Understanding Environmental Influence in an Open-Ended Evolutionary Algorithm

Andreas Steyven[**], Emma Hart, and Ben Paechter

School of Computing, Edinburgh Napier University,
10 Colinton Road, Edinburgh, Scotland, United Kingdom
{a.steyven,e.hart,b.paechter}@napier.ac.uk

Abstract. It is well known that in open-ended evolution, the nature of the environment plays in key role in directing evolution. However, in Evolutionary Robotics, it is often unclear exactly how parameterisation of a given environment might influence the emergence of particular behaviours. We consider environments in which the total amount of energy is parameterised by availability and value, and use surface plots to explore the relationship between those environment parameters and emergent behaviour using a variant of a well-known distributed evolutionary algorithm (mEDEA). Analysis of the resulting landscape show that it is crucial for a researcher to select appropriate parameterisations in order that the environment provides the right balance between facilitating survival and exerting sufficient pressure for new behaviours to emerge. To the best of our knowledge, this is the first time such an analysis has been undertaken.

Keywords: evolutionary robotics; parameter selection; environment-driven evolution; distributed online adaptation

1 Introduction

Due to technological advances in both hardware and software, the vision of sending swarms of robots into unchartered terrains to monitor and map environments is becoming much closer to being realised. This brings significant new challenges for evolutionary robotics, with the need for completely distributed evolutionary algorithms to evolve controllers that enable robots to survive for long-periods of time. The issue of survival is key if robots are to effectively accomplish any kind of task: user-driven tasks cannot even be achieved if the integrity of the swarm is compromised through lack of ability to survive.

A number of recent algorithms tackle this issue, notably mEDEA [1] and its variations e.g. mEDEA$_{rf}$ [7] and MONEE [6,4]. However, the emerging behaviours arising from the interactions of an open-ended evolutionary algorithm with its environment are not well understood, perhaps in part due to the time-consuming experimentation that needs to be done to conduct sweeps of the parameters that define the environment. It is common in optimisation to explore

[**] This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version was published in PPSN XIV 2016, http://dx.doi.org/10.1007/978-3-319-45823-6_86.

the relationship between algorithmic parameters and fitness. However, evolutionary robotics adds an additional dimension in that it is not only the algorithms parameters that change but also the *environmental* parameters.

Given that it is the environment that provides the pressure to adapt in a purely open-ended scenario, it is crucial to gain some understanding of these landscapes. Particularly in simulation, it is easy to arbitrarily select environmental parameters such as the number of available energy sources or their corresponding energy-values. However, arbitrary choices can inadvertently create landscapes which have a major influence on the evolution of behaviour. For example, assume a researcher wishes to investigate whether individual learning speeds up environment-driven evolution: if an environment is created that has too much energy available then it is unlikely to exert sufficient pressure for individual learning to be beneficial or even emerge. Quantifying 'too much' (or 'too little') is of course difficult. In order to address this, we conduct an analysis of an open-ended evolution algorithm operating in a variable environment. To the best of our knowledge, this is the first time this has been attempted.

Using an open-ended evolutionary algorithm, mEDEA$_{rf}$ [7], we consider evolved behaviours in environments in which the total energy available is parameterised by two variables that determine the availability and value of energy pellets within in the environment. Using a 3-dimensional visualisation of the energy landscape for mEDEA$_{rf}$ we show:

- the energy landscape contains three distinct regions: energy-poor, energy-neutral and energy-rich, as well as a 'dead-zone' in which robots cannot survive
- the energy-rich region is relatively large compared to other regions but is very rugged
- that on the energy-neutral line, distinct behaviours evolve at different places along the line

We propose that the energy-neutral region provides the most obvious settings for conducting experimentation that aims to extend a robots ability to survive or accomplish tasks.

