

Right on the MONEE

Combining Task- and Environment-Driven Evolution

Evert Haasdijk
e.haasdijk@vu.nl

Berend Weel
b.weel@vu.nl

A.E. Eiben
a.e.eiben@vu.nl

VU University Amsterdam
The Netherlands

ABSTRACT

Evolution can be employed for two goals. Firstly, to provide a force for adaptation to the environment as it does in nature and in many artificial life implementations – this allows the evolving population to survive. Secondly, evolution can provide a force for optimisation as is mostly seen in evolutionary robotics research – this causes the robots to do something useful. We propose the MONEE algorithmic framework as an approach to combine these two facets of evolution: to combine environment-driven and task-driven evolution. To achieve this, MONEE employs environment-driven and task-based parent selection schemes in parallel.

We test this approach in a simulated experimental setting where the robots are tasked to collect two different kinds of puck. MONEE allows the robots to adapt their behaviour to successfully tackle these tasks while ensuring an equitable task distribution at no cost in task performance through a market-based mechanism. In environments that discourage robots performing multiple tasks and in environments where one task is easier than the other, MONEE's market mechanism prevents the population completely focussing on one task.

Categories and Subject Descriptors

I.2.8 [Artificial Intelligence]: Problem Solving, Control Methods, and Search—*Heuristic methods*

Keywords

Embodied Evolution, Task Distribution, Multi-Objective Optimization, Open-ended Evolution, Evolutionary Swarm Robotics, On-line Evolution

Track

Artificial Life/Robotics/Evolvable Hardware (ALIFE)

1. INTRODUCTION

In real life, evolution is objective-free: there is no objective function that scores living organisms and determines their

chances of producing offspring. By virtue of this unbounded nature, biological evolution has resulted in the high levels of adaptability and robustness that we see in natural living organisms. To exploit this creative potential in a system of evolving robot controllers, we would want to give evolution as much freedom as possible, pushing for open-ended, unbounded adaptivity, not constrained by user-defined objective functions.

On the other hand, we want to ensure that the robots perform meaningful tasks if the system is to be of any practical relevance, pushing for specific task-related objectives. The ability to balance these aspects of evolution represents a vital step towards implementing the vision that underlies our research, which is one of autonomous, functional, responsive and self-sufficient robot collectives that can cope with situations unforeseen by their designers.

In this paper we present MONEE (Multi-Objective aNd open-Ended Evolution) to solve the problem of combining environmentally driven and task-driven evolution in a single algorithmic framework. The principal idea behind MONEE is to employ two basic selection mechanisms in different roles: environmental selection for open-ended evolution and parent (or mate) selection for task-driven adaptation.

We investigate a proof-of-concept implementation of the MONEE paradigm where a population of robots has two tasks: it must collect red and green pucks. The corresponding research question we seek to answer first here is:

I Does MONEE indeed promote task-driven behaviour? In other words, given a scenario with some task(s) for the robots –and measurable task performance– will the robots adapt their behaviour to perform the task(s)?

As the 'Multi-Objective' part of the name implies, MONEE accommodates settings with multiple tasks. As Jones and Mataric note [7], collectively tackling multiple tasks also entails a division of work. If there are multiple tasks, the population of robots as a whole must tackle all of them, even though individual robots may specialise in only a subset. To cope with such cases the MONEE framework uses a market mechanism. This mechanism regulates task-based rewards during mate selection according to the market logic that scarcity increases worth. In our multiple task context this implies that tasks that only a few robots (can) perform yield relatively high rewards and therefore higher selection probabilities. This raises the second research question:

II Does MONEE and its market mechanism provide for equitable task distribution? That is, in a setting with

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO'13, July 6–10, 2013, Amsterdam, The Netherlands.
Copyright 2013 ACM 978-1-4503-1963-8/13/07 ...\$15.00.

multiple tasks, will the population perform all tasks or will it focus on only a subset, in particular if individual robots are forced to specialise in single tasks?

Scenarios where some tasks are easier than others run the risk of focussing only on the simpler tasks at the cost of underachieving on the more challenging ones. Therefore, we also investigate a skewed scenario, where one task is easier than the other to address the third research question:

- III Does MONEE with its market mechanism provide for equitable task distribution when one task is easier than the other?

2. RELATED WORK

Evolutionary Robotics has been widely studied since the early 1990s [15]. Initially, research focussed on individual robots, but since then substantial effort has been directed at evolution in larger numbers of interacting autonomous robots in swarms [21], research projects include for instance the Swarmanoid project [6]) or modular robots (e.g. M-tran[8]). In all these cases, evolution is used to achieve some fixed user-defined objective such as locomotion or explicit coordination.

