance criterion for evolved agent collective behaviors was the *total value of mines gathered cooperatively*.

Table 2 presents the values gathered cooperatively attained for the four reproduction schemes, operating under the pure-evolution and neuro-evolution approaches. For each approach, the values attained in the control experiments are presented below the values gathered cooperatively. The value in parentheses presented next to each of the values gathered cooperatively is the standard deviation. A high standard deviation indicates that the agent collective was less stable in its gathering behavior. High standard deviations were the result of many agent populations (of the 100 replications) becoming extinct before the end of a simulation. A low standard deviation indicates a low portion of agent populations dying out prematurely and hence a high stability in the gathering task. Here, the term *stability* indicates that, for the gathering and transport parameter values evolved, a particular value gathered cooperatively (plus or minus some variance) was expected.

The control experiments demonstrated that both the pure-evolution and neuro-evolution approaches (using the SREL and locally restricted reproduction scheme) were operating within a region of the parameter space (defined by the four agent controller parameters) where a high value gathered cooperatively was attainable. This was especially the case for the neuro-evolution approach, which, when using the SREL and locally restricted, and SREL and panmictic reproduction schemes, was able to attain values gathered cooperatively over an order of a magnitude higher than comparative agent collectives.

Also, *table 2* highlights that, agents using the SREL and panmictic reproduction scheme and operating under the neuro-evolution approach, were able to achieve a higher stability comparative to the other reproduction schemes. This is theorized to be a result of the panmictic reproduction scheme that selects a partner agent from anywhere in the environment.

Under the neuro-evolution approach, panmictic reproduction encourages and preserves the heterogeneity and diversity in the *n* sub-populations corresponding to the *n* agent controllers. Locally restricted reproduction restricts the diversity produced in child sub-populations (hence agent controllers) by only selecting from agent sub-populations local to the proximity of the reproducing agent.

Under the pure-evolution approach, all agent controllers are initialized with the same heuristics, and the agent controllers do not evolve over successive generations. This heterogeneity of controllers under the neuro-evolution approach, and the homogeneity of controllers under the pure-evolution approach, refers only to the structure of the agent controllers, and not to the evolvable parameters (as used in both approaches).

The result of the SREL and locally restricted agent reproduction scheme being most appropriate for both approaches (pure-evolution and neuro-evolution) is theorized to be consequent of agents only reproducing at the end of their lifetimes. Using the SREL setting, agents that have performed their task well and have thus survived until the end of allotted lifetime, are allowed reproduce. Given that the reproduction action costs 90% of the parents' energy, agents using the MRDL setting have less of a chance of producing offspring that are well suited to successful task accomplishment.

	SREL Panmictic	SREL Local	MRDL Panmictic	MRDL Local
Neuro-Evolution	**159.67** (12.96)	**300.95** (46.56)	**37.92** (7.75)	**30.61** (4.90)
Control	**610.23** (9.10)	**870.67** (60.34)	**92.91** (3.67)	**68.50** (2.93)
Evolution	**23.59** (33.37)	**39.10** (17.20)	**32.56** (10.00)	**22.85** (17.60)
Control	**60.25** (1.85)	**71.70** (3.50)	**43.04** (5.46)	**54.28** (0.63)

Table 2. The values attained for the total value gathered cooperatively (standard deviations in parentheses) under pure-evolution and neuro-evolution. Values attained in the control experiments are presented under the respective approach and reproduction scheme setting used.

This is especially the case for the neural-evolution approach, since neural network controller weights need sufficient time to change and produce an effective agent behavior, in order for that behavior to be propagated in the next generation of agents. In the case of the heuristic controller, child agents inherit only recombined and mutated agent parameter values and an average of parent fitness. However, the nature of the SREL setting holds, in that only agents with appropriate controller parameter settings will have survived until the end of their allotted lifetime (that is, those agents with a high fitness).

Hence, *table 2* illustrates that for the pure-evolution and neuro-evolution approaches, the SREL and locally restricted reproduction scheme is the most appropriate for the given task. It is theorized the superior performance of the neuro-evolution approach is a result of agent lifetime behavior adapting over successive generations, and no direct reliance upon controller parameter values. The heuristic behavior under the pure-evolution approach relies directly upon the values of the gathering and transport capacities in order for an agent to decide where to move and what type of mine can be gathered.