2 Related Work

The completely distributed evolutionary algorithm for open-ended evolution *mEDEA* was first proposed in [1]. It was tested using a scenario in which environmental pressure forces robots to compete for limited resources in order to gain energy. The algorithm was demonstrated to be both efficient with regard to providing distributed evolutionary adaptation in unknown environments, and robust to unpredicted changes in the environment. The basic algorithm has been extended in a number of ways.

Haasdijk *et al* [6] extended mEDEA so that in addition to surviving and operating reliably in an environment, a robot could also perform user-defined tasks. Their new framework MONEE (Multi-Objective aNd open-Ended Evolution algorithm) showed initially that task-driven behaviour can be promoted without compromising environmental adaptation. More recently, they investigated the trade-off between the survival and task-accomplishment that evolution must establish when the task is detrimental to survival, finding that task-based selection exerts a higher pressure than the environment. Fernandez *et al* [3] study the impact of adding explicit selection methods to the mEDEA algorithm in a task-driven scenario. They evaluate four selection methods that induce different intensities of selection pressure, using tasks that include obstacle avoidance and foraging, finding that higher selection pressure results in improved performances, especially in more challenging tasks. Hart [7] also extended mEDEA by including selection based on a fitness value that was calculated relative to those robots in the immediate vicinity, thus maintaining the decentralised nature of the algorithm, and additionally using this relative fitness value to control the frequency and range of broadcasting. Parameter tuning of *algorithmic* parameters to optimise algorithmic task-performance was investigated by [5]. However, to the best of our knowledge, no methodical investigation of *environment* parameter settings has been conducted: researchers tend to select arbitrary values or simply use those defined in previous papers.

3 Algorithm Description

Evolution of robot controllers is performed by the mEDEA$_{rf}$, first introduced in [7]. The algorithm is an extension of the original mEDEA algorithm of Bredeche *et al* [1] with the addition of an explicit fitness measure. This influences the spread of genomes through the population in order to increase survivability, thus ensuring the integrity of the swarm.

mEDEA$_{rf}$ utilises an agent driven by a control architecture whose parameters are defined by the currently active genome. The genome defines the weights of an Elman recurrent neural network (RNN) consisting of 16 sensory inputs, one bias node (feeding into the hidden layer) and 2 motor outputs (translational and rotational speeds). 8 ray-sensors are distributed around the robot's body. They detect the proximity to the nearest object and its type. The RNN has 1 hidden layer with 16 nodes, thus 322 weights are defined by the genome. This setup is adapted from [1]. An overview of the algorithm is given in Algorithm 1 and reader is referred to [7] for more detail. In brief, for a fixed period, robots move according to their control algorithm, broadcasting their genome that is received and stored by any robot within range. At the end of this period, a robot uses roulette-wheel selection to choose a genome from its list of collected genomes according to a relative fitness value, and applies a variation operator. This takes the form of a Gaussian random mutation operator, inspired from Evolution Strategies. Robots that have not collected any genomes temporarily become inactive, thus reducing the population size.

Each robot estimates its fitness in terms of its ability to survive based on the balance between energy lost and energy gained, delta Energy (δ_E): this term is initialised to 0 at $t = 0$ (when the current genome was activated) and is decreased by 1 at each time-step, and increased by E_{token} if it crosses an energy token. Given δ_E, a robot calculates a fitness value which is relative to those robots in a range r according to equation 1, where f_i' is the relative fitness of robot i at time t, $mean_{sub_i}$ is the mean δ_E of the robots within the subpopulation defined by all robots in range r of robot i, and sd_{sub_i} is the standard deviation of the δ_E of the subpopulation.

