Incorporating User Feedback in Embodied Evolution

Micha Kemeling
Vrije Universiteit Amsterdam, Department of Computer Science
De Boelelaan 1081 A
Amsterdam, The Netherlands 1081 HV
micha.kemeling@gmail.com

Evert Haasdijk
Vrije Universiteit Amsterdam, Department of Computer Science
De Boelelaan 1081 A
Amsterdam, The Netherlands 1081 HV
e.haasdijk@vu.nl

ABSTRACT

We investigate the possibilities of incorporating user feedback at run-time in an embodied evolution setting. User feedback in this case consists of a user ranking a small sample from the population in an ongoing evolutionary process according to some criterion. We describe an approach to develop a surrogate model, disseminate that model across the evolving robot collective and incorporate it as an additional objective during the evolutionary run. We evaluate the method in a number of scenarios, showing that this approach does to some extent cause the collective to absorb new objectives, including objectives to refrain from particular behaviour, but not to the level that is achieved by explicitly adding the true objective function.

CCS CONCEPTS

•**Computer systems organization** → **Evolutionary robotics;** •**Mathematics of computing** → *Evolutionary algorithms;*

KEYWORDS

Evolutionary robotics, Embodied Evolution, Interactive Evolution

ACM Reference format:
Micha Kemeling and Evert Haasdijk. 2017. Incorporating User Feedback in Embodied Evolution. In *Proceedings of GECCO '17 Companion, Berlin, Germany, July 15-19, 2017,* 2 pages.
DOI: http://dx.doi.org/10.1145/3067695.3082549

1 INTRODUCTION

Embodied Evolution was introduced by Ficici et al. [4] in pursuit of a vision of robot collectives that autonomously adapt to their tasks and environment by means of evolution. In embodied evolution, the evolutionary process is distributed over the robots: in contrast to regular evolutionary algorithms, there is no central authority that regulates parent and survivor selection. Instead, the robots interact locally and autonomously to assess behaviour and to exchange and select genetic material. An important consideration in this context is that the robot collective requires no (human) oversight. Once the objectives for evolution and mechanisms for interaction have been implemented, the system is supposed to take care of itself: "the robot population evolves in a completely hands-free and autonomous manner" [7].

In this paper, we investigate the possibility of sparse user interaction (user-**on**-the-loop) to define new objectives at runtime in an embodied evolution implementation. The aim is to enable a scenario where a user, while the robots are performing their tasks and their controllers evolve, can point out (un)desirable behaviour in a few robots. The robots then process this feedback and insert it into the distributed evolutionary process as a new objective.

Particularly in design applications, interactive evolutionary computation has been successfully employed. Interactive evolution is associated with increased search ability, increased exploration and diversity [2, Ch. 1]. In such cases, there is no explicit model for fitness computation, and the evaluation of the fitness depends on the human user [e.g., 3]. This amounts to a user-**in**-the-loop paradigm where users act as a fitness function and interactively labels or ranks each individual. It was soon realised that integrating user feedback in this manner implies, in the words of Biles [3], a "fitness bottleneck," and work by e.g., Johanson and Poli [6] explored the use of artificial neural networks (ANNs) to serve as a surrogate model of user preferences. Johanson and Poli used hundreds of user-supplied ratings to train an ANNs that then served as automated raters. These "proved somewhat successful, but were not able to generate nice sequences with the reliability of human rated runs."

In reinforcement learning, Akrour et al. [1] have shown that it is possible to learn a ranking function to reflect a user's preferences in an iterative process where the reinforcement learning process proffers candidate solutions for a user to rank.

User interaction to define additional objectives in an ongoing evolutionary process has not been investigated to date, and it is also unclear whether it is feasible to develop surrogate models from infrequent, even one-off, user feedback. Distributed evolutionary systems such as embodied evolution pose a further challenge because the robots are physically distributed without any central entity, so it is improbable that a user could provide feedback encompassing all the robots in the collective.

We investigate whether it is possible to incorporate feedback in an ongoing embodied evolution process, seeking to answer the research question *is it possible to guide the evolutionary process by indicating (un)desirable behaviour in a subset of the population?*

We consider the following set-up: during a run, a subset of the robot collective is ranked according to a new objective (this can relate to desired as well as to undesired behaviour). A surrogate model is then built on the basis of this ranking, and the ranked

GECCO '17 Companion, Berlin, Germany

978-1-4503-4939-0/17/07…$15.00
DOI: http://dx.doi.org/10.1145/3067695.3082549

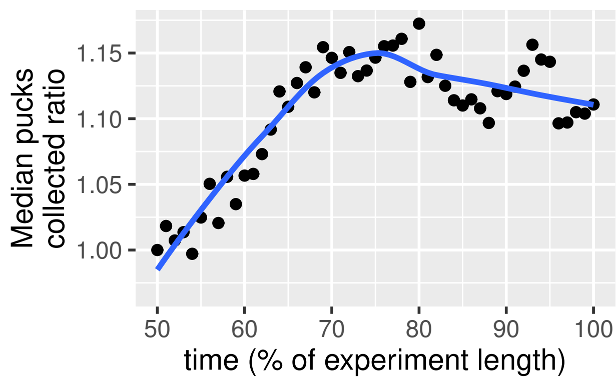

Figure 1: Median number of type 3 pucks collected after user feedback compared to runs without user feedback. The feedback indicates to start collecting these pucks that played no role previously.

