

# Biologically Inspired Embodied Evolution of Survival

Stefan Elfving<sup>1,2</sup>

Eiji Uchibe<sup>2</sup>

Kenji Doya<sup>2</sup>

Henrik I. Christensen<sup>1</sup>

<sup>1</sup> Centre for Autonomous Systems, Numerical Analysis and Computer Science,  
Royal Institute of Technology (KTH), S-10044 Stockholm, Sweden

<sup>2</sup> Neural Computation Unit, Initial Research Project, Okinawa Institute of Science and Technology, JST  
12-22 Suzaki, Gushikawa, Okinawa 904-2234, Japan

**Abstract-** Embodied evolution is a methodology for evolutionary robotics that mimics the distributed, asynchronous and autonomous properties of biological evolution. The evaluation, selection and reproduction are carried out by and between the robots, without any need for human intervention. In this paper we propose a biologically inspired embodied evolution framework, which fully integrates self-preservation, recharging from external batteries in the environment, and self-reproduction, pair-wise exchange of genetic material, into a survival system. The individuals are, explicitly, evaluated for the performance of the battery capturing task, but also, implicitly, for the mating task by the fact that an individual that mates frequently has larger probability to spread its gene in the population. We have evaluated our method in simulation experiments and the simulation results show that the solutions obtained by our embodied evolution method were able to optimize the two survival tasks, battery capturing and mating, simultaneously. We have also performed preliminary experiments in hardware, with promising results.

## 1 Introduction

Evolutionary robotics (ER) [3] is a framework for automatic creation of control systems of autonomous robots, inspired by the Darwinian principle of selective reproduction of the fittest. In standard ER reproduction is not integrated with the other autonomous behaviors. The selection and execution of genetic operations are therefore performed in a centralized manner.

Watson *et al.* [6] introduced the embodied evolution (EE) methodology for ER. EE was inspired by a vision where a large number of robots freely interact with each other in a shared environment, and performing some task. The robots produce offsprings by mating, i.e. a physical exchange of genetic material, and, naturally, the probability for a robot to produce offsprings is regulated by the robot's performance of the task. In short, EE is a methodology for ER that mimics the distributed, asynchronous and autonomous properties of biological evolution. The evaluation, selection and reproduction are carried out by and between the robots, without any need for human intervention.

In this paper we propose a biological inspired EE framework that fully integrates mating and recharging from battery packs in the environment, into a survival system. To study biologically inspired evolution in a robotic setting, we believe it is necessary to use a robotic platform with the

ability for self-preservation, i.e. recharging from external batteries, and self-reproduction, i.e. pair-wise exchange of genetic material.

We very much share the vision of Watson *et al.*, but the following issues have to be considered to realize biologically inspired EE:

- Limitation of the number of individuals.
- Power supply to sustain the robots internal batteries for an extended amount of time.
- Method for exchange genetic material between robots.
- The purpose of the evolution.

In the original EE framework each physical robot equals one individual in the population. Although this may be the ideal case, it makes the methodology inapplicable for most evolutionary computation tasks, because of the large number of robots required for an appropriate population size. To overcome the limitation of the population size, we have used a subpopulation of virtual agents for each physical robot, utilization time sharing for evaluation of the virtual agents.

Battery power is generally considered as a limitation for ER, because the robots have to interrupt their activity for a considerable amount of time to recharge their batteries. In experiments using the Khepera robot this problem is often solved by using an electrified floor to provide continuous power to the robots. From our more biological point of view we don't consider the sustaining of the robots internal battery power as a problem, but instead it is natural constraint for biological survival. This means that the performance of an individual is determined by its ability to find and physically recharge from external energy sources, and if an individual's battery power becomes too low the individual dies.

In our method mating is an essential part of the survival system. The individuals have to find mating partners and physically exchange genetic material with the partners. At the end of the lifetime an individual produces one offspring by selection and reproduction among the genomes the individual has mated with during its life. If an individual has not been able to perform any successful matings, the individual is not allowed to produce any offsprings.