$$f_i'(t) = \frac{\delta_i(t) - mean_{sub_i}(t)}{sd_{sub_i}(t)} \tag{1}$$

Note that evolution is asynchronous, in keeping with the paradigm of a distributed algorithm without central control. If a robot runs out of energy and has an empty genome list, it remains stationary until it receives a new genome from a passing robot at which point it starts a new lifetime. Thus at any time-step, each robot potentially has a different ‘age’.

```
genome.randomInitialise();
agent.load(genome);
while forever do
    if genome.isNotEmpty() then
        while lifetime < maxLifetime and energy > 0 do
            agent.move();
            if neighbourhood.isNotEmpty() then
                rf = agent.calculateRelativeFitness(neighbourhood);  // eq. 1
                broadcast(genome,rf);
            end
        end
        genome.empty();
    end
    if genomeList.size() > 0 then
        genome = applyVariation(select_{rhoulette-wheel}(genomeList));
        agent.load(genome);
        genomeList.empty();
    end
end
```

Algorithm 1: Pseudo code of our adapted version of the mEDEA algorithm based on vanilla mEDEA by Bredeche *et al.* [1]

4 Method

All experiments are conducted in simulation using Roborobo! by Bredeche et al. from [2]. A static environment is created, using an arena previously described in [3,6,4]. The robot cannot pass through the outer and inner walls, however, it is possible to broadcast through an obstacle. Energy *tokens* are randomly scattered in the environment. If a robot moves over a token, its energy is increased by an amount E_{token}. The energy token disappears when consumed and reappears after a fixed amount of time later at a different random location. Fixed parameters describing the simulation are given in table 1.

Energy is consumed in three ways. There is a fixed cost to 'living' of 0.5 units per timestep, regardless of whether the robot moves or not. A robot moving consumes an amount of energy E_m that is related to its rotational speed v_{rot}, translational speed v_{trans}, and their respective maximum values $v_{\text{rot}_{\text{MAX}}}$ and $v_{\text{trans}_{\text{MAX}}}$, and is given by

$$E_m = (v_{\text{rot}}/v_{\text{rot}_{\text{MAX}}} + v_{\text{trans}}/v_{\text{trans}_{\text{MAX}}})/4 \tag{2}$$

Finally, a robot consumes energy when communicating. This is an important factor in the real-word but one that it is often overlooked in simulation models. The model used is exactly as described in [10], with an energy cost of $E_{RX} = 0.082$ units for receiving and a cost of $E_{TX}(r) = 0.075$ units for transmitting.

The goal of the experiments is to understand the energy landscape in terms of the median δEnergy of a robot in the population as a function of the two environmental parameters: *count*, the number of energy tokens available, and *value*, the energy value of each token. Table 1 shows the ranges of values considered for each parameter. Parameters are set before the beginning of the experiment and remain fixed throughout. Each experiments was repeated for 5 independent runs. This number is rather low for a noisy application of this type but was chosen to speed up computation due to the high number of experiments that had to be run in total.

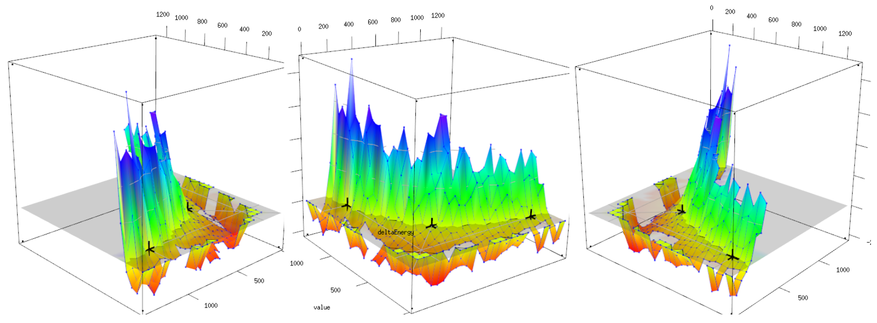
Table 1: Simulation and Experimental Parameters for all experiments

Simulation parameters	
Arena size	1024 pixel by 1024 pixel
Max. robot lifetime	2500 iterations
Token re-spawn time	500 iterations
Sensor range	196 pixel
Variable Parameters	
Number of robots	50, 75, 100
Number of tokens (*count*)	0 - 1300 (in steps of 50)
Energy value per token (*value*)	0 - 1400 (in steps of 50)
Experimental parameters	
Number of runs	5
Maximum iterations	375000 (= 150x2500)
Start energy	500
Maximum range $r_{\max}$	128

