Evolvability in Evolutionary Robotics: Evolving the Genotype-Phenotype Mapping

Lukas König
Institute AIFB
Karlsruhe Institute of Technology
76128 Karlsruhe, Germany
lukas.koenig@kit.edu

Hartmut Schmeck
Institute AIFB
Karlsruhe Institute of Technology
76128 Karlsruhe, Germany
hartmut.schmeck@kit.edu

***Abstract*—A completely evolvable genotype-phenotype mapping (ceGPM) is studied with respect to its capability of improving the flexibility of artificial evolution. By letting mutation affect not only controller genotypes, but also the mapping from genotype to phenotype, the future effects of mutation can change over time. In this way, the need for prior parameter adaptation can be reduced. Experiments indicate that the ceGPM is capable of robustly adapting to a benchmark behavior. A comparison to a related approach shows significant improvements in evolvability.**

I. Introduction

The evolutionary setup highly affects the success of an evolutionary run. In this paper, an approach is studied to generalize evolutionary setups in order to make them applicable to a greater variety of target behaviors. Any static mutation operator is usually applicable to a small set of target behaviors only. Here, the genotype-phenotype mapping (GPM) is made "completely evolvable", i. e., the mapping from the space of genotypes (encodings of controllers based on sequences of integers) to the space of phenotypes (robot behaviors encoded as finite state machines) can be evolved. In this way, the effects that mutation has on the phenotypes may change during a run when the interpretation of the genotypes changes. Furthermore, in a recursive process, the effects of mutation of the GPM can change over time, thus, potentially improving the evolution of GPMs as well as the evolution of controllers.

A measure for evolvability of a ceGPM is defined and applied to the ceGPM presented in [4] (*ceGPM-old*) as well as to a newly proposed ceGPM (*ceGPM-new*). It is shown that the *ceGPM-old* does not gain evolvability to a statistically significant extent. In contrast, the newly presented *ceGPM-new* is capable of gaining evolvability.

For related work on evolvability, cf. [1], [2], [5].

II. Scenario

Robot platform. The experiments are performed on a simulated swarm of mobile Jasmine IIIp robots. Each robot is sized $26 \times 26 \times 26$ mm^3 and has two wheels as actuators and seven infra-red sensors for distance measurement placed around the top. For more information on the robot platform visit http://www.swarmrobot.org.

Behavioral automaton MARB. The behavioral part of the robot controllers is represented by a model called *Moore Automaton for Robot Behavior (MARB)*, cf. [3] and Fig. 1.

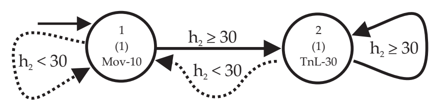

Figure 1. Simple example of a MARB. Implicit transitions to the initial state are indicated as dashed lines.

Translator automaton MAPT and the *ceGPM-old*. Both versions of the ceGPM (*ceGPM-old* and *ceGPM-new*) are based on a translator model called *Moore Automaton for Protected Translation (MAPT)*.

The MAPT model has the same structural properties as the MARB model. It is a translator automaton that uses a *genotype*, i. e., a sequence of numbers as input and produces a sequence of script instructions as output that can be interpreted to produce another automaton. A central idea of the model is that a MAPT produces MARBs, but also MAPTs making it possible to retranslate itself after mutations. Being structurally the same, there are two semantic differences between MARBs and MAPTs: (1) The sensor variables in the MARB model depend on environmental observations of the robot while the sensor variables of a MAPT are fed by virtual sensor values which point to a genotypic sequence. (2) The output of a MARB state is a motoric command while a MAPT's output consists of script instructions.

The *ceGPM-new*. The new ceGPM proposed here extends the described translator model by adding a new instruction to the script language. It is supposed to solve a structural problem of the MARB and MAPT models shown in Fig. 1. Both MARB and MAPT redirect to the initial state if a state has no active outgoing transitions. However, it is essential for both models to be capable of generating interconnected subparts. This is hindered by visiting the initial state too frequently. The new script instruction is $CPL(X, Y)$. It inserts

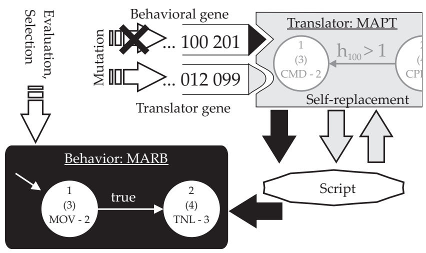

Figure 2. Schematic view of the translation process. Mutation operates on genotypes (turned off for the behavioral genotype here) while evaluation and selection consider the performance of the behavioral automaton only.

a "completing" transition from state X to state Y associating to it a new condition c_{cpl}. Let the outgoing conditions of X be $c_1, \ldots, c_n$; then the new condition associated to the inserted transition is $c_{cpl} = \bigwedge_{i=1}^{n} \neg c_i$.