Objective-free evolution as well as self-replication have been studied in Artificial Life since Rasmussen's (1990) [16] and Ray's (1991) [17] work. Such research primarily investigates evolutionary dynamics in the absence of tasks, but as a result of implicit or environmental criteria that impact the ability to spread genomes through the population. Such open-ended approaches have gained interest from the evolutionary robotics community, for instance in Bianco and Nolfi's experiments with self-assembling organisms [2] and more recently in the MEDEA algorithm [3].

Open-ended approaches have been considered as a strategy to promote behavioural diversity in multi-objective settings by, for instance, Mouret and Doncieux [13]. Lehman and Stanley's novelty search [9] also embraces open-endedness to tackle elusive problems where a straightforward objective function leads to sub-optimal behaviour. These recent advances do define objective functions, though: the definition of novelty for Lehman and Stanley and the secondary objective for Mouret and Doncieux are ad-hoc, task-specific definitions of behavioural diversity that amount to tangential and creative redefinitions of the original objective function. Thus, such methods are not the completely objective-free approaches where survival and rate of procreation determine fitness rather than the other way around.

Bredèche et al. describe MEDEA [3], an open-ended evolutionary algorithm where autonomous robots move around an arena while continually broadcasting their genome over a short range. Meanwhile, they also receive genomes from other robots that come in communication range. When a robot's lifetime expires, it randomly selects one of the received genomes, modifies that using mutation and starts a new life of broadcasting this new genome. This set-up promotes, with only environmental selection, robot movement through the environment: genomes that cause the robot to move around a lot are spread at a much higher rate than genomes that cause their host to stand still.

MONEE extends this open-ended approach with a currency-based system where an individual can accumulate credits through task performance – the better a robot performs

a task, the more credits it earns for that task. When an individual puts its genome forward as a potential parent, it also passes information on its earnings for each defined task as an indication of its worth. The genomes with the highest associated credits are then selected to produce new offspring (inspired by [19], but there an individual's capital was fixed and did not reflect proficiency at any task).

The MONEE scheme is reminiscent of parental investment, which has been investigated in ALife settings, including experiments with robots [10, 22, 19]. In artificial life parental investment is often used to give the offspring a starting value of (virtual) energy [11, 12, 5, 18] and a parent's energy level is often linked to task performance (e.g., agents tasked with eating grass to gather energy [5]). The MONEE scheme differs subtly but crucially from such parental investment schemes: a parent does not actually invest when impregnating an egg because the credits aren't transferred but copied; there is no cost involved.

Distributed on-line evolutionary systems such as Watson et al's embodied evolution employ task-related (virtual) energy to determine parent and survivor selection [24, 25], but these consider single objectives only.

Market-based schemes provide a well known solution to the task allocation problem in multi-agent and multi-robot settings, for instance in [23, 20]. Market-based parent selection MONEE exploits this method to achieve multi-objective task-driven adaptation of robot behaviour.

Fitness sharing is a well-known technique that was introduced to promote genetic diversity and so prevent premature convergence in evolutionary algorithms. With fitness sharing, an individual's fitness is reduced if there are many similar (in terms of their genetic makeup) individuals in the population. MONEE's market mechanism is similar in the sense that it also reappraises fitness, favouring tasks that are less commonly tackled by robots in the population. A crucial difference with traditional fitness sharing is that MONEE considers an individual's *behaviour*, not its genetic make-up (although syntactic fitness sharing in genetic programming shares this distinction [14]). Maybe more importantly, MONEE modifies fitness not to prevent premature convergence, but to ensure that the robot population tackles multiple tasks. Traditionally, fitness sharing is not necessarily associated with multiple objectives, but with maintaining diversity in general – typically, but not exclusively, in single-objective settings.

3. MONEE: MULTI-OBJECTIVE & OPEN-ENDED EVOLUTION

The core of MONEE is inspired by the MEDEA algorithm described by Bredèche et al. [3] and Schwarzer's artificial sexuality algorithm [19].

The robot –actually, their controllers'– lifecycle in MONEE consists of two phases: life and rebirth. The robot controllers have a limited, fixed, lifetime during which they perform their actions; moving about, foraging, et cetera. When their lifetime ends, they enter the rebirth phase and become 'eggs': stationary receptacles for genomes that are transmitted by passing live robots. This rebirth phase also lasts a fixed amount of time, at the end of which the egg selects parents from the received genomes to create a new controller. The robot then reverts to the 'life' role with this new controller. Thus, robots (or rather, their controllers)

can procreate by transmitting their genome to eggs, and the more eggs a robot inseminates, the more chances it has for procreation. Because the transmission of genomes is continuous and at close range (e.g. through infrared), the more a robot moves about the arena, the better its chances of producing offspring. This aspect of MONEE is open-ended in the sense that it is objective-free: there is no calculated performance measure that defines the chances of being selected as parent, there is no task. Only the environment dictates what robots may or may not become parents.