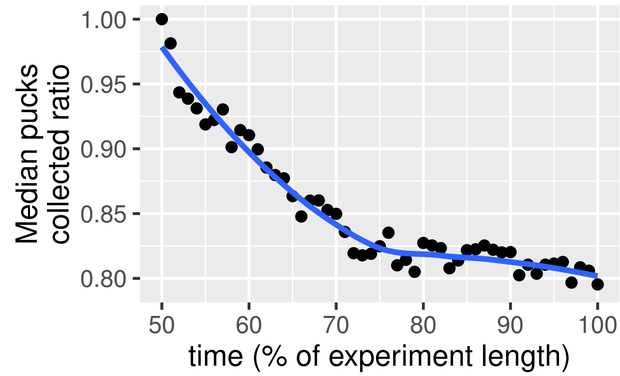

Figure 2: Median number of type 2 pucks collected after user feedback compared to runs without user feedback. The feedback indicates to stop collecting these pucks.

robots start using this as a new objective in addition to any objectives that were already defined. The robots distribute the new objective across the collective by means of gossiping.

The experiments extend those reported by Haasdijk et al. [5] where robots move around an arena with distinct types of puck that the robots can sense and pick up. The baseline set-up (i.e., without any user guidance) defines the task to pick up pucks of types 1 and 2 (there are 4 types in total), other puck types are ignored. After 500 000 iterations, we simulate user interaction to add an objective: 15 robots are ranked according to the new objective, and a linear regression model is created to map a robot's self-metrics (nr. of pucks collected per type, distance travelled, obstacles hit, and position) to a ranking. This model is then used as an additional objective and disseminated across the robot collective through a gossiping approach. We conduct two sets of experiments: in the first set, the additional objective is to collect pucks of type 3. The second set adds the objective to stop collecting pucks of type 2.

2 RESULTS AND DISCUSSION

The plots in fig. 1 and 2 show how the robot population reacts to the user intervention; they show the ratio of the number of pucks (of type 3, resp. type 2) collected between the baseline runs and the runs where the new objectives are added. The plots show median results over 50 replicate runs. The increase in collections of puck type 3 (fig. 1) is low; the robots eventually pick up between 10 and 15% more pucks compared to the baseline without the new objective. The decrease in number of pucks of type 2 (fig. 2) is much more substantial: the robots pick upto 20% fewer pucks than they do without feedback.

These results shows that it is possible to add objectives to an embodied evolution process as the run progresses. The change in behaviour is modest but significant, even with a simple linear surrogate model, and dissemination through a gossiping protocol suffices. It seems to be somewhat easier to reduce undesirable behaviour than to promote new behaviour, but further analysis will have to confirm this.

It remains to be investigated whether the modest level of behavioural change is due to the limitations of surrogate modelling (although initial experiments with more complex neural net-based models) or to the sparsity of feedback in this set-up. Another intriguing possibility is that the addition of new objectives during an evolutionary run necessitates an increase in controller complexity or partial re-initialisation to escape from achieved convergence.

These results are encouraging, even though there is obvious room for improvement. Extending embodied evolution to incorporate user-on-the-loop feedback enables a responsive adaptivity, where the user can easily intervene to guide adaptation as (s)he sees fit.

REFERENCES

[1] Riad Akrour, Marc Schoenauer, and Michele Sebag. 2011. Preference-Based Policy Learning. In **Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011. Proceedings, Part I**, Dimitrios Gunopulos, Thomas Hofmann, Donato Malerba, and Michalis Vazirgiannis (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 12–27. DOI: http://dx.doi.org/10.1007/978-3-642-23780-5{_}11

[2] Peter Bentley and David Corne. 2002. **Creative Evolutionary Systems**. Morgan Kaufmann, San Francisco.

[3] John Biles. 1994. GenJam: A genetic algorithm for generating jazz solos. In **Proceedings of the International Computer Music Conference**. International Computer Music Association, Michigan Publishing, Ann Arbor, MI, 131–137.

[4] Sevan G Ficici, Richard A Watson, and J Pollack. 1999. Embodied Evolution: A Response to Challenges in Evolutionary Robotics. In **Proceedings of the Eighth European Workshop on Learning Robots**, J L Wyatt and J Demiris (Eds.). Springer-Verlag, London, UK, 14–22. citeseer.ist.psu.edu/article/ficici99embodied.html

[5] Evert Haasdijk, Nicolas Bredeche, and A E Eiben. 2014. Combining Environment-Driven Adaptation and Task-Driven Optimisation in Evolutionary Robotics. **PLoS ONE** 9, 6 (2014), e98466. DOI: http://dx.doi.org/10.1371/journal.pone.0098466

[6] Brad Johanson and Riccardo Poli. 1998. GP-Music: An Interactive Genetic Programming System for Music Generation with Automated Fitness Raters. In **Proceedings of the Third Annual Conference**. MIT Press, Cambridge, MA, 181–186.

[7] Richard A Watson, Sevan G Ficici, and Jordan B Pollack. 2002. Embodied Evolution: Distributing an evolutionary algorithm in a population of robots. **Robotics and Autonomous Systems** 39, 1 (4 2002), 1–18. http://eprints.ecs.soton.ac.uk/10620/