Furthermore, as a benchmark to illustrate the efficacy of evolved agent controller parameter values, additional control experiments were run using the four reproduction schemes under the pure-evolution and neuro-evolution approaches. These control experiments utilized the maximum possible values (at initialization) for the gathering and transport parameters. That is, 100, 100, 100, and 300 for the *mine type A*, *B*, *C*, and *transport* capacities respectively.

The resulting values gathered cooperatively (average taken over 100 runs) were always low with high standard deviations (comparative to values attained in other experiments) for collectives using the pure-evolution approach. The low values and high standard deviations for each of the reproduction scheme settings operating under the pure-evolution approach indicate that all agent populations died prematurely.

Under the neuro-evolution approach, low values gathered cooperatively and high standard deviations were attained, indicative of few collectives (of the 100 replications) surviving until the final simulation iteration. This was a result of high values for the agent controller gathering and transport capacities (*table 1*) yielding correspondingly high gathering and transport costs, where these costs usually exceeded an agent's fitness.

4 Conclusions

This paper compared the efficacy of different agent reproduction scheme settings for accomplishing a cooperative gathering task. Results indicated that agent collectives utilizing the single reproduction at end of lifetime (SREL) and the locally restricted reproduction scheme yielded a superior performance in a collective gathering task. This agent reproduction scheme setting attained the highest performance in terms of the evaluation criterion for both a heuristic agent controller (operating under a pure-evolution approach) and a neural network agent controller (operating under a neuro-evolution approach). The evaluation criterion was defined as the total value of resources gathered cooperatively in a simulated environment within a given time period.

References

1. Avouris, N., Gasser, L.: Distributed Artificial Intelligence: Theory and Praxis. Kluwer Academic Publishers. Dordrecht (1992).
2. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press, Oxford (1998).
3. Hertz, J., Krogh, A., & Palmer, R.: Introduction to the Theory of Neural Computation. Addison-Wesley. Redwood City (1991).
4. Gomez, F., & Miikkulainen, R (1997).: Incremental evolution of complex general behavior. *Adaptive Behavior*, vol. 5(1): 317-342.
5. Nolfi, S., Baldassarre, G., & Parisi, D.: Evolution of collective behavior in a team of physically linked robots. In Gunther, R., Guillot, A, Meyer, J. (Eds.), *Applications of Evolutionary Computing* (pp. 581-592). Springer-Verlag: Heidelberg (2003).
6. Michel, O.: Evolutionary Neuro-genesis Applied to Mobile Robotics. In, Honovar, V. (Ed.). *Advances in Evolutionary Synthesis of Neural Systems*. MIT Press: Cambridge (1999).
7. Floreano, D. & Urzelai, J. (2000). Evolutionary Robots with on-line self-organization and behavioral fitness. *Neural Networks*, vol. 13(1): 431-443.
8. Gomez, F. Robust Non-Linear Control through Neuro-evolution. PhD Thesis. Department of Computer Sciences. University of Texas: Austin (2003).
9. Gomez, F., Miikkulainen, R. Solving Non-Markovian Control Tasks with Neuro-evolution. In Dean, T. (Ed.), *Proceedings of the International Joint Conference on Artificial Intelligence*, (pp. 1356-1361). Morgan Kaufmann: Stockholm (1999).
10. Moriarty, D., Miikkulainen, R. (1997). Forming Neural Networks through Efficient and Adaptive Co-evolution. *Evolutionary Computation*, 5(4): 373-399.
11. Whiteson, S., Kohl, N., Miikkulainen, R., Stone, P. Evolving Keep-away Soccer Players through Task Decomposition. In *Proceeding of the Genetic and Evolutionary Computation Conference*, (pp. 356-368). Springer-Verlag: Berlin (2003).
12. Yong, C., & Miikkulainen, R.: Cooperative Co-evolution of Multi-Agent Systems. *Technical Report*. Computer Sciences Department, University of Texas. Austin, USA (2001).
13. Bryant, B., Miikkulainen, R.: Neuro-evolution for Adaptive Teams. In *Proceedings of the 2003 Congress on Evolutionary Computation*, (pp. 2194-2201). IEEE Press: Canberra (2003).
14. Eiben, A. E. & Smith, J. E.: Introduction to Evolutionary Computing, Springer-Verlag: Berlin (2003).