Normally in ER and EE experiments the evolution optimizes the weights of a neural network controller that selects the low-level motor actions of the robot. Given the fact that EE has severe time limitation and that the foraging behavior and, especially, the mating behavior, requiring cooperation between two agents, constitute relatively diffi-

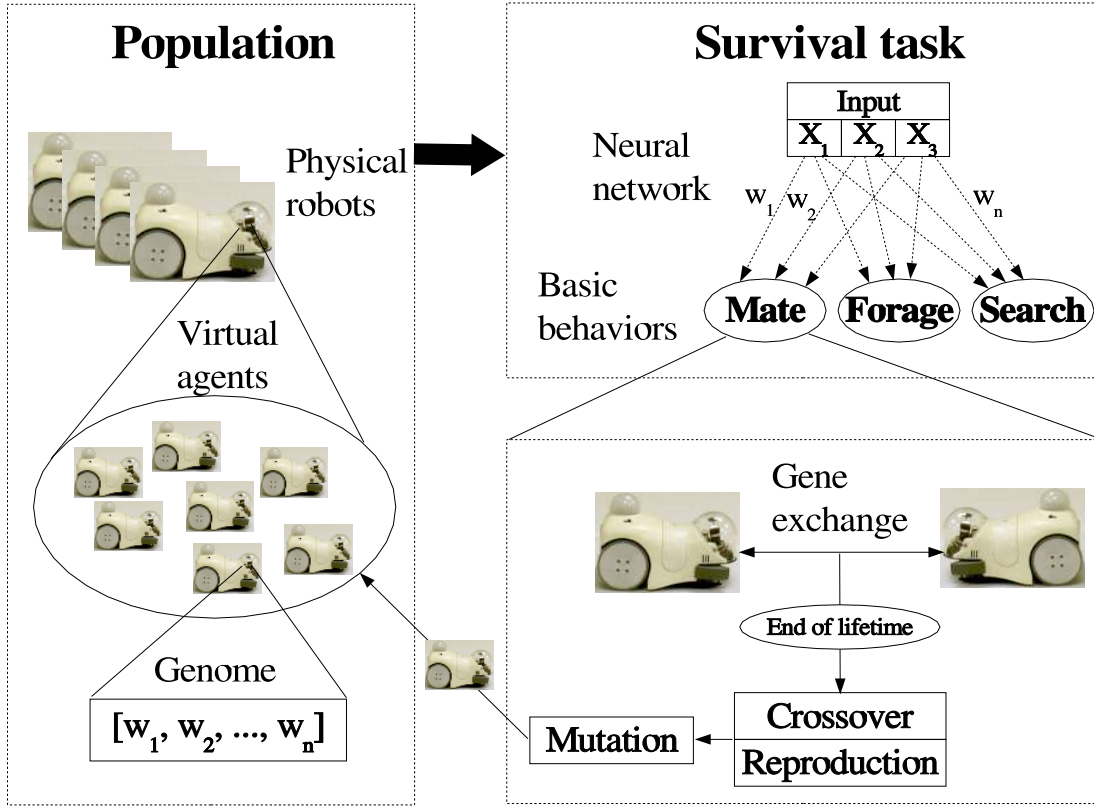


Figure 1: Overview of our embodied evolution scheme. Each physical robot contains a subpopulation of virtual agents. The virtual agents are evaluated for the survival task by time sharing. At a mating occasion a virtual agent saves the fitness value and genome of its mating partner, which are then used for selection reproduction of one offspring at the end of the virtual agent’s lifetime.

cult tasks, we find this approach unrealistic for our survival system. Instead we consider that the agents already have the knowledge to execute basic behaviors, such as mating, foraging and avoidance. The role of the evolution is therefore not to evolve the correct low-level motor action, but to select the appropriate basic behavior according to current situation and to optimize the recharging from the external battery packs.