To add task-driven parent selection to this basic evolutionary process, the robots can, during their lifetime, amass credits by performing tasks. For instance, a robot could get one credit for every piece of ore it collects, one for successfully solving some puzzle, and so on. If multiple tasks are defined, the robots maintain separate counts for the credits awarded for each task, for instance one counter for the pieces of ore collected and another one for the number of puzzles solved. When a robot inseminates an egg, it passes the current credit counts along with the genome and the egg uses that information to select parents when it revives.

When a robot's egg phase finishes, it compares the parents' credits for each genome it has received. To enable this comparison across tasks, the egg calculates an exchange rate between tasks. This ensures that genomes that invest in tasks for which few credits are found overall (presumably hard tasks) are not eclipsed by genomes that favour easier tasks. The pseudo-code in algorithm 1 details this market mechanism.

The credits relate task performance to reproductive success: besides the open-ended goal of 'merely' transmitting genomes to eggs, robots must also become proficient at the defined tasks for these genomes to be selected. The more proficient a robot is at a task, the higher its chances of procreating. The comparison of credits across multiple tasks introduces an exchange rate between the earnings per task: the more common credits are for a particular task, the less their worth and vice versa. Thus, parent selection becomes a marketplace for skills and features that the user requires. This system naturally caters for multi-objective approaches and allows the user to prioritise tasks in a straightforward manner.

```

for every defined task do                                // total credits
  for every received genome do
    | creditstask ← creditstask + genome.creditstask
  end
  creditsoverall ← creditsoverall + creditstask
end
for every defined task do                                // exchange rate per task
  | ratetask ← creditsoverall / creditstask
end
for every received genome do                              // credits per genome
  for every defined task do
    | genome.rating ←
    | genome.rating + (genome.creditstask · ratetask)
  end
end
// select, mutate and revive
parent ← rank_based_selection(received_genomes)
child ← mutate(parent)
reactivate(child)

```

Algorithm 1: MONEE's market mechanism

4. EXPERIMENTAL SET-UP

We implemented the MONEE algorithm in a simple 2D simulator called RoboRobo[4]. In our experiments, 100 simulated e-pucks are placed in an environment that contains obstacles and pucks. The sides of the square arena are roughly 330 robot body lengths long (1024 pixels in the simulator), and it contains a number of obstacles (see Fig. 1). We run 64 repetitions of each experiment.

There are two types of puck: green and red, defining a concurrent foraging scenario. Concurrent foraging is a variation of regular foraging where the arena is populated by multiple types of objects to be collected [7], rather than just a single resource. In our case, these objects are green and red pucks and the collection of each different colour is a different task. The pucks are spread throughout the arena, and they are immediately replaced in a random location when picked up. The robots move around the arena, spreading their genome as they encounter eggs and dying when their allotted time has passed. They collect pucks simply by driving over them and the more pucks they gather, the more likely their genome is to be selected once an egg they impregnated revives.

To detect pucks, the robots have 16 sensors that detect either red or green pucks (i.e., 8 sensors per puck-type). Each set of 8 sensors is laid out in the same manner as the standard e-puck infrared sensors: 6 face forward, 2 face to the rear. Because individual puck sensors only detect a single type of puck, collecting one type of puck is a task distinct from (but very similar to)

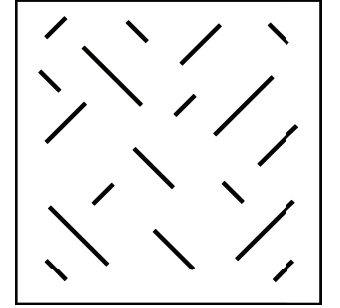


Figure 1: Experiment arena.

collecting the other type of puck. Thus, behaviour to collect either type of puck has to evolve separately.

Each robot is controlled by a single-layer feed forward neural network which controls its left and right wheels. The inputs for the neural network are the robot's puck and obstacle sensors.

The robot's genome directly encodes the neural network's weights (3 types of sensor $\times$ 8 sensors $\times$ 2 outputs plus 2 bias connections plus 4 feedback (current speed and current rotation to either output) = 54 weights) as an array of reals.

As mentioned, the robots alternate between periods of explorative puck gathering and motionless genome reception. To prevent synchronised cycles among the robots, we add a small random number to each robot's fixed lifetime. This forces desynchronised switching between life and re-birth even though our runs start with all robots perfectly in sync at the first time-step of their lifetime.

At the end of the egg phase offspring is created by selecting a parent from the received genomes as shown in algorithm 1 and mutating the weights in that genome using gaussian perturbation with a single, fixed mutation step size $\sigma = 1$. This single-parent, mutation-only scheme is common in evolution strategies that are known to perform well on problems with continuous-valued genomes [1].

