A MARKOV-CHAIN-BASED MODEL FOR SUCCESS PREDICTION OF EVOLUTION IN COMPLEX ENVIRONMENTS

Lukas König, Sanaz Mostaghim and Hartmut Schmeck
Institute AIFB, Karlsruhe Institute of Technology, Kaiserstr. 89, 76128 Karlsruhe, Germany

Keywords: Evolutionary robotics, Swarm robotics, Success prediction, Mathematical model, Markov chain, Selection.

Abstract: In this paper, a theoretical and experimental study of the influence of environments on the selection process in evolutionary swarm robotics is conducted. The theoretical selection model is based on Markov chains. It is proposed to predict the success rate of evolutionary runs which are based on a selection mechanism depending on implicit environmental properties as well as an explicit fitness function. In the experiments, the interaction of explicit and implicit selection is studied and a comparison with the model prediction is performed. The results indicate that the model prediction is accurate for the studied cases.

1 INTRODUCTION

Evolutionary Robotics (ER) is a methodology for the automatic creation of robotic controllers. Similarly to classic Evolutionary Computation (EC) methods, in ER individuals from a population of robot controllers are selected for mating (optionally using recombination) and are randomly mutated to achieve some desired behavioral property of a single robot or a collection of robots (*swarm*). For instance, a single robot can be trained to avoid obstacles or a swarm of robots can be trained to collectively transport a heavy object (Gross and Dorigo, 2009). While the field of ER goes beyond the evolution of robot controllers and captures, e. g., approaches to evolve real robot hardware, in this paper, we focus on the evolution of controllers for a swarm of mobile robots, i. e., the field of *Evolutionary Swarm Robotics (ESR)* or, more generally, on the evolution of agent behaviors in complex environments.

A key problem in ESR is to accurately select evolved controllers for producing offspring with respect to performing a desired behavior (Nolfi and Floreano, 2001). This means that "better" controllers in terms of the desired behavioral qualities should have a higher chance of being selected than "worse" ones. However, it is usually not possible to grade arbitrary evolved controllers detached from the environment in which the desired task has to be accomplished. Therefore, ESR scenarios typically require the existence of an environment (real or abstracted) where the controllers can be tested in. Using an environment to establish the quality of controllers makes ESR more closely related to natural evolution than most classic EC approaches. Here, the selection process can be seen from the classic EC or the biological point of view.

The Classic EC Point of View. In classic EC, selection is usually performed by considering the fitness (i. e., a numeric value that reflects the relative quality of an individual) of all individuals of a population and favoring the better ones. In ESR, fitness is computed by observing an individual's performance in an environment. This environmental fitness calculation can be noisy, fuzzy or time-delayed. Additionally, the environment is often responsible for an implicit preselection of individuals. In such cases the individuals have to match some environmental requirement (e. g., spatial proximity) to be even considered for fitness-based selection. For instance, in a decentralized scenario where robots perform reproduction when they meet, evolution selects implicitly for the ability to find other robots – in addition to the explicit selection based on a given fitness function (Bredeche and Montanier, 2010). Nolfi and Floreano refer to these two different factors influencing selection as *explicit* vs. *implicit fitness* (Nolfi and Floreano, 2001). We will stick to this denomination in the following; for abbreviation we will also write "fitness" when meaning explicit fitness. Thus, the environment adds fuzziness, noise and time-dependencies to the calculation of fit-

König L., Mostaghim S. and Schmeck H..
A MARKOV-CHAIN-BASED MODEL FOR SUCCESS PREDICTION OF EVOLUTION IN COMPLEX ENVIRONMENTS.
DOI: 10.5220/0003677700900102
In *Proceedings of the International Conference on Evolutionary Computation Theory and Applications* (ECTA-2011), pages 90-102
ISBN: 978-989-8425-83-6

ness and it introduces a complex implicit fitness to the selection process. Furthermore, by performing adaptive parameter control (Eiben et al., 2000), which is a promising approach for reducing parameter complexity in ESR, even more complicated fitness functions can arise. Finally, an erroneous design of the fitness function can also corrupt the fitness measure.

The Biological Point of View. In evolutionary biology, the *reproductive fitness* of an individual is calculated from the ability of the individual to both survive and reproduce with the consequence of contributing to the gene pool of future generations. There are several competing definitions on how to exactly calculate reproductive fitness in nature (e. g., short-term vs. long-term calculations), cf. (Sober, 2001), which, however, are beyond the scope of this paper. Reproductive fitness in nature as well as explicit and implicit fitness in ESR reflect the ability of an individual to be selected to produce offspring. However, while reproductive fitness in natural evolution is an observable but (mostly) unchangeable property, implicit and explicit fitness in ESR can be designed to guide the evolution in a certain direction. There, explicit fitness is rather straight-forward to design as it captures properties that can be encoded and calculated as numbers. For example, when evolving collision avoidance, driving can give positive fitness points while being close to a wall or producing a collision can be graded negatively. Implicit fitness, on the other hand, is more complex and difficult to influence as the entire environment has to be designed accordingly. Implicit fitness can also include complex long-term properties. E. g., A robot can promote its own offspring by helping its descendants to produce new offspring. As the offspring contains partially the same genes as the robot itself, the robot's implicit fitness increases due to the higher chance of contributing to the gene pool although its own reproduction rate is not improved.

As opposed to nature, in ESR we want to direct evolution in a certain direction. Therefore, we can design the explicit fitness function and, to some extent, implicit environmental selection properties according to desired behavioral criteria. For instance, we can look at a swarm that is explicitly selected for the ability to find a shortest path from a nest to some forage place. If the shortest path is too narrow to fit all the individuals passing it at a time, evolution might implicitly select for individuals that use a longer path or those who can decide to take a path based on congestion rates. Depending on the exact properties of the different paths, implicit selection might completely overrule explicit selection in this example. Overall, it turns out that in complex environments explicit fitness can play a subordinated role while implicit fitness has the major impact on selection. On the other hand, explicit fitness is easier to define in a proper way to drive evolution in a desired direction. Therefore, both implicit and explicit selection have to be used to induce a successful ESR run. In (Bredeche and Montanier, 2010), the impact of the environment has been experimentally investigated on a similar scenario using only implicit selection. There, robots evolved to explore the environment as they were selected for mating when they came spatially close to each other. In a second experiment, robots learned foraging being implicitly forced to collect energy or to die otherwise.

There are many approaches from the field of classic EC (namely EA, ES, GA, GP, EP, spatially distributed EA, etc.) to model the selection processes in a population of an evolutionary run, e. g., (Prügel-Bennett and Rogers, 2001), (Arnold, 2001), (Pietro et al., 2004), etc.; furthermore, there are models of natural processes from the field of evolutionary biology, e. g., (Kessler et al., 1997), and general concepts like genetic drift (Kimura, 1985) and schema theory (Holland, 1975). However, these models do not apply well to ESR scenarios due to the above mentioned differences concerning explicit and implicit selection.

The goal in this paper is to theoretically and experimentally study the influence of environment on the evolution. We present a model based on Markov chains that can be used to predict the success of an ESR run depending on implicit selection properties and the *selection confidence* of a system, i. e., a measure for the probability of selecting the "better" out of two different robots in terms of the desired behavioral properties. We use a mating procedure that is based on the idea of tournament selection meaning that k robots are selected (implicitly) by the environment, one of which is selected (explicitly) to overwrite the controllers of the $k - 1$ other ones by its own. In biological terms, this can be described as sexual reproduction without recombination with k parents and no genders, i. e., every individual can mate with all others. Both selection confidence and implicit selection probabilities of the environment are parameters to the model. We focus on the selection process without directly modeling a controller mutation, i. e., we look at the process "between mutations". The model can be used to estimate the success probabilities of superior mutations over inferior ones in an evolutionary run before it is actually performed in a real environment.