There has been very few studies conducted in the field of EE apart from pioneer work by Watson *et al.* In which they used 8 Khepera robots to evolve a very simple neural network controller for a phototaxis task. The controller had two input nodes. One binary input, indicating which of the two light sensors receiving more light, and one bias node. The input nodes were fully-connected to two output motor neurons, controlling the speed of the wheels, giving totally four integer weights. In their experiments mating is not a directed behavior, instead the mating can be considered as a migration procedure. An agent broadcasts its genes according to a predefined scheme, and the other robots within communication range can then pick up the genes.

Usui *et al.* [5] used six Khepera robots to evolve an avoidance behavior. Their method is, though, more an “is-

land model” parallel genetic algorithm (GA) than an EE method. Each physical robot ran an independent GA for a subpopulation of virtual agents, where the virtual agents were evaluated by time sharing. Migrated genomes, broadcasted by other robots, were re-evaluated for the new robot.

Nehmzow [2] used EE to evolve three different sensory-motor behaviors, using two small mobile robots. In the experiments, the two robots first evaluated their current behavioral strategies, and after a fixed amount of time the robots initiated a robot seeking behavior. The robots then performed an exchange of genetic material via IR-communication, and genetic operations were applied according to fitness values. Each robot stored two strings: the currently active string and the best solution so far. If the GA did not produce an improved result, the best solution was used in the next generation.

## 2 Method

**Fig. 1** shows an overview of our proposed method for biologically inspired EE. Each physical robot has a subpopulation of virtual agents that are evaluated for the survival task by time sharing. The genome of each virtual agent consists

of an array of weights for the neural network controller. The neural network uses the the available sensory information to select the appropriate basic behavior in each time step. The most important part of the proposed method is that the selection of mating partners and reproduction of new individuals are integrated in the survival task. A virtual agent has to find suitable mating partners and perform a physical exchange of genetic material.

At the mating occasion a virtual agent saves the fitness value and genome of its mating partner. To keep the population size at a fixed level a virtual agent produce one offspring at the end of the lifetime. Among the genomes collected during the lifetime, the virtual agent selects one according the fitness values. The offspring is then created either by crossover, for which one of the two potential children is selected randomly, or reproduction of the fittest individual.

One general problem, especially for complex tasks, in evolutionary algorithms is that randomly created individuals often cannot complete the task at all, resulting in a fitness value of 0. In our setting this means that a virtual agent dies, i.e. energy becomes zero, or that the virtual agent is not able to perform any successful mating attempts during its lifetime. In out biologically inspired EE this means that the virtual agent is not able to transfer its genes to the next generation by producing an offspring. Note that individuals that die before the lifetime has expired has the ability to spread their genes by mating with individuals that leads a full life.

To minimize the need for creating random individuals, with low survival potential, the child that was not selected to be offspring, both from crossover and reproduction, is saved in a list at the physical robot. When a virtual agent dies or has not performed any matings, the new virtual agent is selected, randomly, from the list of genomes not used in earlier crossover and reproduction operations. The selected genome is then removed from the list. A random individual is only created if the list is empty, i.e. the early stage of the evolutionary process.

### 3 Experimental Setup

#### 3.1 Cyber Rodent Robot

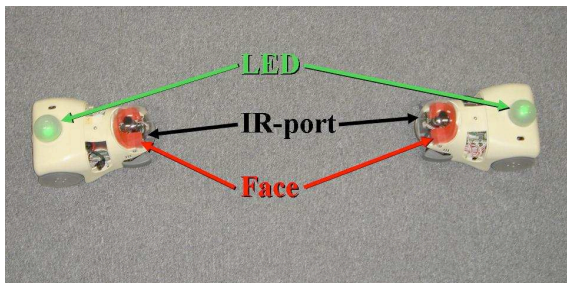


Figure 2: Two Cyber Rodent robots in mating position

This study has been performed within the Cyber Rodent (CR) project [1]. The main goal of the CR project is

to study the adaptive mechanisms of artificial agents under the same fundamental constraints as biological agents, namely self-preservation and self-reproduction. The CR, shown in **Fig. 2**, has two main features: the ability to exchange data and programs via IR-communications, for self-reproduction, and to capture and recharge from battery packs in the environment, for self-preservation.