Data is gathered from the robots every 2500 iterations. Recall from section 3 that each robot chooses a new genome once it has depleted all its energy or reached the maximum lifetime, leading to asynchronous generation changes throughout the population. Hence, the data gathered at each interval represents a snapshot across robots of multiple ages and therefore does not necessarily capture the peak performance of each robot (i.e. it may include very 'young' robots). However, given that the goal of the experiment is to understand the interplay of the specific algorithm and environment under consideration, this is not a relevant factor.

5 Analysis

Figure 1 shows three rotated 3-dimensional plots of surface obtained using 100 robots after 375,000 iterations. The x and y axes represent the *count* and *value* variables, while the z axis represent the median δ_E of the robot population over the last 2500 iterations. The grey plane marks a value for δ_E of zero, at which point robots have an energy balance of zero, i.e. the same amount of energy as they started the experiment with. Three broad regions are noticeable: a large region in which the robots have positive δ_E (green and blue value above the grey plane), a region lying on the plane itself, and finally a region below the plane in which robots are spending more energy than they are collecting, i.e. $\delta_E < 0$. In order to explore this in more detail, a 2-dimensional top-down projection is shown in figure 2 obtained from populations of 50, 75 and 100 robots, and is discussed in detail below.



(a) rotated 90° right (b) centred (c) rotated 90° left

Fig. 1: View on the resulting surface from different angles. The figure was created by plotting the median δ_E of the last 2500 iterations of the experiment. The grey plane marks a value for δ_E of zero, at which point robots in an experiment have an energy balance of zero. In other words, the same amount of energy as they started the experiment with. A 3D model can be found at [9]

5.1 Different performance regions

Figure 2 shows clearly that the landscape is defined by four different regions:

A) Dead Zone: In this region, the environment does not provide enough energy for the algorithm to evolve controller that can survive a full run. Low values for both parameters, *count* and *value* result in the extinction of the whole robot population within a few generations. The random genomes that the controllers are initialised with generally result in a random spinning behaviour, rather than movement. This random behaviour, combined with the lack of energy tokens in the immediate vicinity in which the robot is born, mean that robots cannot survive given its inability to move.

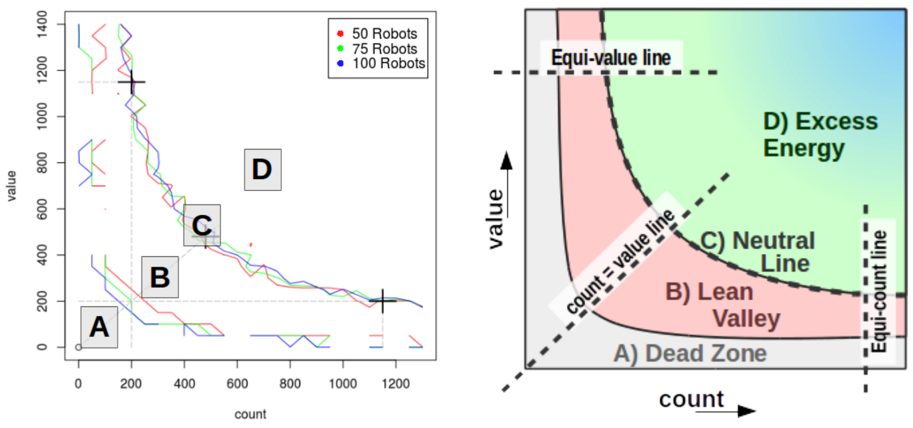


Fig. 2: Overview of landscape, as plot of the real data on the left and as a cartoon version on the right. 4 different regions are shown: A) Dead Zone, B) Lean Valley, C) Neutral Line, D) Excess Energy.