2 PRELIMINARIES

In this section we first describe the algorithm for the evolutionary model that is the basis for the presented theoretical framework. Then we make some preliminary assumptions and define Markov chains as used throughout this paper.

The Evolutionary Model. We assume an environment E including a population of n robots. We leave the terms "environment" and "robot" (or, more generally, "agent") loosely defined as we want to capture as many as possible of the numerous and partially conflicting definitions in the literature. For a discussion of the definitions of environments in multi-agent systems, cf. (Weyns et al., 2005). In this paper, an environment is thought of as a system that is at every point in time in some state out of a specific state space. Robots (or agents) are entities that can get information about the current environment state over a defined set of sensors and influence the succeeding state by a defined set of actions. Being in some sense part of the environment themselves, robots can also get information about their own internal state through sensors and change it through actuators. The process of deciding from sensory data which actions to perform is described by the controller of a robot.

Alg. 1 describes the evolutionary process as presumed for applying the proposed prediction model. The algorithm is stated from a population point of view, but it can also be applied in a decentralized way as in (Watson et al., 2002) (cf. Sec. 5). The function *Initialize* places n robots at random positions in the environment E and assigns them some arbitrary (empty or pre-defined) controllers from the vector $\vec{\Gamma}$. We assume that the population of robots is identified by the population of controllers $\vec{\Gamma} = (\gamma_1, \dots \gamma_n)$. The controllers may be of any common type like artificial neural networks or finite state machines; the latter ones are used in this paper. $Execute(\vec{\Gamma})$ runs the controllers for one step; this can occur sequentially or concurrently (as required for decentralized scenarios). The fitnesses of the controllers are stored in $\vec{F}$ and computed by an explicit fitness function f. Function *Mutate* indicates the mutation operator which is performed repeatedly at time intervals of *mutInterval*. Since mutation is not part of the prediction model, it is assumed that the model is applied between mutation operations (or at least between such mutations that actually change the behavioral quality of a robot). The mating process is a variant of the well-known tournament selection from classic evolutionary computation: Function *Match* selects a set of controllers T with size k. This selection can depend on environmental properties (e. g., spatial proximity). Among the selected controllers T, the controller *index* is fitness-proportionally chosen by the function $Select$ to overwrite the controllers in T.

Assumptions. In the following, we declare three assumptions that are valid throughout this paper. These assumptions should provide a simplified view on an ESR run, and still capture its essential properties.

Algorithm 1: Basic ESR run as required for the application of the prediction model.

```
input : Population of $n$ initial controllers
        $\vec{\Gamma} = (\gamma_1, \dots \gamma_n)$; environment $E$;
        tournament size $k$; maximal runtime
        $maxTime$; explicit fitness function $f$.
output: Evolved population.
$Initialize(\vec{\Gamma}, E)$
for $t \leftarrow 1$ to $maxTime$ do
    $Execute(\vec{\Gamma})$ // Run controllers
    $\vec{F} := f(\vec{\Gamma})$ // Compute explicit fitnesses
    if $t \bmod mutInterval == 0$ then
        $Mutate(\vec{\Gamma})$ // Mutation
    end
    //Mating
    $T := Match(k, \vec{\Gamma}, E)$
    if $T \neq \emptyset$ then
        $index := Select(T, F)$
        forall the $\Gamma(i) \in T$ do
            $\Gamma(i) = \Gamma(index)$
        end
    end
end
return $\vec{\Gamma}$
```

1. Mating is based on a variant of tournament selection in which the building of tournaments is not necessarily uniformly randomized (see Alg. 1). Tournament selection is a natural choice in ESR scenarios due to the existence of communication constraints (e. g., a limited communication distance or a limited number of communication channels). An obvious way to deal with this is to select small groups of robots that match the constraints and to let them reproduce.
2. At any time step, the population $\vec{\Gamma}$ can be divided into two subpopulations each of which contains only individuals of (nearly) equal quality in terms of the desired behavior. Without loss of generality, we say that $\mathcal{S}$ is a subpopulation with *superior* behavior in terms of the desired behavior than the *inferior* subpopulation $\mathcal{I} = \vec{\Gamma} \backslash \mathcal{S}$. We denote this relation of behavioral quality of $\vec{\Gamma}$ as $R(\vec{\Gamma}) = {}^{s}/_{i}$ with $s = |\mathcal{S}|$ and $i = |\Gamma| - s = |\mathcal{I}|$ being the number of individuals in population $\mathcal{S}$ and $\mathcal{I}$, respectively. Shorter, we write just $R(\vec{\Gamma}) = s = |\mathcal{S}|$ if $|\vec{\Gamma}|$ is known denoting only the superior individu-

als. I. e., , only one or maximally two basically different behaviors are existing at the same time; they can obviously be split into a better (or equal) subpopulation $\mathcal{S}$ and a worse (or equal) subpopulation $\mathcal{I}$. This assumption is less restrictive than it looks like, as it describes the most commonly observed situation during typical ESR runs. Furthermore, it reflects the biological situation where a major factor of evolution, especially in small populations, is thought to be genetic drift. This leads to large *neutral plateaus* of nearly equal fitness that are rather rarely affected by superior or inferior mutations (Kimura, 1985). From this point of view the capability of selecting superior individuals over inferior ones reflects the expected success of evolution.

3. On average, the reproduction capability of a robot depends only on its own controller and the controllers of the other agents and not on other factors from the environment. Particularly, a robot's long-term chance of being selected has to be independent of its current state in the environment. For instance, if robots reproduce when meeting each other, we assume that there is no obstacle in the environment which prohibits reproduction for parts of the population by separating them in a closed area. This assumption should hold in all "reasonable" scenarios where the robots are capable of improving their chance for reproduction by altering their controller. Of course, in a real-world scenario it can happen, e. g., that a robot falls into a hole which it cannot leave whatever controller itself or the other robots have. However, this seems to be a situation which cannot be resolved by an improvement of the evolution process, but rather is a problem of appropriate hardware design for a given scenario. Therefore, we assume that such situations do not occur during an ESR run.

Markov Chains. Given a finite state space M a (first order) Markov chain is given by the probabilities to get from one state to another. There, the Markov property has to be fulfilled, which requires that future states depend only on the current state, but not on the past states. We define a Markov chain in a common way according to (Grinstead and Snell, 1997).

Definition 1. *Markov Chain.*

A Markov chain is given by a finite state space $M = \{m_1, \ldots, m_{|M|}\}$ *and a matrix of probabilities* $p_{ij}, 1 \leq i, j \leq |M|$ *determining the probability to get in one step from* m_i *to* m_j.

3 COMPLETELY IMPLICIT SELECTION (CIS)

In this section, we investigate a Completely Implicit Selection process using a homogeneous Markov chain model. It works without any explicit fitness function, i. e., the function f returns the constant 1.

3.1 Mating Size 2

First we start with the mating size set to $k = 2$ (CIS-2). This is the most simple version of the model and a natural starting point.

An example for a scenario that is captured by the CIS-2 model is an environment where a robot has to come close to another robot for mating. Mating is performed by randomly selecting one robot that copies its controller to the other one. Here, the environment implicitly selects for the ability of finding another robot – maybe in a labyrinth; however, selection pressure is quite low as a robot that waits for another one to come has during mating the same chances of passing on its controller as the other one that actively explored the environment. Therefore, superior individuals are necessary for making mating possible but they are not *explicitly* selected for.