The CR is a two-wheel mobile robot equipped with an omni-directional vision system, eight distance sensors, color LEDs for visual signaling, an audio speaker and microphones. Currently, the project has four CRs.

#### 3.2 Environmental setting

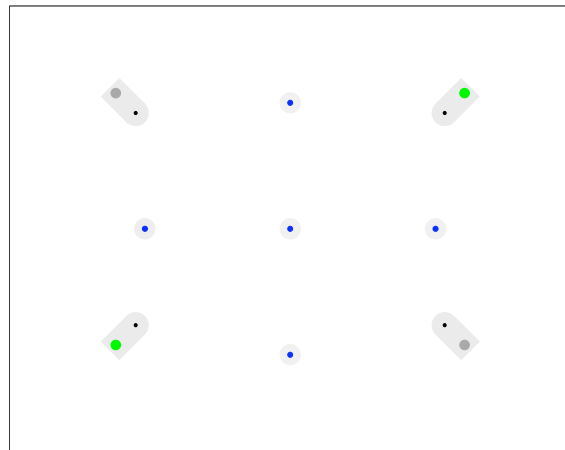


Figure 3: The simulation environment used for our experiments, with 4 simulated robots and 5 batteries.

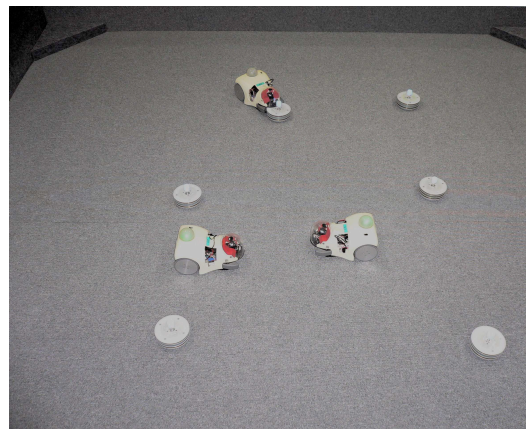


Figure 4: The hardware environment used for our experiments, with 3 CRs and 6 batteries

**Fig. 3** and **Fig. 4** show the simulation and hardware environments, respectively, that we have used in our experiments. The field in both environments was approximately 2.3m x 2.3m.

#### 3.3 Survival Task and Fitness Function

The task considered in this study considers the two basic biological tasks for survival: self-preservation by capture

and recharge from the battery packs in the environment, and self-reproduction by transfer of gene material via IR-communication.

We have used a simple virtual internal battery, to represent the energy level of the virtual agents. At birth a virtual agent is assigned an initial energy level, which is decreased by one in each time step. For each time step the agent is recharging from a battery pack the energy level is increased with a fixed value, up to maximum limit. To prevent the agents from continuing to recharge from the same battery, we have set a maximum number of time steps a battery can be recharged from and, also, after recharging the agent executing a random rotating motion. If the energy level becomes zeros the agent dies and a new virtual agent is created as described in section 2.

To ensure that a virtual agent encounters a variety of opponents during its lifetime, we have used a time sharing scheme where the virtual agents are switched after a fixed number time step, considerably smaller than the lifetime, or after the agent performed a successful mating.

We have used the following function for computing the fitness, when an agent executes a successful mating:

$$Fitness = no. of captured batteries / (time / 100),$$

i.e. the number of batteries an agent has captured up to the mating occasion, scaled with time the agent has lived. The fitness function only promotes the foraging part of the task explicitly, but both mating and optimization of the recharge time are implicitly promoted. An optimization of the recharge time prevents the agents from dying, i.e. the energy of the virtual battery becomes zeros, and also maximizes the available time for mating and foraging. Mating is promoted by the fact that an agent that mates frequently has larger probability to spread its gene in the population. In contrast an agent that does not perform any matings has zero probability to produce offsprings.