B) Lean Valley (negative δ_E): This region starts at the edge of the dead zone that marks the point where there is just enough energy available that some robots survive until the end of the experiment, i.e. it marks the point where a robot has spent all its initial energy and started picking up tokens from the environment. Moving down towards the bottom of the valley, an increasing number of robots survive as there is more energy in environment, with the corollary that each robot has less total energy — the energy available is shared between more robots. The bottom of the valley marks the minimum δ_E that still enables survival. Moving upwards out of the valley on the other side, robots gradually get better in both harvesting energy from the environment and managing their residual energy as a result of evolving better strategies. For example, good strategies optimise movement, or avoid moving towards tokens in which there are other robots close by.

C) Neutral Line (δ_E=0): This line marks the points in the environment where the environment provides exactly enough energy to enable a robot to maintain an energy balance of zero, i.e. the costs of moving and communicating are just balanced by energy harvested.

D) Excess Energy ($\delta_E > 0$): In the final region, in which both *cost* and *value* are high, robots are able to locate more energy in the environment than is required to maintain their initial energy E_0, either due to the abundance of pucks or the high energy value of pucks.

5.2 Environmental Influence on Behaviour

In order to properly understand the evolved behaviours that lead to the landscapes just described, a more detailed analysis is required. Figure 3 examines pairings of $(count, value)$ along the three dashed lines in 2, i.e. equivalent-*value* (**a-b**), equivalent-*count* (**c-d**) and the diagonal in which $count = value$ line (**e-f**). The figure shows boxplots of the δ_E values at specific pairings of $(count, value)$ and the ratio of genome broadcasts made to unique genomes received over a lifetime. The latter quantity leads to insights into behaviour as it relates to the number of *unique* robots encountered by an individual robot: a robot will broadcast indiscriminately to any robot in its range but will only collect unique genomes. At the equivalent-count and equivalent-value lines, we fix the parameter *count* and *value* respectively, and successively increase the other parameter in steps of 50.

5 points are shown. The first point on a) corresponds to a total energy E_{tot} that is the same as the first points on graphs (c) and (e) below it etc.[1]. For a specific value of E_{tot}, then is clear that high *value* combined with low *count* leads to robots that have increased δ_E when compared to robots with high *count* but low *value* (graph (a) compared to graph (e)). Robots must therefore evolve behaviours that enable them to seek out the rare but high-value pucks. These robots also have high broadcast:genome ratios, suggesting the robots are frequently coming into contact with the *same* robots. A possible explanation lies in the fact that the robots appear travel in small groups, thus broadcasting continually to the same robots; the rare occurrence of pucks leads to many robots having to travel towards the same regions of the space. On the other hand, a high *count* leads to robots that receive more unique genomes than in the high *value* case: this is suggestive of a more random movement pattern that enables each robot to encounter many unique robots during its lifetime. In this case there is low selection pressure to evolve focused movement due to the abundance of pucks.

5.3 Behaviours in the neutral region

We propose that the energy neutral region is of greatest interest for researchers wishing to conduct research moving beyond genetic evolution of survival, for example using individual or social learning [8] or task-driven research [4]. In this region, on the one hand, robots are able to survive, while on the other, the environment does not *over*-provide, thus ensuring that there is scope for robots to learn novel behaviours. We further investigate three specific points within this region there is approximately the *same* amount of energy available in the environment (table 2). The table shows the median age increases with increasing *count* — it is easier to maintain sufficient energy to survive as availability increases. The lower median observed at low *count* reflects the fact that many robots do not survive long. The time to find a new unique genome (age:genome) is shortest at high *count*, reflecting frequent encounters with novel robots. Broadcast:genomes is highest at low *count* as observed in the previous section. All three configurations lead to the same energy balance of 0, but diverse behaviours result in the gain in energy being offset by movement and broadcasting in each case.