Note that MONEE does not prescribe any particular controller implementation nor any choice of variation operator.

The implementation we chose here of an artificial neural network with the weights encoded as real-valued genes provide a convenient, flexible and well-established representation.

Experiment details	
Robot group size	100
Simulation length	1,000,000 time-steps
Number of repeats	64
Number of pucks	150 or 50 green, 150 red
Arena	See fig. 1
Controller details	
Controller	Perceptron neural net
Input nodes	8 obstacle sensors, 16 puck detectors, 2 bias and 2 recurrent nodes
Output nodes	2 (left and right motor values)
Evolution details	
Representation	Real valued vectors
Chromosome length	54
Mutation	Gaussian $N(0, 1)$
Parent selection	Rank-based
Robot lifetime	2000 time-steps
Egg-phase	200 time-steps
Comm. range	ca. 9 body lengths

Table 1: Experimental set-up

Table 1 summarises the experimental set-up as outlined above. To assess the level of environmental and task-related adaptation (research question I) we define a control experiment with random parent selection in the eggs: just as in mEDEA, only the environment determines a genome’s chances of being selected and the number of pucks collected has no influence whatsoever. To answer research questions II and III, we define eight variants of the experiment: configurations with and without market, with multi- or mono-skilled robots and with even or uneven distributions of green and red pucks. These configurations are explained in the remainder of this section.

Does the market matter? The market mechanism is a defining feature of the MONEE paradigm. It causes less commonly tackled tasks to reap higher rewards because of the exchange-rate per task. It would be more straightforward to simply use the total number of pucks collected, regardless of their colour, as the basis for parent selection inside an egg (of course, the environmental selection caused by the robots having to come close to eggs to inseminate them remains unaltered). To determine the influence of the market mechanism we run one set of experiments with the market mechanism enabled, where credits for ‘rare’ tasks are valued higher than those for ‘common’ tasks. As a control, we run one set with the market mechanism disabled, so that genomes are selected on the basis of their total number of credits (pucks collected, in our case) without regard for their relative rarity.

Mutually exclusive skills. Equitable task distribution is more challenging when the tasks that the robots must perform are to some extent exclusive, for instance because they

require irreconcilable skills. To test whether MONEE can handle such situations, we also run experiments where the environment constrains multi-skilled robots so that the robots must specialise in collecting one type of puck. Without this constraint, robots can collect green and red pucks equally well without any penalty when selecting both or merely one colour. In the mono-skill experiments the speed of robots depends on their specialisation level: the robot’s speed is multiplied by the ratio of most prevalent pucks it has collected. Thus, if a robot collects exclusively pucks of one colour, its speed is maximal. If it collects 75% green (or red) pucks, its speed is reduced by 25% and if it collects red and green pucks in equal amount, the speed is halved. This penalty is recalculated whenever a robot picks up a puck. It is important to note that this is enforced by the environment, not during the parent selection phase when an egg revives. The environment causes specialising robots to move faster, so that they perform better than non-specialised robots: their higher speed allows them to collect more pucks during their lifetime, but more importantly, it allows them to impregnate more eggs. This results in an increase in the proliferation of mono-skilled genomes without altering the selection process inside the eggs.

Uneven distribution of pucks. Another determinant for the difficulty of task distribution in our experiments is the ratio of puck colours. This can be seen as a proxy for having a difficult (rare pucks) and an easy (common pucks) task. To determine how a robot swarm with MONEE handles uneven distributions of pucks, we run two variants of our experiments: one with 150 pucks of each colour, one with 50 green and 150 red pucks. An equitable task distribution would lead to robot swarms that collect pucks in the natural ratio as they are found in the environment: 50% and 25% green pucks in the respective set-ups.

5. RESULTS AND ANALYSIS

Research question I. First and foremost, we want to establish whether the robots actually learn to gather pucks (do something useful) and whether they learn to cope with the environment (survive) under the MONEE regime. Figure 2 shows clearly that the population does learn to tackle pucks increasingly well after a brief initial phase with MONEE. There is no appreciable difference in the number of pucks collected with and without the market mechanism. Obviously, the baseline algorithm (mEDEA, where the number of pucks collected has no influence on parent selection) collects far fewer pucks – there is no pressure to adapt behaviour to collect pucks and they are collected accidentally while moving about to spread genomes.

As a measure of adaptation to the environment, we count the number of times genomes were received by eggs. Figure 3 shows the median number of inseminations per 1,000 time-steps for MONEE and for the baseline mEDEA implementation. The initial peak is caused by the fact that the robots are concentrated in a small part of the arena initially. As they spread out over the available area, the number of inseminations first decreases and then recovers as the robots adapt their behaviour. There is no appreciable difference between vanilla mEDEA and MONEE in terms of this measure of environmental adaptation.