In the general CIS-2 scenario, two robots mate when they match some arbitrary mating criterion. The winner is then chosen by uniform distribution (as the fitness function returns always 1) which means that it is uniformly random which robot gives its controller to the other one. As each of the robots in a mating tournament may be from one of the sets $\mathcal{S}$ or $\mathcal{I}$, one of the four situations $\mathcal{II}$, $\mathcal{IS}$, $\mathcal{SI}$, $\mathcal{SS}$ can occur in a mating tournament. For the cases $\mathcal{II}$ and $\mathcal{SS}$, there is no change made to the population; for the cases $\mathcal{IS}$ and $\mathcal{SI}$ chances are 0.5 for both overwriting the controller from $\mathcal{S}$ with the controller from $\mathcal{I}$ and vice versa. Therefore, the population will either gain a new $\mathcal{S}$ robot and lose an $\mathcal{I}$ one or the other way around, cf. Fig. 1).

The mating process of a population with n individuals can be written as an $n+1 \times n+1$ transition matrix. The rows and columns correspond to the different possible states of a population and the entries denote the probabilities for a state transition. The $n + 1$ different possible states are denoted by $^0/_n$, $^1/_{n-1}, \ldots, {}^n/_0$ where $^i/_s$ means i superior and $s = n - i$ inferior robots in the population (we will also write simply i for this situation if n is known). An entry p_{ij} is the probability that a population that is currently in state i changes to state j after one mating event. This implies that every row of the matrix sums up to 1.

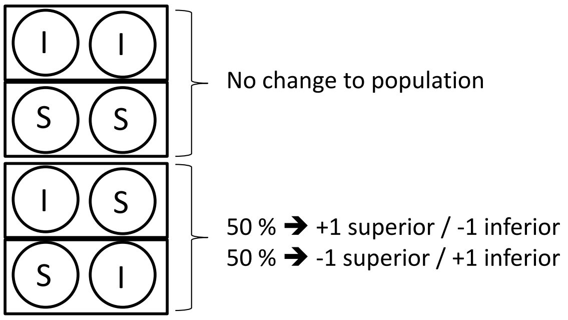


Figure 1: Without any explicit fitness, the winner of a tournament is drawn in a uniformly random way. For $k = 2$, there are equal chances for the population to gain or to lose a superior individual.

For the mating procedure described above, the matrix P_{CIS-2} is given by

$$P_{CIS-2} =$$

$$\begin{array}{c|ccccccc}
 & ^{0}/_{n} & \cdots & ^{i-1}/_{n-i+1} & ^{i}/_{n-i} & ^{i+1}/_{n-i-1} & \cdots & ^{n}/_{0} \\
\hline
^{0}/_{n} & 1 & & & & & & \\
\vdots & & \ddots & & & & 0 & \\
^{i}/_{n-i} & & & \frac{c_i}{2} & s_i & \frac{c_i}{2} & & \\
\vdots & & 0 & & & & \ddots & \\
^{n}/_{0} & & & & & & & 1
\end{array}$$

with

$$\forall i \in \{1, \ldots, n-1\} : \; c_i, s_i \in [0, 1], \; c_i + s_i = 1.$$

There, the c_i, s_i are the probabilities that in population state $^{i}/_{n-i}$ a mating induces a state change (c_i, i. e., two different robots mate) or the population stays in the same state (s_i, i. e., two uniform robots mate). In the states $^{0}/_{n}$ and $^{n}/_{0}$ there are no different individuals in the population, therefore, no state change can be induced by mating. These states cannot be left once one of them is entered and the population remains stable henceforth. In a transition matrix such a state is always indicated by a 1 at a diagonal entry.

3.2 Eventual Stable States ($k = 2$)

We are now interested in the long-term development of a population, namely in the question if the population will eventually enter the stable state $^{n}/_{0}$. This is the desired case where all individuals received a superior controller. On the other hand, if the population enters the state $^{0}/_{n}$ this means that there are only inferior individuals in the population left and the selection mechanism was not capable of preserving the superior controller. Therefore, the probability for eventually entering the stable state $^{n}/_{0}$ is an indicator for the quality of the chosen selection mechanism.

The transition matrix P_{CIS-2} from above defines a homogeneous Markov chain with the states $S = \{^{0}/_{n}, ^{1}/_{n-1}, \ldots, ^{n}/_{0}\}$. Note that two states of the chain are absorbing in the sense that there is no way to leave them. The set of absorbing states is $A = \{^{n}/_{0}, ^{0}/_{n}\} \subset S$. As an absorbing state can be reached from every state $s \in S \backslash A$, the matrix P is called *absorbing* and the non-absorbing states $T = S \backslash A$ are called *transient*.

If the matrix is raised to the power of n, an entry p_{ij} of the resulting matrix displays the probability that state j is reached after n steps if the population started in state i. As we are interested in the eventual stable state of the system we want to calculate the limit

$$P^{\infty}_{CIS-2} = \lim_{n \to \infty} P^{n}_{CIS-2}.$$

As shown in (Grinstead and Snell, 1997) the matrix P^{∞}_{CIS-2} exists meaning every entry $p^{(\infty)}_{ij}$ converges. The limit can generally be calculated for every absorbing Markov chain (thus, particularly for all chains in this paper). The limit matrix has non-zero entries in the columns which denote the absorbing states and zero entries at all other positions (as there is a non-zero chance for every transient state to reach an absorbing state). Therefore, it is only necessary to calculate the absorbing columns of the limit matrix.

For any absorbing Markov matrix P the non-zero columns of P^{∞} can be calculated by the following procedure. First, the *canonical form* CF_P of the matrix P is generated by shifting all absorbing states to the end in rows and columns such that an identity submatrix is built at the right bottom corner of P. For the matrix P_{CIS-2} the $^{0}/_{n}$-state is already at the correct position; the $^{n}/_{0}$-state has to be shifted to the next to last position (in rows and columns):

$$CF_{P_{CIS-2}} := \left(\begin{array}{ccccc|cc}
s_1 & c_1 & \cdots & & & c_1 & 0 \\
 & \ddots & & & & 0 & 0 \\
\cdots & c_j & s_j & c_j & \cdots & \vdots & \vdots \\
 & & & \ddots & & 0 & 0 \\
 & & \cdots & c_{n-1} & s_{n-1} & 0 & c_{n-1} \\
\hline
0 & & \cdots & & 0 & 1 & 0 \\
0 & & \cdots & & 0 & 0 & 1
\end{array} \right)$$

The new matrix has now generally the form

$$CF_P = \left(\begin{array}{c|c} Q & R \\ \hline 0 & I \end{array} \right)$$

where Q consists of transitions between transient states, R consists of transitions from transient states to absorbing states, I is an identity matrix reflecting transitions within absorbing states and 0 is a zero matrix. The matrix N_P with

$$N_P = (I - Q)^{-1}$$

(where I is an identity matrix with the same size as Q) is called the *fundamental matrix*. Now, in the matrix

$$L_P = N_P \cdot R$$

an entry l_{ij} is the probability that the absorbing chain will be absorbed in the absorbing state j if the process starts in state i. Therefore, L contains exactly the non-zero columns of the desired limit matrix P^{∞}.