Parameter	Value
Lifetime	400 time steps
Time sharing	100 time steps
Max. energy	200 units
Initial energy	100 units
Recharge energy	5 units/time steps
Max recharge time	40 time steps
Total Population size	100
No. of physical robots	4
Virtual subpopulation size	25

Table 1: Parameters used for the survival task

### 3.4 Basic Behaviors

Our long time goal is to study the combination of learning and evolution. We have therefore used reinforcement learning (RL) [4] for training of the basic behaviors. The general

goal of RL is to learn a policy,  $\pi$ , that maximizes the cumulative future discounted reward. The value of a state  $s$ , the state value function, under policy  $\pi$  is given by

$$V^\pi(s) = E_\pi \left[ \sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t = s \right],$$

where  $E_\pi$  is the expected value given that the agent follows policy  $\pi$ .  $\gamma$ ,  $0 \leq \gamma \leq 1$ , is the discount parameter for future rewards, and  $r_t$  is the scalar reward for taking  $a_t$  in state  $s_t$ . Similarly, the value for taking action  $a$  in state  $s$ , the action value function, is given by

$$Q^\pi(s, a) = E_\pi \left[ \sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t = s, a_t = a \right],$$

In this study we have used Sarsa( $\lambda$ ) with tile coding function approximation and replacing traces (for algorithm details see e.g. [4]) to learn the basic behaviors. Sarsa is *on-policy* RL algorithm, which learns an estimate of the action value function,  $Q^\pi$ , while the agent follows policy  $\pi$ .

The basic behaviors were trained in the same environmental setting used for the evolutionary experiments (see section 3.2). After the learning was completed the action values were saved to be used for the embodied evolution. The three basic behaviors in our study were:

- **Mate** is used for moving the robot to an appropriate mating position and exchanging genetic materials via IR-communication. The IR-port is located in the front of the CR, slightly to the right of the center, directed straight forward. It is therefore necessary that the CRs face each other in a relatively small angle range for successful IR-communication. The **Mate** behavior uses the angle and distance to the LED (green color) or the face (red color) as sensory input, and is therefore only available if the CR has visual contact with another CR.
- **Forage** is used for approaching and capturing of battery packs (blue color). The **Forage** behavior uses the angle and distance to the closest battery as sensory input, and is therefore only available if the CR has visual contact with a battery pack
- **Search** is an obstacle avoidance behavior that uses the two largest readings from five front and one back proximity distance sensors. The **Search** behavior is always executable.

All sensory inputs are mapped to the normalized linear interval  $[0, 1]$ .

### 3.5 Neural Network controller

The one-layered neural network controller used in the experiments are shown in **Fig. 5**. The controller contains two parts for

- Optimizing the recharge time when CR has captured a battery (left part of the figure). This part of the neural network uses only three types of input: the current energy level, the number of time steps the CR has