Research question II. The second question we posed is whether MONEE provides for equitable task distribution over the robot collective. To answer this question, we consider the ratio between the number of green and red pucks collected by the population: this ratio should reflect the ratio in which the pucks are distributed throughout the environment. Thus, the percentage of green pucks collected indicates whether both tasks have been tackled equally successfully: an equitable task distribution would lead to populations where 50% (or 25 % in with the uneven distributions) of the collected pucks is green.

Figures 4 and 5 show the distribution of the ratio of green to red pucks gathered by the populations in the final stages of the experiments with equal amounts of red and green pucks in the multi- and mono-skill environment, respectively. In the multi-skilled setting, the percentage of green pucks collected tends to the natural ratio of 0.5. With the market mechanism enabled, the distribution is more closely concentrated around this natural ratio than it is without the market.

We are particularly interested in how this scheme holds up in environments where individual robots are forced to specialise in a subset of the tasks.

Figure 5 shows that in such mono-skill settings the market mechanism is essential to keep the population from ‘tipping’ – focussing on one task to the exclusion of the other. Although the distribution with market enabled is not as neatly focussed as it is in the multi-skilled setting, the population still collects both puck types in more or less equal amounts. Without the market mechanism, the majority of experiments resulted in a population that almost exclusively

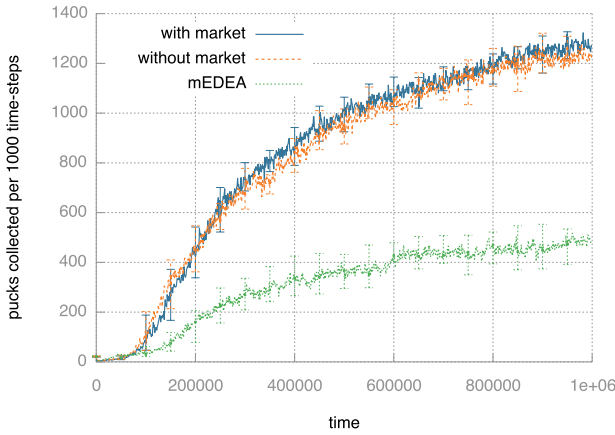


Figure 2: Median number of pucks collected by the population per 1,000 time-steps. Plots show results for MONEE with and without market mechanism and with random parent selection (i.e., without any referral to the number of pucks collected – mEDEA). The vertical bars indicate the 95% confidence interval for the medians. The robots clearly adapt behaviour to collect pucks. The number of pucks collected barely differs whether the market mechanism is in force or not. With random parent selection, the robots gather far fewer pucks: collection is a result of accidentally running over them during random movement.

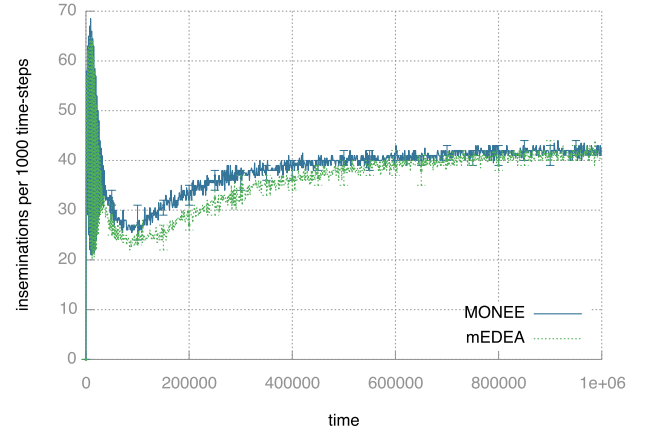


Figure 3: Median number of egg inseminations per 1,000 time-steps with MONEE and with random parent selection (mEDEA). The vertical bars indicate the 95% confidence interval for the medians. Both curves indicate successful adaptation to the environment as robots become increasingly adept at spreading the genomes.