For the matrix P_{CIS-2} the limit calculates to

$$L_{P_{CIS-2}} = \begin{array}{c|cc} & {}^{0}/_{n} & {}^{n}/_{0} \\ \hline {}^{0}/_{n} & 1 & 0 \\ {}^{1}/_{n-1} & 1-\frac{1}{n} & \frac{1}{n} \\ {}^{2}/_{n-2} & 1-\frac{2}{n} & \frac{2}{n} \\ \vdots & \vdots & \vdots \\ {}^{n-2}/_{2} & \frac{2}{n} & 1-\frac{2}{n} \\ {}^{n-1}/_{1} & \frac{1}{n} & 1-\frac{1}{n} \\ {}^{n}/_{0} & 0 & 1 \end{array}$$

independently of the c_i and s_i probabilities. Note that the probability for ending in the superior state increases linearly with the number of superior robots at the beginning. Simultaneously, the probability for ending in the inferior state decreases at the same range. This result is quite intuitive as in the CIS-2 scenario the winner within a tournament is drawn by uniform probabilities. The order in which they are put together to tournaments does not have an influence on the overall probabilities of reaching one of the stable states.

The fact that the probabilities for entering the superior state are symmetrical to entering the inferior state has as a consequence that the long-term success of CIS scenarios is compromised. During evolution, a population does not persist in a stable state, but is attacked by mutations that can cause a transition from one stable state (neutral plateau) to another. If the chances for an inferior mutation to overrule the population are the same as for a superior mutation, the population cannot constantly remain in an improvement process. Usually the initial population has a low behavioral quality, therefore, a CIS-2 scenario can lead to improvements in the beginning, but they cannot remain stable in the long-term.

3.3 Mating Size k

The above CIS-2 scenario can be stated in a more general form for a mating neighborhood of size k (CIS-k). At first, an explicit fitness function is still omitted.

The CIS-k scenario considers an k-sized tournament for mating. One of the k controllers is selected by a uniform probability to be copied to all other robots in the tournament during mating. Using the same notation as in Sec. 3.1, the CIS-k transition matrix P_{CIS-k} is given in Fig. 2.

As in the CIS-2 case an entry p_{ij} of the matrix P_{CIS-k} is the probability that a population in state i switches to state j by a mating event. By c_{ij} we denote the probability that in a population that is currently in state ${}^{i}/_{n-i}$ the next mating event is based on a ${}^{j}/_{k-j}$ tournament, i. e., a tournament with j superior and $k - j$ inferior individuals. The diagonal elements of the matrix (marked by a box in the figure) denote the probability that the population state does not change by a mating. Therefore, they are given by the sum of the probabilities for a tournament with only superior and a tournament with only inferior individuals, i. e., $p_{ii} = c_{i0} + c_{ik}$.

For calculating the left/bottom non-diagonal entries p_{ij} ($i > j$) we have to consider the probability $c_{i,i-j}$ that a ${}^{i-j}/_{k-i+j}$ tournament occurs in an ${}^{i}/_{n-i}$ population; such a tournament can turn a population from state ${}^{i}/_{n-i}$ to ${}^{j}/_{n-j}$. This probability has to be multiplied by the probability that an inferior individual will win the tournament (since $j < i$ means that the number of superior individuals decreases). As the individuals are drawn by uniform distribution from the tournament, this probability is depending only on the number of superior and inferior individuals. It is given by $\frac{k-(i-j)}{k}$. The overall probability that defines an entry $p_{ij}, i > j$ is given by

$$p_{ij} = \frac{k-(i-j)}{k} \cdot c_{i,i-j}$$

Analogously the right/top non-diagonal entries $p_{ij}, i > j$ can be computed by

$$p_{ij} = \frac{k-(j-i)}{k} \cdot c_{i,k-j+i}$$

It has to hold for all c_{ij} in the matrix P_{CIS-k}:

$$\forall i \in \{1, \ldots, n-1\} : \sum_{j=\max(0,i+k-n)}^{\min(k,i)} c_{ij} = 1 \ \left(c_{ij} \in [0,1]\right).$$

This implies that the sum of every row i of the matrix P_{CIS-k} is 1.

During one mating, at most $k - 1$ individuals can be turned from $\mathcal{S}$ to $\mathcal{I}$ or vice versa. That is reflected by the fact that all probabilities c_{ij} with $j < 0$ or $j > k$ are zero. Therefore, at most the $k - 1$ elements left and right of the diagonal elements in matrix P_{CIS-k} are non-zero. Furthermore, in populations with $i < k$ superior ($n - i < k$ inferior) individuals all probabilities c_{ij} with $j > i$ ($j > n - i$) have to be zero as at most i superior ($n - i$ inferior) individuals can be in a tournament. Therefore, the matrix C which is given

$$P_{CIS-k} =$$

$$\begin{array}{c|cccccccccccccc}
 & {}^{0}\!/_{n} & {}^{1}\!/_{n-1} & {}^{2}\!/_{n-2} & \cdots & {}^{k}\!/_{n-k} & {}^{k+1}\!/_{n-k-1} & \cdots & {}^{i-2}\!/_{n-i+2} & {}^{i-1}\!/_{n-i+1} & {}^{i}\!/_{n-i} & {}^{i+1}\!/_{n-i-1} & {}^{i+2}\!/_{n-i-2} & \cdots \\
\hline
{}^{0}\!/_{n} & \boxed{1} & 0 & 0 & & 0 & 0 & & 0 & 0 & 0 & 0 & 0 & \\
{}^{1}\!/_{n-1} & \frac{k-1}{k}c_{1,1} & \boxed{c_{1,0}+c_{1,k}} & 0 & \vdots & \frac{1}{k}c_{1,1} & 0 & \vdots & 0 & 0 & 0 & 0 & 0 & \cdots \\
{}^{2}\!/_{n-2} & \frac{k-2}{k}c_{2,2} & \frac{k-1}{k}c_{2,1} & \boxed{c_{2,0}+c_{2,k}} & & \frac{2}{k}c_{2,2} & \frac{1}{k}c_{2,1} & & 0 & 0 & 0 & 0 & 0 & \\
\vdots & & & & & & & \ddots & & & & & & \\
{}^{i}\!/_{n-i} & 0 & 0 & 0 & & 0 & 0 & \cdots & \frac{k-2}{k}c_{i,2} & \frac{k-1}{k}c_{i,1} & \boxed{c_{i,0}+c_{i,k}} & \frac{k-1}{k}c_{i,k-1} & \frac{k-2}{k}c_{i,k-2} & \cdots \\
\vdots & & & & \vdots & & & & & & & \ddots & &
\end{array}$$

Figure 2: General transition matrix for selection without an explicit fitness function in a population of size n, using mating tournaments of size k. Diagonal elements are marked by a surrounding box; they represent transitions where no state change occurs as only one agent type ("superior" or "inferior") is selected in a tournament. The matrix has at most $k-1$ non-zero elements at the left and the right side of the diagonal elements of every row. All rows sum up to 1.

by the probabilities c_{ij} for $0 \le i \le n, 0 \le j \le k$ has the form

$$C = \begin{array}{c|cccccc}
 & {}^{0}\!/_{k} & {}^{1}\!/_{k-1} & {}^{2}\!/_{k-2} & \cdots & {}^{k-1}\!/_{1} & {}^{k}\!/_{0} \\
\hline
{}^{0}\!/_{n} & 1 & 0 & 0 & \cdots & 0 & 0 \\
{}^{1}\!/_{n-1} & c_{1,0} & c_{1,1} & 0 & \cdots & 0 & 0 \\
\vdots & & \vdots & & & \vdots & \\
{}^{i}\!/_{n-i} & c_{i,0} & c_{i,1} & c_{i,2} & \cdots & c_{i,k-1} & c_{i,k} \\
\vdots & & \vdots & & & \vdots & \\
{}^{n-1}\!/_{1} & 0 & 0 & 0 & \cdots & c_{n-1,k-1} & c_{n-1,k} \\
{}^{n}\!/_{0} & 0 & 0 & 0 & \cdots & 0 & 1
\end{array}$$

There are $k \cdot (n-k) + n + 1$ non-zero entries in C. Every row has to sum up to 1 in this matrix as well.