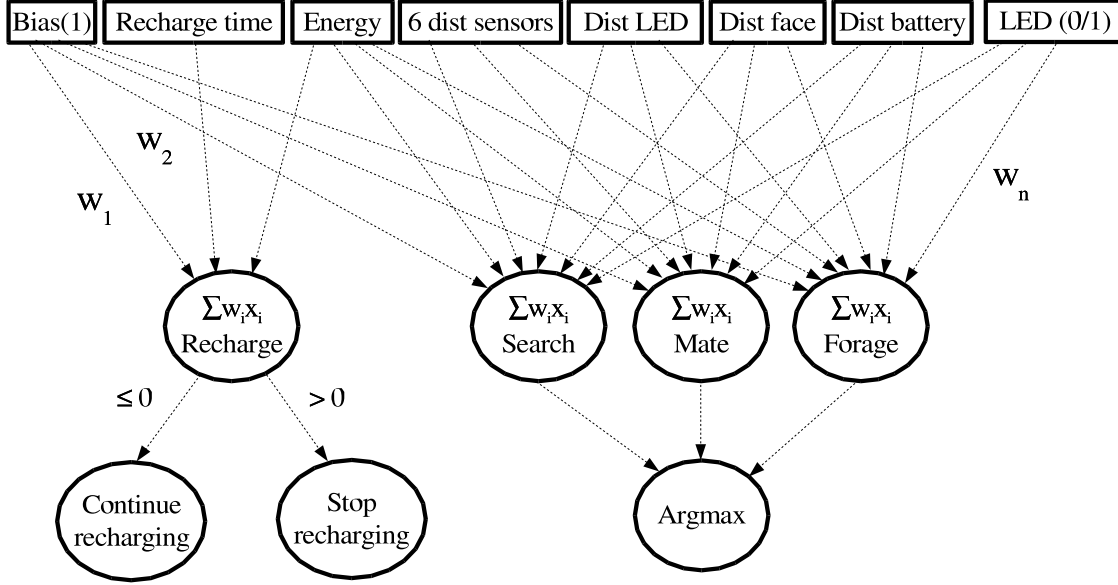


Figure 5: The neural network controller for selection of appropriate basic behaviors (right part), and controlling the recharge time (left part).

currently being recharging and the input from a bias unit. If the output, the weighted sum of the inputs, is less or equal to zero the CR continues recharging, otherwise the CR stops recharging.

- Selecting the appropriate basic behavior in each time step (right part of the figure), when the CR is not recharging. The input to the this part of the neural network consists of the current energy level, the six proximity distance sensors, the distances to the closest face, LED and battery, the bias unit and whether the CR had the LED turned on in the previous time step. In each time step the available basic behavior represented by output node with the largest activation is selected. In addition to selecting the basic behaviors the right part of the network also controls the LED used for visual signaling to the other CRs. If the activation for the **Mate** output node is larger than the activation for the **Forage** output node, the LED is turned on (green color), otherwise the LED is turned off.

In same manner as for the basic behaviors, each type of sensory input is mapped to the normalized linear interval  $[0, 1]$ , except for the discrete recursive input information about the LED status in the previous step.

The genome consists of the 39 neural network weights (3 for optimizing the recharge time and 36 for selecting the appropriate basic behavior), coded as real values. The initial weight values are uniformly distributed random values. When producing an offspring standard 1-point crossover is applied with a fixed probability. After crossover or reproduction each weight is mutated with a fixed probability by adding an uniformly distributed random number within the mutation range. In the experiments, we used a form of tournament selection, meaning that genome of the fittest mating

partner, together with the individual's own genome, is selected for reproduction.

Parameter	Value
Initial weight range	$[-1, 1]$
Crossover probability	0.6
Mutation probability	0.1
Mutation range	$[-0.5, 0.5]$

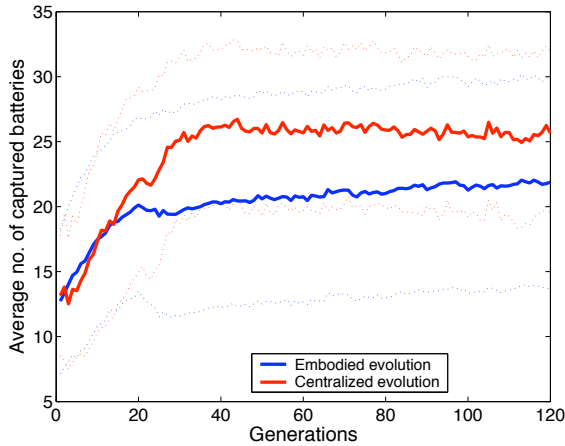
Table 2: Parameters used for evolving the neural network controller