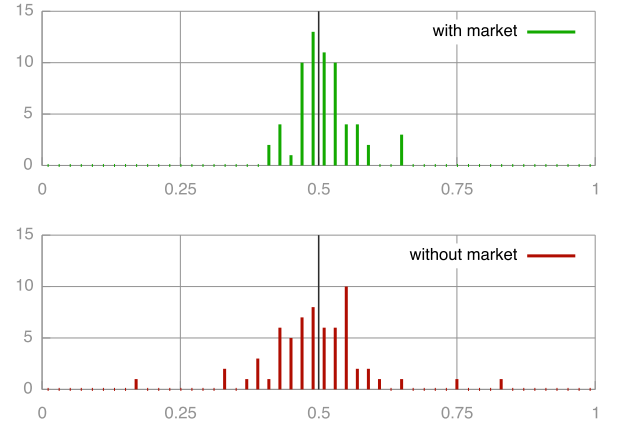


Figure 4: Histograms of green puck ratios across the population with (top) and without (bottom) market mechanism over the final 1,000 time steps of simulation in the multi-skilled setting. The environment contains equal amounts of red and green pucks. The distribution with market mechanism is tighter around the ‘natural’ ratio at 0.5. A two-sample Kolmogorov-Smirnov test to compare the distributions yields $p = 0.0803$.

collects puck of one colour or the other. Rarely does the population gather even roughly the same number of green and red pucks.

Research question III. When there are substantially fewer green pucks than red ones, there is a risk of the population neglecting the green pucks because collecting red pucks is relatively easy. Figure 6 (bottom graph) shows that this does indeed happen in the mono-skill environment when no market mechanism is employed: the ratio of green to red

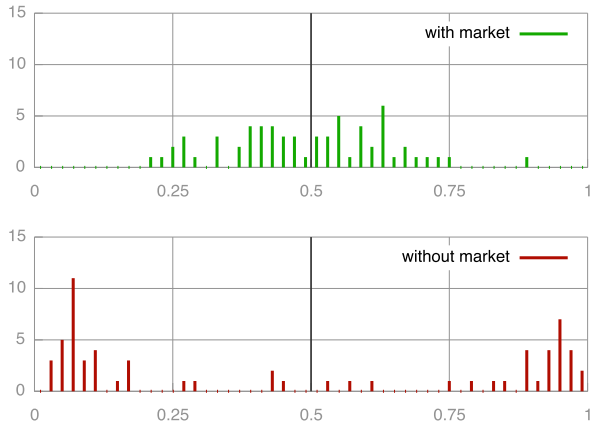


Figure 5: Histograms of green puck ratios across the population with (top) and without (bottom) market mechanism over the final 1,000 time steps of simulation with specialisation. The environment contains equal amounts of red and green pucks. Without the market mechanism, the robot collective tends to specialise in one type of puck, indicated by the two peaks near the extremes. A two-sample Kolmogorov-Smirnov test to compare the distributions yields $p = 7.7438 \times 10^{-07}$.

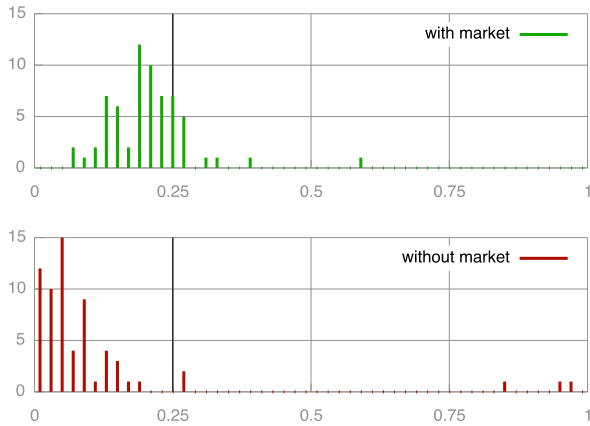


Figure 6: Histograms of green puck ratios across the population with (top) and without (bottom) market mechanism over the final 1,000 time steps of simulation in the mono-skill setting. The environment contains 150 red and 50 green pucks. Without the market mechanism, the robot collective tends to disregard the green pucks. A two-sample Kolmogorov-Smirnov test to compare the distributions yields $p = 1.3980 \times 10^{-16}$.

pucks tends to be close to 0 and only a few of the pucks collected are green. With the market mechanism in effect, however (top graph), the population maintains a substantial (although lower than the natural) ratio of ca. 20% green pucks.

Figure 7 shows the results in the multi-skilled environment: the market mechanism causes the population to col-

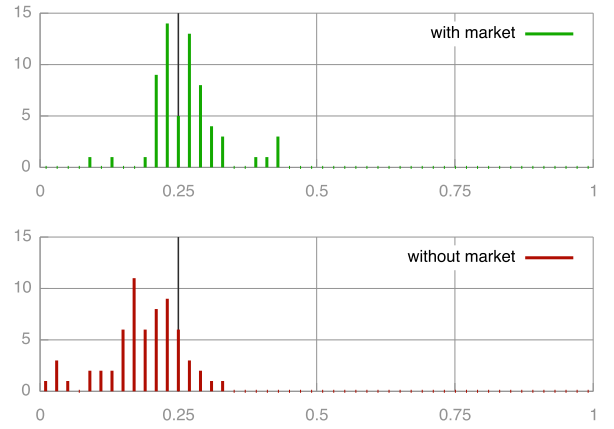


Figure 7: Histograms of green puck ratios across the population with (top) and without (bottom) market mechanism over the final 1,000 time steps of simulation in the multi-skilled setting. The environment contains 150 red and 50 green pucks. The distribution with market mechanism is substantially tighter around the ‘natural’ ratio at 0.25. A two-sample Kolmogorov-Smirnov test to compare the distributions yields $p = 3.4867 \times 10^{-08}$.

lect the 25% green pucks that we would expect from an equitable task distribution. Without the market the ratio of green pucks is substantially lower at ca. 20%.