3.4 Eventual Stable States (Arbitrary k)

By the same procedure as in Sec. 3.2 the probability for a population eventually reaching the stable states ${}^{0}\!/_{n}$ and ${}^{n}\!/_{0}$ when starting in some state ${}^{i}\!/_{n-i}$ can be computed. As the choice of an individual in a tournament is still uniform, it is not surprising that for all mating sizes k and all probability matrices C the probability distribution is the same as in the CIS-2 case: $L_{P_{CIS-k}} = L_{P_{CIS-2}}$.

However, the expected time to absorption, i. e., the number of mating events until a stable state is reached, decreases if k is increased. The expected time to absorption of a Markov chain given by matrix P is given by a vector t; a position t_i of the vector is the expected number of mating events until the chain is in an absorbing state if it starts in state i. The vector t can be computed by

$$t = N_P \cdot v$$

where N_P is the fundamental matrix of P (cf. Sec. 3.2) and v is a column vector all of whose entries are 1.

For example, the expected time to absorption is depicted in Fig. 3 for population size $n = 10$ and tournament sizes $k = 2, \ldots, 9$. Obviously, the time to absorption decreases drastically from mating size 2 to mating sizes 3 and 4. However, it is important to note that "time" is measured here in terms of the number of mating events. Depending on the environment it may take longer in terms of evolution time to select tournaments of bigger size than those of smaller size.

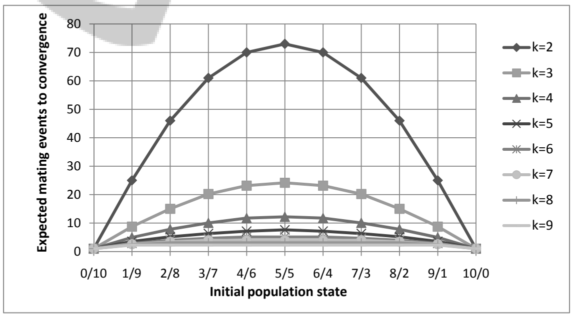


Figure 3: Expected time to absorption as a function of the initial population state (${}^{0}\!/_{10}, \ldots, {}^{10}\!/_{0}$) for tournament sizes $k = 2, \ldots, 9$ in a population with $n = 10$ individuals.

4 COMBINATION OF EXPLICIT AND IMPLICIT SELECTION (EIS)

In the CIS scenarios above, explicit fitness has not been considered in the model. In this section an extension is introduced to model explicit fitness. We introduce an explicit fitness to the model by making the probability for superior individuals to be winners in a tournament higher than that of inferior individuals. In the evolution process described by Alg. 1 explicit

T. M.	$^{i-k+1}/_{n-i+k-1}$	.	$^{i-2}/_{n-i+2}$	$^{i-1}/_{n-i+1}$	$^{i}/_{n-i}$	$^{i+1}/_{n-i-1}$	$^{i+2}/_{n-i-2}$	.	$^{i+k-1}/_{n-i-k+1}$
Tourn.	$^{k-1}/_{1}$	.	$^{2}/_{k-2}$	$^{1}/_{k-1}$	$^{k}/_{0}$ or $^{0}/_{k}$	$^{k-1}/_{1}$	$^{k-2}/_{2}$	.	$^{1}/_{k-1}$
$^{i}/_{n-i}$	$\frac{1-c}{k} c_{i,k-1}$	.	$\frac{(k-2)(1-c)}{k} c_{i,2}$	$\frac{(k-1)(1-c)}{k} c_{i,1}$	$c_{i,k} + c_{i,0}$	$\frac{k-(1-c)}{k} c_{i,k-1}$	$\frac{k-2(1-c)}{k} c_{i,k-2}$	.	$\frac{k-(k-1)(1-c)}{k} c_{i,1}$

Figure 4: Non-zero entries of an inner row $^{i}/_{n-i}$ of a general EIS transition matrix. The first heading denotes the column of the transition matrix, the second one the corresponding mating tournament, i. e., the column of the probability matrix C the $c_{i,j}$ values are taken from.

fitness is given by the function f. It is calculated from environmental variables and is intended to measure the desired behavioral qualities. Factors like noise in the environment, delayed fitness calculation and erroneous design of the fitness function can corrupt the fitness measure. Therefore, the probability that a superior individual is selected explicitly over an inferior individual is usually below 1.

To reflect the influence of f to selection, a *confidence factor* $c \in [0, 1]$ is introduced which states how accurately f differentiates between superior and inferior individuals. A low value for c means that the explicit fitness cannot increase the chance that a superior individual is chosen in a tournament. For a value $c = 0$ the EIS model is equivalent to the CIS model. A high value means that it is likely for a superior individual to be chosen; $c = 1$ means that in every tournament that contains at least one superior individual such an individual will win.

The confidence factor is included to the model as follows: at the left/bottom side of the diagonal of the transition matrix the entries are multiplied by $(1 - c)$. At the right/top side of the diagonal the enumerator $k-(j-i)$ is replaced by $k-(j-i)(1-c)$. By this means the chance for switching to a state right of the diagonal gets higher when c is increased (to maximally the $c_{i,i-j}$ value) while the chance for switching to a state left of the diagonal decreases (to minimally 0). The diagonal entries do not have to be changed as the corresponding tournaments consist of uniform individuals. It is obvious that each row of the matrix still sums up to 1.

Generally, for population size n, tournament size k and fitness confidence factor c the $n + 1 \times n + 1$ transition matrix P_{EIS-k} is given by

$$p_{ij} = \begin{cases} c_{i0} + c_{ik} & \text{if } i = j, \\ \frac{(k-(i-j))(1-c)}{k} c_{i,i-j} & \text{if } i - k < j < i, \\ \frac{k-(j-i)(1-c)}{k} c_{i,k-j+i} & \text{if } i < j < i + k, \\ 0 & \text{otherwise} \end{cases}$$

A complete inner row of the most general form of the transition matrix is given in Fig. 4. Note that the restrictions to the tournament probability values given in matrix C in Sec. 3.3 are valid as well here.

Influence of the Probability Matrix C. The probability matrix C (cf. Sec. 3.3) defines the probabilities for different mating tournaments to occur. It is influenced by the environment and the concrete selection strategy. For a real ESR scenario it has to be identified experimentally in preliminary tests or estimated. A first guess can be a uniform distribution within every row.

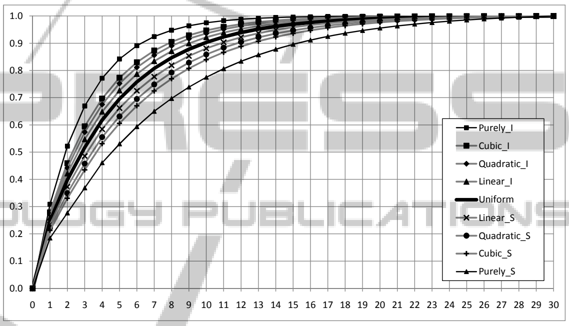


Figure 5: Probabilities for converging eventually to the superior state $^{n}/_{0}$ as a function of the initial population state (the numbers i labeling the X-axis denote the initial population state $^{i}/_{n-i}$); plotted for population size $n = 30$, tournament size $k = 4$, confidence factor $c = 0.2$, and 9 different tournament probability distributions C. The plot "Uniform" refers to a uniform distribution in C. The plots "Cubic_I", "Quadratic_I" and "Linear_I" refer to the distributions of C where the chances for inferior tournaments are higher, the plots "Cubic_S", "Quadratic_S" and "Linear_S" to those where the chances for superior tournaments are higher; the plots "Purely_I" and "Purely_S" refer to the extreme cases where only $^{1}/_{3}$ and $^{3}/_{1}$ tournaments are selected, respectively (cf. description in text).