## 4 Experimental Results

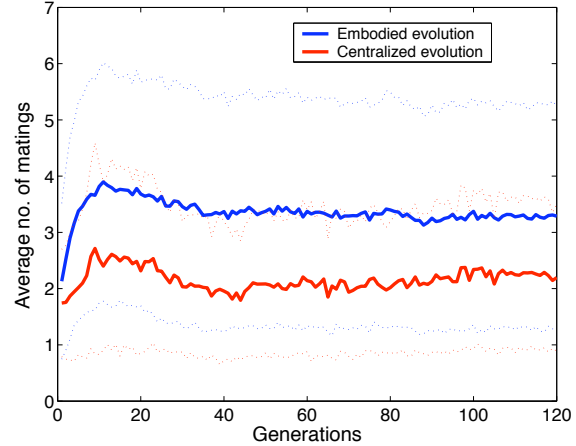
### 4.1 Simulation Experiments

To evaluate our method we have compared the results from the EE with a standard GA with centralized selection and reproduction (hereafter CE). For the CE we have used tournament selection with tournament size of 2. Except for centralization of the selection and reproduction, all settings were identical to EE, as described in section 3.2. The reproduction in the EE is asynchronous and therefore, one generation is not well-defined for the population. In the presented results for the EE, a generation for a virtual agent is considered to be complete if the virtual agent stays alive for the full lifetime. Because of the differences in time-scale for reproduction the results for CE and EE are not direct comparable, but they illustrate well the differences between the two evolutionary processes.

**Fig. 6** shows the results from the simulation experiments. The figures show the average number of captured batteries (**Fig. 6(a)**) and average number of matings (**Fig. 6(b)**) of 20



(a) Average number of captured batteries per generation.



(b) Average number of mating per generations.

Figure 6: Experimental results in the simulation environment for our proposed EE method (blue) and centralized evolution (red). The figures show the average results of 20 simulation experiments for the virtual agents that complete the survival task, i.e. lead a full lifetime and perform at least one mating. The thick solid lines show the average values and the thin dotted lines show the standard deviation

simulation experiments for the virtual agents that complete the survival task, i.e. lead a full lifetime and performed at least one mating.

For CE (red) the evolution converged after about 40 generations, with on average approximately 25 captured batteries and 2 matings per generation. For EE (blue) there was a significant increase in number of captured batteries from approximately 13 to 20 batteries in first 20 generations. For the rest of evolutionary process there was a small, but stable fitness increase, resulting in about 22 captured batteries after 120 generations. During the first 10 generations the average number of mating increased from 2 to 4, which then slowly decreased to relatively stable level of 3.5 after 40 generations.

For both cases the variance remained large through the evolution, which is explained by the fact that the performance of a virtual agent depends on random factors, such as the behaviors of the other active virtual agents, and the position of the CRs and the batteries at the start of each time sharing.

It is reasonable for CE to capture more batteries than EE because it was explicitly promoted in the fitness function. In addition the selection involves the whole population, not only the individuals that a virtual agent mates with as for EE. However, the goal of our survival task was not only to promote battery capturing, but also to promote mating. From this point of view the results for our EE method is very promising. Because the individuals obtained by EE performed significantly more matings, i.e. approximately 3.5 on average compared with 2 matings for CE.

The reason why EE could promote mating behaviors is that an individual that mates frequently can spread its genes to more individuals, and also receives more genes for selection. In contrast, for CE, only one mating is required for

maximum spread of an individual's genes, and an individual that spends less time trying to find mating partners has more time for battery capturing.

## 4.2 Preliminary Hardware Experiments

To evaluate our proposed EE method in the real hardware setting, we used individuals that were evolved for 40 generations in one of the simulation experiments. The individuals were then transferred to the real CRs and evolved for approximately 10 additional generations. The virtual population size in the hardware was set to 5, using the 5 fittest individuals in generation 40 from the simulated robots. Due to hardware failure we could only use 3 out of 4 available CRs.

For the basic behaviors, we used exactly the same learned  $Q$ -values as for the simulation experiments. The two main differences between the hardware and simulation environments are that (1) the sensor information, both from the vision system and the distance sensors, has much more uncertainty in the hardware setting, and (2) the basic behaviors functions well in the hardware setting, but use considerable longer time to perform the tasks. This is mainly caused by the larger uncertainty in the sensor values, but also by that the behaviors are not optimized for the individual hardware robots.