6. CONCLUSION

In this paper we introduced the MONEE algorithmic framework as a method of combining objective-free environment-driven evolution with task-driven evolution in a population of autonomous robots. To achieve this, MONEE employs two selection mechanisms: the environment is defined so that robots must move around to literally spread their genome. Robots in a passive ‘egg’ state listen for broadcast genomes and use task-dependent credits associated with the genomes for parent selection.

In a simulated experimental setting where the robots are tasked to collect two different kinds of puck, we have shown that MONEE allows the robots to adapt their behaviour to successfully tackle tasks: the robot population gathers increasingly more pucks under the MONEE regime. Also, taking the number of inseminations as a measure of adaptation to the environment, we have shown that MONEE results in a similar level of adaptation to the environment as purely environment-driven evolution.

MONEE’s market mechanism ensures an equitable task distribution at no cost in task performance. This proved particularly beneficial in environments that discourage robots performing multiple tasks. We found that in such mono-skill environments, the market mechanism prevents the population completely focussing on one task.

Similarly, we found that the market mechanism prevents the population focussing on the easier of two tasks. With an uneven distribution of pucks, the market mechanism prevents the smaller portion of pucks to be disregarded, in particular when the environment causes individual specialisation.

This new paradigm opens the door to significant further research: we feel that the successful combination of open-ended, survival-driven and objective-based, task-driven evolution is a crucial step on the road towards collectives of autonomous robots that can adapt to and operate effectively in unforeseen and dynamic circumstances. These two aspects of evolution combined can equip the robots with the adaptivity that coping autonomously with such uncertainty requires.

Further investigations into the MONEE paradigm are ongoing. We are currently investigating the effects of user-defined premiums to prioritise tasks and the resilience of MONEE implementations to dynamic task compositions.

7. ACKNOWLEDGMENTS

This work was in part made possible by the European Union FET Proactive Initiative: Pervasive Adaptation funding the SYMBRION project under grant agreement 216342. The authors would like to thank Nicolas Bredèche, Jean-Marc Montanier, Nikita Noskov and our partners in the SYMBRION consortium for many inspirational discussions on the topics presented here. We thank SARA Computing and Networking Services (www.sara.nl) for their support in using the Lisa Compute Cluster.