In an EIS scenario, the probabilities in C can have an impact on the convergence probabilities of an evolutionary run. Fig. 5 shows the probabilities of a population for converging to the superior state (i. e., all individuals are superior) as a function of the initial population state (i. e., the number of superior individuals in the initial population) for different probability distributions C. The thick black plot in the middle corresponds to a uniform distribution in every row of C. For the 3 gray plots right of the middle, the probability for a tournament $^{i}/_{k-i}$ to occur is increased in a linear, quadratic or cubic manner with the number of superior individuals (i. e., the probability is set to $(i + 1)^e$ for $e \in \{1, 2, 3\}$ and then normalized such that every row of C sums up to 1; the population state is not taken into account). Symmetrically, For the 3 gray plots left of the middle, the probability is increased with the number of inferior individuals (i. e., for a tournament $^i/_{k-i}$ it is set to $(k - i + 1)^e$ and then normalized). The leftmost and rightmost plots labeled "Purely_I" and "Purely_S", respectively, belong to the extreme settings where selection is performed with a probability of 1.0 in $^1/_3$ and $^3/_1$ tournaments, all other entries of C being set to 0 (except for the impossible cases in the three upper or lower rows; here the column which is as near as possible to the $^1/_3$ or $^3/_1$ tournament, respectively, is set to 1.0). These two plots can be seen as the limits of the polynomial plots described above for $e \rightarrow \infty$, i. e., within this range all the polynomial plots lie, when e is allowed to be an arbitrary number.

Note that the plots that gain the highest chances for eventually converging to the superior state (left of the middle) are those that have the highest chance of selecting *inferior* tournaments. Accordingly, the plots with the lowest success rate are those that select mainly for *superior* tournaments. This observation is against the intuition that selection should always favor superior individuals over inferior ones. In the given scenario selecting tournaments with few superior individuals that, in consequence, have a relatively high chance of converting a lot of inferior individuals pays off more than selecting superior tournaments which can convert few inferior individuals at a time and include a risk that an inferior individual converts a lot of superior ones.

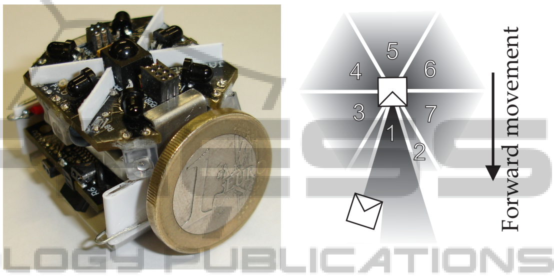
5 EXPERIMENTS

The presented model is firstly applied to two rather artificial ESR scenarios with a centralized selection operator. Secondly, a more realistic decentralized ESR scenario is studied. In the first part of this section the evolutionary setup is described. Afterwards, the experimental results are presented and discussed.

5.1 Evolutionary Setup

In the experiments, we utilize the evolutionary setup described in (König et al., 2009) based on finite state machines (FSM).

Robot Platform. The experiments have been performed on a simulated Jasmine-IIIp robot platform. The Jasmine IIIp series is a swarm of micro-robots sized $29 \times 29 \times 26$ mm^3 (cf. Fig. 6(a)). Every robot can process simple motoric commands like driving forward or backward or turning left or right. Every robot has seven infra-red sensors (as depicted in Fig. 6 (b)) returning values from 0 to 255 in order to measure distances to obstacles. The Jasmine-IIIp robot has more sensory capabilities which are described at www.swarmrobot.org. In this paper only the above described capabilities are used. In simulation a robot drives 4 mm per simulation step or turns 10 degrees to the left or right. When a robot collides with another robot or with a wall, a crash simulation is performed that positions the robot at a new random free place at a distance of at most 4 mm from the crash position and turns it by a random angle (if no such position exists, the robot remains at its last position before the crash).



(a) (b)

Figure 6: Jasmine-IIIp robot. (a) Photography of a Jasmine IIIp next to a 1-Euro coin. (b) Placement of infra-red sensors for distance measurement around a Jasmine IIIp robot; sensors 2 to 7 are using an infra-red light source with an opening angle of 60 degrees to detect obstacles in every direction of vision. Sensor 1 has an angle of 20 degrees to allow detection of more distant obstacles in the front.

Controller Model. Robot controllers are encoded as FSMs which implement a model called *Moore Automaton for Robot Behavior (MARB)* as presented in (König and Schmeck, 2008) and (König et al., 2009). Each state of a MARB defines one elementary action for the robot to execute. The transitions between the states are attached to conditions that can be evaluated using the sensory input of the robot. Conditions can be atomic (i. e., *true*, $false$ or a comparison of two sensor outputs or constants using one of the relations "$<, >, \leq, \geq, =, \neq, \approx, \not\approx$") or conjunctions and disjunctions of other conditions.

Example conditions are: *true*; $h_1 < h_2$; $20 > h_7$; $(h_1 \approx h_2 \; OR \; h_2 \not\approx 120)$. A condition is evaluated to *true* or $false$ by replacing the sensor variables $h_1, \ldots, h_7$ by the current sensor data of the infra-red sensors $1, \ldots, 7$. There, the relation $\approx$ is *true* if and only if the two operands differ by at most 5. For a current state of the automaton the next state is calculated by evaluating the outgoing conditions and taking the first transition whose condition evaluated to *true*.

Fig. 7 shows an example MARB. The dotted transitions do not have to be defined explicitly: to avoid

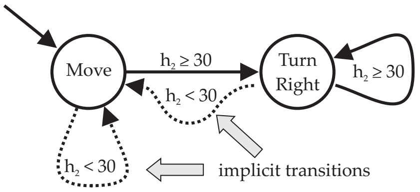


Figure 7: An example MARB with two states. The dotted transitions are inserted implicitly by the model. The states perform the operations Move and Turn Right, respectively. The automaton represents a simple collision avoiding behavior moving forward as long as no obstacle is ahead ($h_2 < 30$) and turning right if an obstacle is ahead ($h_2 \geq 30$).

deadlocks the initial state is always the successor of a state that has no outgoing conditions that evaluate to *true*. For more information on MARBs we refer to (König et al., 2009).

Note that it is not important for the applicability of the prediction model that FSM controllers are used; e. g., artificial neural networks or any other controller model could as well be used.

Scenario. The experimentation environment is given in Fig. 8. In all experiments the populations consist of $n = 30$ robots which are placed at random positions in the environment facing in random directions. The desired behavior (in terms of "superiority") is the capability of driving as far distances as possible. The explicit fitness function f is calculated by summing up every 10 simulation steps the distance driven during the last 10 steps. Additionally, the whole sum is divided by 1.3 afterwards (cf. *evaporation* in (König et al., 2009)). The evolutionary runs are performed until convergence to a stable state.

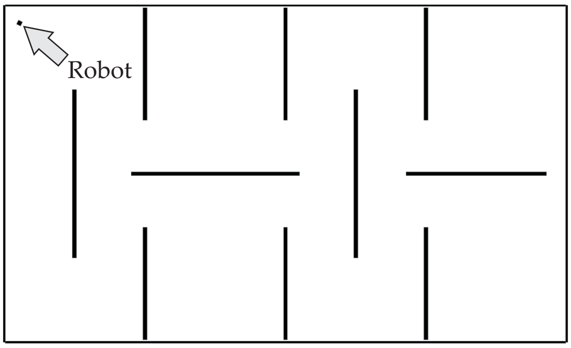


Figure 8: The experimentation environment with a robot drawn to scale. Black rectangles denote walls.