The differences between the simulator and hardware make the survival task considerably more difficult in the hardware environment, resulting in less captured batteries, as seen in **Fig. 7**. The figure shows the average number of captured batteries for the 5 virtual agents in each of the three CRs in the hardware setting. Even though the fitness values are small compared with the simulation experiments, the result are promising. For all three robots the fitness values are significantly increased over the short evolution



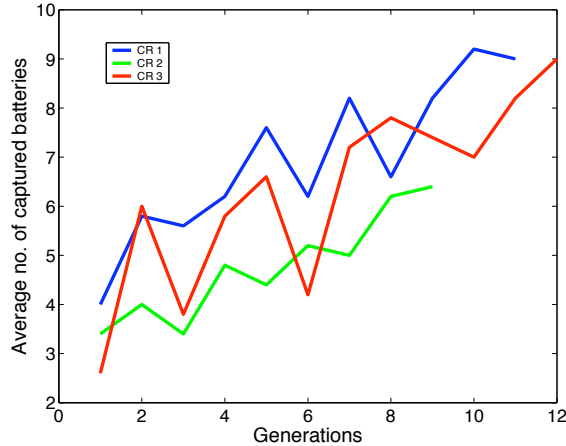


Figure 7: Average number of captured batteries for the 5 virtual agents in each of the three CRs in the hardware setting. The average was computed for all virtual agents that lead the full lifetime, where virtual agents that did not performed any matings received a fitness value, i.e. number of captured batteries, of 0.

time. The weaker performance of the virtual agents for CR 2 compared to the other two robots is probably explained by individual hardware differences between the robots. This suggests that an important issue for future hardware experiments is to optimize the basic behaviors for each robot individually.

## 5 Conclusions

This paper has proposed an EE method that integrates foraging, i.e. capturing and recharging from external batteries, and mating, i.e. pair-wise exchange of genetic material between robots, into a survival system. In the proposed method each physical robot has subpopulation of virtual agents that are evaluated for the survival task by time sharing. At a mating occasion a virtual agent saves the fitness value and genome of its mating partner, which are then used for selection and reproduction to produce one offspring at the end of the virtual agent's lifetime.

The used fitness function only, explicitly, promotes the capturing of batteries, but mating is, implicitly, promoted by the fact that an individual that mates frequently has larger probability to spread its gene in the population.

The results from our simulation experiments show that the individuals obtained by our EE method is able to optimize the performance of both the mating and battery capturing task, simultaneously. We have also performed preliminary hardware experiments with promising results.

## Acknowledgments

This research was conducted as part of "Research on Human Communication"; with funding from the Telecommunications Advancement Organization of Japan

Stefan Elfving's part of this research has also been sponsored by a shared grant from the Swedish Foundation for In-

ternationalization of Research and Education (STINT) and the Swedish Foundation for Strategic Research (SSF). The funding is gratefully acknowledged.

## Bibliography

- [1] Doya, K. and Uchibe, E. (2005) The Cyber Rodent Project: Exploration of Adaptive Mechanisms for Self-Preservation and Self-Reproduction. *Adaptive Behavior* (in press).
- [2] Nehmzow U. (2002) Physically Embedded Genetic Algorithm Learning in Multi-Robot Scenarios: The PEGA algorithm. In *Proc. of the Second International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*.
- [3] Nolfi S and Floreano D. (2000) *Evolutionary Robotics*. MIT Press.
- [4] Sutton R. S. and Barto A. G. (1998) *Reinforcement Learning: An Introduction*. MIT Press/Bradford Books.
- [5] Usui Y., and Arita T. (2003) Situated and Embodied Evolution in Collective Evolutionary Robotics. In *Proc. of the 8th International Symposium on Artificial Life and Robotics*, 212–215.
- [6] Watson R. A., Ficici S. G. and Pollack J. B. (2002) Embodied Evolution: Distributing an evolutionary algorithm in a population of robots. *Robotics and Autonomous Systems*, 39:1 18.