8. REFERENCES

- [1] H.-G. Beyer and H.-P. Schwefel. Evolution strategies – A comprehensive introduction. *Natural Computing*, 1:3–52, 2002.
- [2] R. Bianco and S. Nolfi. Toward open-ended evolutionary robotics: evolving elementary robotic units able to self-assemble and self-reproduce. *Connection Science*, 4:227–248, 2004.
- [3] N. Bredèche, J.-M. Montanier, W. Liu, and A. F. Winfield. Environment-driven distributed evolutionary adaptation in a population of autonomous robotic agents. *Mathematical and Computer Modelling of Dynamical Systems*, 18(1):101–129, 2012.
- [4] N. Bredèche, J.-M. Montanier, B. Weel, and E. Haasdijk. Roborobo! a fast robot simulator for swarm and collective robotics. *CoRR*, abs/1304.2888, 2013.
- [5] M. Burtsev, V. Red’ko, and R. Gusarev. Model of evolutionary emergence of purposeful adaptive behavior. the role of motivation. In J. Kelemen and P. Sosík, editors, *ECAL*, volume 2159 of *Lecture Notes in Computer Science*, pages 413–416. Springer, 2001.
- [6] M. Dorigo, D. Floreano, L. M. Gambardella, F. Mondada, S. Nolfi, T. Baaboura, M. Birattari, M. Bonani, M. Brambilla, A. Brutschy, D. Burnier, A. Campo, A. Christensen, A. Decugnière, G. Di Caro, F. Ducatelle, E. Ferrante, A. Förster, J. Guzzi, V. Longchamp, S. Magnenat, J. Martinez Gonzales, N. Mathews, M. Montes de Oca, R. O’Grady, C. Pinciroli, G. Pini, P. Rétoznaz, J. Roberts, V. Sperati, T. Stirling, A. Stranieri, T. Stützle, V. Trianni, E. Tuci, A. E. Turgut, and F. Vaussard. Swarmanoid: a novel concept for the study of heterogeneous robotic swarms. *IEEE J. Robot. Autom.*, page in press, 2012.
- [7] C. Jones and M. Mataric. Adaptive division of labor in large-scale minimalist multi-robot systems. In *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 2, pages 1969 – 1974, Oct. 2003.
- [8] H. Kurokawa, E. Yoshida, K. Tomita, A. Kamimura, S. Murata, and S. Kokaaji. Self-reconfigurable m-tran structures and walker generation. *Robotics and Autonomous Systems*, 54(2):142–149, 2006.
- [9] J. Lehman and K. Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation*, 19(2):189–223, 2011.
- [10] S. Mascaro, K. Korb, and A. Nicholson. An alife investigation on the origins of dimorphic parental investments. In H. A. Abbass, T. Bossomaier, and J. Wiles, editors, *Advances in Natural Computation, Proceedings of the Australian Conference on Artificial Life (ACAL 2005)*, volume 3, pages 171–185, 2005.
- [11] F. Menczer and R. Belew. Latent energy environments. In *Santa Fe Institute Studies In The Sciences Of Complexity-Proceedings Volume-*, volume 26, pages 191–210, 1996.
- [12] F. Menczer, W. Willuhn, and R. Belew. An endogenous fitness paradigm for adaptive information agents. In *CIKM Workshop on Intelligent Information Agents*. Citeseer, 1994.
- [13] J.-B. Mouret and S. Doncieux. Encouraging behavioral diversity in evolutionary robotics: an empirical study. *Evolutionary Computation*, 20(1):91–133, 2012.
- [14] Q. Nguyen, X. Nguyen, M. O’Neill, and A. Agapitos. An investigation of fitness sharing with semantic and syntactic distance metrics. In A. Moraglio, S. Silva, K. Krawiec, P. Machado, and C. Cotta, editors, *Genetic Programming*, volume 7244 of *Lecture Notes in Computer Science*, pages 109–120. Springer Berlin Heidelberg, 2012.
- [15] S. Nolfi and D. Floreano. *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. MIT Press, Cambridge, MA, 2000.
- [16] S. Rasmussen, C. Knudsen, R. Feldberg, and M. Hindsholm. The coreworld: Emergence and evolution of cooperative structures in a computational chemistry. *Physica D: Nonlinear Phenomena*, 42(1):111–134, 1990.
- [17] T. S. Ray. Is it alive or is it GA. In R. K. Belew and L. B. Booker, editors, *Fourth International Conference on Genetic Algorithms*, pages 527–534, 1991.
- [18] M. Scheutz and P. Schermerhorn. Predicting population dynamics and evolutionary trajectories based on performance evaluations in alife simulations. In H.-G. Beyer and U.-M. O’Reilly, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2005)*, pages 35–42. ACM, ACM, 2005.
- [19] C. Schwarzer, C. Höslér, and N. Michiels. Artificial sexuality and reproduction of robot organisms. In P. Levi and S. Kernbach, editors, *Symbiotic Multi-Robot Organisms: Reliability, Adaptability, Evolution*, pages 384–403. Springer-Verlag, Berlin-Heidelberg-New York, May 2010.
- [20] F. Tang and L. Parker. A complete methodology for generating multi-robot task solutions using asymptre-d and market-based task allocation. In *Robotics and*

- Automation, 2007 IEEE International Conference on*, pages 3351–3358, Apr. 2007.
- [21] V. Trianni. *Evolutionary Swarm Robotics – Evolving Self-Organising Behaviours in Groups of Autonomous Robots*, volume 108 of *Studies in Computational Intelligence*. Springer-Verlag, Berlin-Heidelberg-New York, 2008.
 - [22] J. Ventrella. Genepool: Exploring the interaction between natural selection and sexual selection. *Artificial Life Models in Software*, pages 81–96, 2005.
 - [23] W. Walsh and M. Wellman. A market protocol for decentralized task allocation. In *Multi Agent Systems, 1998. Proceedings. International Conference on*, pages 325–332, July 1998.
 - [24] R. A. Watson, S. G. Ficici, and J. B. Pollack. Embodied evolution: Distributing an evolutionary algorithm in a population of robots. *Robotics and Autonomous Systems*, 39(1):1–18, Apr. 2002.
 - [25] S. Wischmann, K. Stamm, and F. Wörgötter. Embodied evolution and learning: The neglected timing of maturation. In F. Almeida e Costa, editor, *Advances in Artificial Life: 9th European Conference on Artificial Life*, volume 4648 of *Lecture Notes in Artificial Intelligence*, pages 284–293. Springer-Verlag, Lisbon, Portugal, September 10–14 2007.