5.2 Experimental Results

We investigate the capability of the EIS model for predicting correctly the probabilities of converging to the superior ($^{n}/_{0}$) or inferior ($^{0}/_{n}$) stable state. As a convergence to the superior state means that the selection mechanisms worked as desired, the percentage of convergence to the superior state can be used as a measure of success. The initial populations are divided in two sets $\mathcal{S}$ and $\mathcal{I}$ of individuals which perform the desired behavior in a superior and inferior way, respectively.

The initial population state is varied within the state space $S' = \{^{1}/_{29}, ^{2}/_{28}, \ldots, ^{29}/_{1}\}$ (the states $^{0}/_{30}$ and $^{30}/_{0}$ are already converged, therefore, they are not tested). The superior individuals are equipped with a wall following behavior that makes the robots explore parts of the arena. The inferior individuals are constantly driving small circles by switching in every other step between a driving and a turning state and, therefore, they are expected to have a lower fitness than the superior ones (although, as in every complex environment, this is not necessarily the case).

In the first tests, a global selection operator is assumed that is based on a fixed probability matrix C to select the mating tournaments (cf. Sec. 3.3), i. e., if a population is in state $^{i}/_{n-i}$, row i of matrix C is used to determine the probabilities for the tournament types ($^{0}/_{k}, \ldots, ^{k}/_{0}$) to select. According to these probabilities, a tournament type $^{j}/_{k-j}$ is selected and such a tournament is chosen randomly from the current population for mating. Afterwards, the tournament winner is chosen according to the explicit fitness as described in Alg. 1. As the quality of the fitness function is not known in advance, the confidence factor c is unknown. The aim of these tests is to show that there exists a confidence factor c such that the experimental data matches the model prediction.

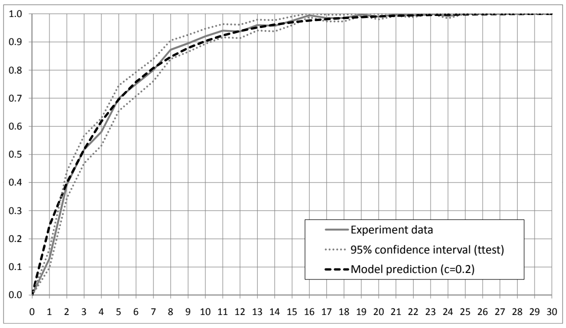


Figure 9: Probabilities of converging to the superior stable state $^{30}/_{0}$ as a function of the initial population state $i \mathrel{\hat{=}} {}^{i}/_{n-i}$. Matrix C is set to a uniform distribution in every row. Tournament size is set to $k = 4$. The gray solid line shows the average values from the experimental data, the gray dotted lines denote the according 95% confidence interval. The black dashed line is the model prediction with $c = 0.2$.

Furthermore, this experimental setup could be used as a preliminary experiment for a real evolutionary scenario to find a c that matches the experimental data best. However, that is only possible if c is independent of the matrix C. This is not the case for the setup described here, see below.

Fig. 9 shows the results of an experiment where the probabilities of matrix C are uniformly distributed within every row, i. e., every tournament formation has an equal selection probability within the same row of C (except for impossible formations which are set to 0, cf. Sec. 3.3). The tournament size was set to $k = 4$. The experiment has been run 400 times for each of the non-stable initial population states $^{1}/_{29}, \ldots, ^{29}/_{1}$.

The chart shows the probabilities of converging to the superior state $^{30}/_{0}$ as a function of the initial population state. The percentage of experimental runs that converged to the superior state are shown by the gray solid line. Two gray dotted lines denote the according 95% confidence interval given by a statistical Student's ttest calculation. The black dashed line denotes a prediction by the EIS model using a confidence factor $c = 0.2$. This confidence factor is determined by a minimal error calculation in steps of 0.1. I. e., all model data points are subtracted from the according experiment data points, adding the absolut values to an error sum; then the value of $c \in \{0.0, 0.1, \ldots, 1.0\}$ is determined for which the error sum is minimal.

As the experimental data follows the model prediction, mostly within the 95% confidence interval, we conclude that the model prediction is accurate in this case.

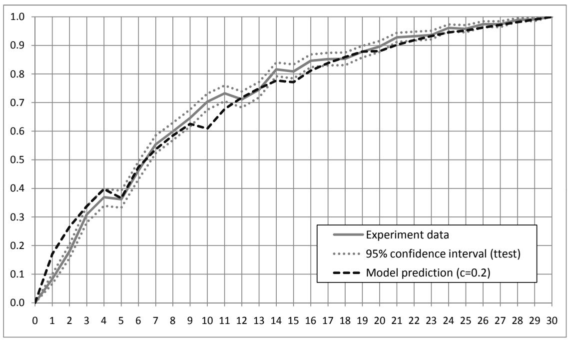


Figure 10: Probabilities of converging to the superior stable state (cf. above chart). Tournament size is set to $k = 8$, matrix C is set to favor $^{4}/_{4}$ tournaments. As above, the gray lines denote experiment data and the black dashed line is the model prediction with $c = 0.2$.

The next experiment is performed with a tournament size of $k = 8$ by using a matrix C that has a high probability of selecting a tournament $^{4}/_{4}$ and a low probability for selecting all the other tournaments. Namely, the probability for selecting a $^{4}/_{4}$ tournament is set to $1 - 10^{-4}$ while all the other possibilities uniformly divide the remaining value of 10^{-4} among them. (In rows where the $^{4}/_{4}$ tournament is not applicable, the other tournaments are uniformly distributed.) Due to the symmetry of the preferred tournament $^{4}/_{4}$ this matrix has a counterintuitive property: using it, the model predicts that there should be "jumps" in the probability plot, i. e., an *increase* in the number of superior individuals in the initial population can cause the probability of converging to a superior state to *decrease*. This is, e. g., the case at population state $^{5}/_{25}$.

This experiment has been repeated 1000 times for each of the 29 non-stable initial population states. The chart in Fig. 10 shows the according experimental results. Again, the best-fitting confidence factor has been calculated to $c = 0.2$. For most of the data points the model lies within the 95% confidence interval which, however, is tighter than in the above experiment. Furthermore, the jumps occur in the experimental data as well, which has been interpreted as a strong indicator that the model works. However, the second jump does not occur at state $^{10}/_{20}$ as predicted, but later at state $^{12}/_{18}$ (the other jumps seem to be at their correct positions). While it is possible that this is due to statistical errors, this seems rather unlikely as the two sequent values for states $^{11}/_{19}$ and $^{12}/_{18}$ are considerably outside the 95% confidence interval. We were not able to establish the reason for this inconsistency, therefore, it has to be left for future work.

The last experiment without mutation is performed in a more realistic scenario using a decentralized selection method. Here, robots are selected for a mating tournament if they came spatially close to each other. After being selected for a tournament, the robots are excluded from selection for 50 steps to allow for a new fitness calculation. The radius for mating is set to 210 mm which is big enough to assure that nearly all runs converged eventually. For the small percentage of runs that did not converge in the first 200,000 simulation steps the run was terminated and counted as superior if the number of superior individuals in the last population was at least 15 and inferior in all other cases. The tournament size was set to 4. This experiment was repeated 400 times for each non-stable initial population state.