Universal translator. A universal translator U is a MAPT such that for all MARBs B and MAPTs T there exist genotypic sequences b and t which are translated by U to B or T, respectively. The universal translator used here can be downloaded at http://www.aifb.kit.edu/images/6/6b/Utrans.pdf.

Course of evolution. The evolution process is depicted in Fig. 2. In the beginning, an initial MAPT T^0, an initial translator genotype t^0, and an initial behavioral genotype b^0 are given. T^0 is a manually constructed universal translator as described above; t^0 encodes T^0 in the mapping defined by T^0; b^0 is a randomly generated sequence. T^0 translates b^0 into a script which is interpreted to generate a behavioral automaton B^0. This defines the initial behavior. During the run, random mutations can occur and turn the current translator genotype t^i into some genotype t^{i+1}. This replaces the current translator T^i with a new translator T^{i+1} which is generated by translating the mutated translator genotype t^{i+1} with the old translator T^i. Afterwards, the unchanged behavioral genotype b^0 is translated by T^{i+1} to B^{i+1}, thus changing the robot's behavior without changing the behavioral genotype.

Evolvability measure. Evolvability is defined as an adaptation of the GPM to the behavior being evolved. Therefore, behavioral mutations are turned off meaning that translator mutations are the only mutations that can occur. The adaptation of the translator to the desired behavior is measured.

III. Experiments and Results

Method of experimentation. An initial set of experiments was set up to cover a broad range of 324 different parameter combinations. Each of these combinations has been tested in two separate runs for both the *ceGPM-old* (a) and the *ceGPM-new* (b). Setups from the first group that indicated evolvability according to the proposed measure in at least one of the two identical runs were called *potentially successful setups (PSSs)* and studied further in a second set of

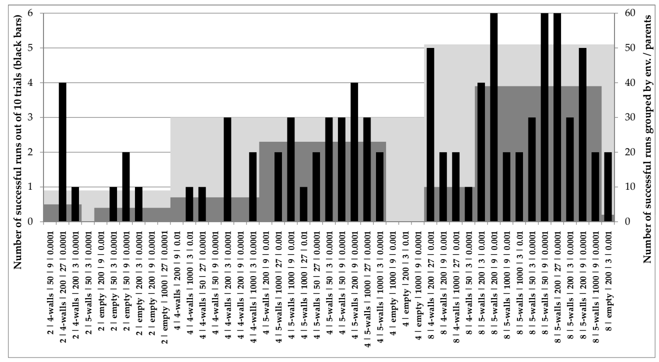

Figure 3. Results of the second set of experiments with the *ceGPM-new* and its own PSSs, i. e., category (3). The ***X***-axis denotes the PSSs by the setting of their variable parameters. The black bars denote the number of successful runs according to the left ***Y***-axis. The gray bars denote the sum of all successful runs grouped by ***num_parents*** or ***env***, respectively, according to the right ***Y***-axis.

experiments. The first setup yielded 12 PSSs for *ceGPM-old* and 45 PSSs for *ceGPM-new*. Fig. 3 shows the performance of the latter in the second set of experiments.

In the second set, both ceGPMs were tested with all PSSs from the initial runs (a) and (b). The second set has been divided into four categories: (1) the *ceGPM-old* with its own PSSs, (2) the *ceGPM-old* with the foreign PSSs, (3) the *ceGPM-new* with its own PSSs, and (4) the *ceGPM-new* with the foreign PSSs. Every run in the second set of experiments has been performed ten times.

Results and discussion. The performance of *ceGPM-old* in the second set of experiments was very low in both its own and the foreign PSSs. Of the 120 runs performed overall in (1) with own PSSs for *ceGPM-old*, only 4 (3.3%) were successful; similarly, of the 450 runs in set (2) with foreign PSSs of *ceGPM-old*, 8 (1.8%) were successful. The runs with foreign PSSs of *ceGPM-new* (4) also did not perform well; of the 120 runs, 5 (4.2%) were successful. None of these setups yielded more than two successful runs out of ten for a specific PSS. In contrast set (3) with own PSSs of *ceGPM-new* yielded 90 (20%) successful runs, cf. Fig. 3.

References

[1] R. Dawkins. The evolution of evolvability. In *Artificial Life Proceedings*, 1987.

[2] V. Feldman. Evolvability from learning algorithms. In *40th annual ACM symposium on Theory of computing*, 2008.

[3] L. König, S. Mostaghim, and H. Schmeck. Decentralized evolution of robotic behavior using finite state machines. *Int. Journal of Intelligent Computing and Cybernetics*, 2, 2009.

[4] L. König and H. Schmeck. A completely evolvable genotype-phenotype mapping for evolutionary robotics. In *Int. Conf. on Self-Adaptive and Self-Organizing Systems*, 2009.

[5] L. G. Valiant. Evolvability. *Electronic Colloquium on Computational Complexity (ECCC)*, 6, 2006.