In this case the tournaments are selected in a decentralized way, therefore, there is no predefined probability matrix C. Instead, C is given by the environment and the given selection parameters, and can be measured during a run. The chart in Fig. 11 visualizes the probabilities of the matrix found by averaging over all occurred tournaments during all the $400 \cdot 29 = 11,600$ runs of this experimental setup. It

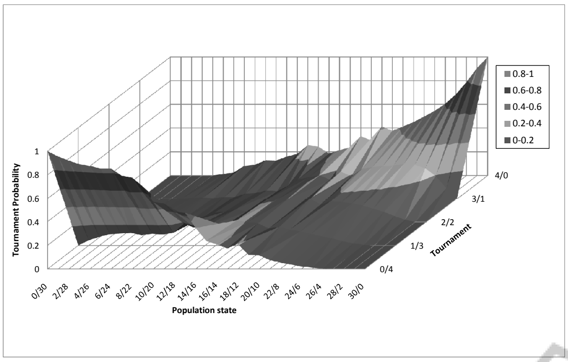


Figure 11: Probability matrix C that originates from the decentralized mating strategy in the third experiment.

can be observed that for a rather heterogeneous population with an approximately equal number of superior and inferior individuals the tournament probabilities are roughly uniform. With more superior individuals in the population, the probability for superior tournaments grows, and the other way around. This seems to be quite intuitive and it can be suspected that similar decentralized selection methods always yield similar probability distributions.

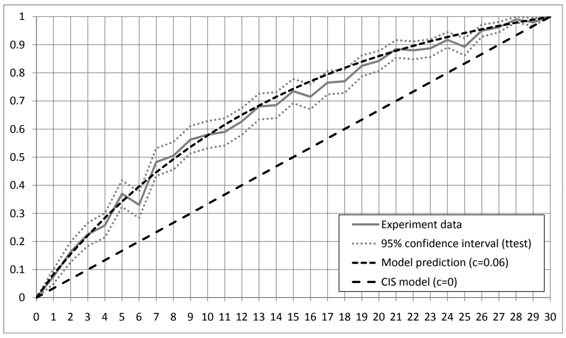


Figure 12: Probabilities of converging to the superior stable state (cf. above charts). Tournament size was set to $k = 4$, matrix C was given implicitly by the decentralized selection method; the according probability values are depicted in Fig. 11. As above, the gray lines denote experimental data and the black short-dashed line is the EIS model prediction with $c = 0.06$. The long-dashed line reflects the CIS model without an explicit fitness function.

Fig. 12 shows the convergence probabilities resulting from this experiment. The EIS model prediction has this time been calculated using the measured values for matrix C as depicted in Fig. 11. The confidence factor c is set to 0.06 which is the best approximation by a precision of 0.01. Additionally, a plot of the CIS model is depicted in the chart as a comparison. Obviously, the chance of converging to the superior state is raised by the explicit fitness function. However, the confidence factor c is decreased considerably compared to the above experiments with a global selection operator. This is a clear sign that the confidence factor is not independent of the matrix C in this scenario. As a consequence, the chance for reaching the superior state is lower in this experiment than in the above experiments. The model prediction is, again, for nearly all data points within the 95% confidence interval.

6 CONCLUSIONS AND FUTURE WORK

In this paper, a mathematical model based on Markov chains has been introduced that can be utilized to estimate the probability that an ESR run will be successful in terms of being capable of improving a population until a desired behavior is found. The first version of the model presented here focuses solely on the selection process. It is assumed that selection is performed by tournament selection which is based on two types of fitness: first an implicit fitness which depends on potentially hidden environmental properties, and second an explicit fitness that is calculated from environmental variables and that can be fuzzy, noisy or delayed. In complex environments both fitnesses may not reflect the desired behavior perfectly (furthermore, especially the implicit fitness is hard to influence and mostly given by the scenario). Our model takes into account the chances for both the implicit and explicit part of the selection process to select the superior of two types of individuals, and calculates the probabilities that a certain population state will eventually converge to a superior state, i. e., a state with only superior individuals. Furthermore, the expected time to convergence in terms of the number of mating events necessary to reach a superior state can be calculated. The model is applicable to nearly all types of ESR scenarios including offline and centralized as well as online and decentralized approaches. It can help predicting the performance of an ESR run in cases where success is of critical importance or where failures are expensive (this is often, but not exclusively, the case in decentralized online scenarios). There are no restrictions to controller types or evolutionary operators except for selection. Experiments in simulation show that the predictions of the model coincide with actual experimental data.

The model depends on a quite large number of parameters arising from probability values that depend on the environment and the fitness calculation procedure. These parameters have a major influence on the results. As the model is only useful if these parameters can be estimated properly before an actual run is performed (e. g., in simulation before starting a real-world run), future work will cover studies of how these parameters can be discovered. One simulation approach to this end has been presented in this paper, but it still has to be studied how well the results match a real-world scenario. The model so far does not cover mutation directly, but assumes rather long selection phases without behavior-changing mutations. Furthermore, while tournament selection is a natural selection method in evolutionary robotics, it is still a constraint of the model that is not categorically necessary. It is planned to extend the model to cover mutations and to be applicable on selection methods different than tournament selection.

ACKNOWLEDGEMENTS

The authors thank Lisa Hofmann for preparatory work on the topic of implicit environmental selection in ESR.

REFERENCES

Arnold, D. V. (2001). *Evolution strategies in noisy environments- a survey of existing work*, pages 239–250. Springer-Verlag, London, UK.

Bredeche, N. and Montanier, J.-M. (2010). Environment-driven embodied evolution in a population of autonomous agents. In *Parallel Problem Solving from Nature*, pages 290–299, Berlin Heidelberg. Springer.

Eiben, A. E., Hinterding, R., and Michalewicz, Z. (2000). Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 3:124–141.

Grinstead, C. and Snell, L. (1997). *Introduction to Probability*. American Mathematical Society.

Gross, R. and Dorigo, M. (2009). Towards group transport by swarms of robots. *Int. J. Bio-Inspired Comput.*, 1:1–13.

Holland, J. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press.

Kessler, D., Levine, H., Ridgway, D., and Tsimring, L. (1997). Evolution on a smooth landscape. *Journal of Statistical Physics*, 87:519–543.

Kimura, M. (1985). *The Neutral Theory of Molecular Evolution*. Cambridge University Press.

König, L., Mostaghim, S., and Schmeck, H. (2009). Decentralized evolution of robotic behavior using finite state machines. *Int. Journal of Intelligent Computing and Cybernetics*, 2(4):695–723.

König, L. and Schmeck, H. (2008). Evolving collision avoidance on autonomous robots. In *Biologically Inspired Collaborative Computing (Proceedings of BICC 2008)*, pages 85–94.

Nolfi, S. and Floreano, D. (2001). *Evolutionary Robotics. The Biology, Intelligence, and Technology of Self-Organizing Machines.* MIT Press.

Pietro, A. D., While, L., and Barone, L. (2004). Applying evolutionary algorithms to problems with noisy, time-consuming fitness functions.

Prügel-Bennett, A. and Rogers, A. (2001). *Modelling genetic algorithm dynamics*, pages 59–85. Springer-Verlag, London, UK.

Sober, E. (2001). The two faces of fitness. In Singh, R., Paul, D., Krimbas, C., and Beatty, J., editors, *Thinking about Evolution: Historical, Philosophical, and Political Perspectives.*

Watson, R., Ficici, S., and Pollack, J. (2002). Embodied evolution: Distributing an evolutionary algorithm in a population of robots. In *Robotics and Autonomous Systems*, pages 1–18.

Weyns, D., Schumacher, M., Ricci, A., Viroli, M., and Holvoet, T. (2005). Environments in multiagent systems. *The Knowledge Engineering Review*, 20(02):127–141.