[1] while this is exactly true for the first and third rows, in the middle row which represents equal count/value it is necessary to approximate

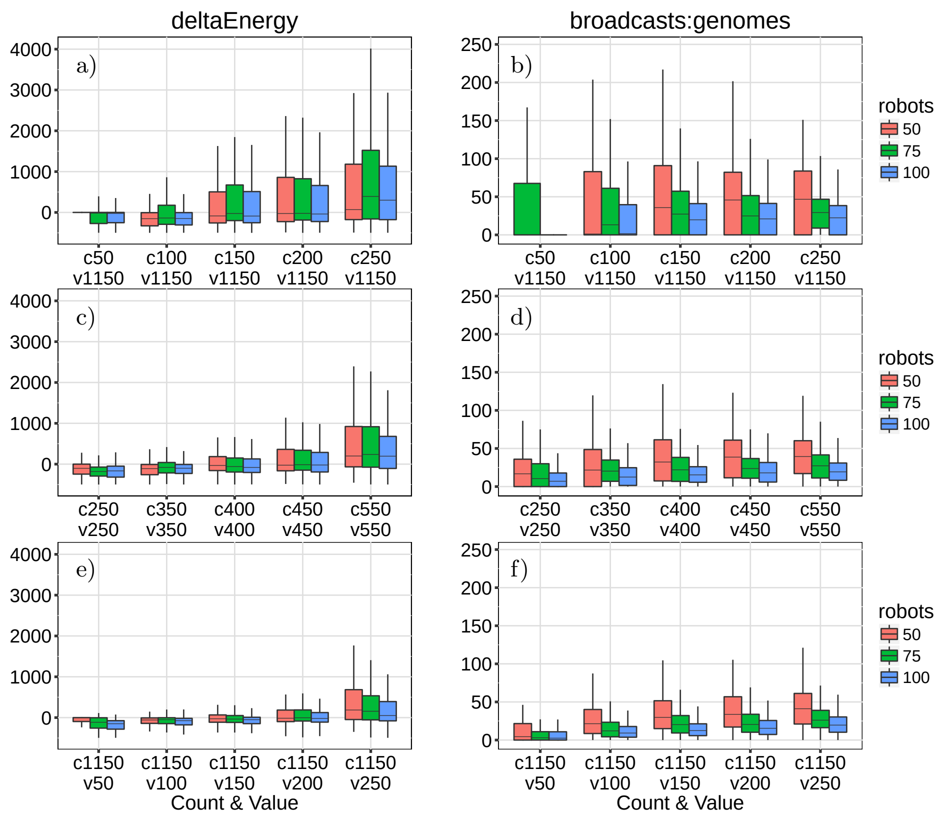


Fig. 3: Cuts through different parts of the landscape. Points towards different behaviours in terms of exploration. a-b) $value$ = 1150, vary $count$; c-d) $count$ = $value$; e-f) $count$ =1150, vary $value$.

6 Conclusion

We have presented the first analysis of the fitness landscape (as a function of environmental parameter) that results from running an open-ended evolutionary algorithm (mEDEA$_r f$) in an environment that is parameterised by two values that control the distribution of energy in the environment. Adjusting the availability and value of energy pucks results in the evolution of a range of different behaviours. Rather than arbitrarily selecting parameters in which to study evolution, we suggest that it is vital to understand *how* these choices will direct evolution, by changing the selection pressure exerted by the environment.

Three distinct regions are observed in which the final energy balance can be negative, neutral, or positive. A fourth region is found in which robots cannot survive. We propose that the energy neutral region is a good region in which to undertake experiments. It provides an environment in which robots are able to survive, enabling experimentation, while at the same time, will reward new behaviours which are able to more efficiently harness energy from the environ-

Table 2: Results obtained at three configurations within the neutral region

		Robots								
		50			75			100		
Count	Value	Age	$\frac{\text{Age}}{\text{Genome}}$	$\frac{\text{Brodcasts}}{\text{Genome}}$	Age	$\frac{\text{Age}}{\text{Genome}}$	$\frac{\text{Brodcasts}}{\text{Genome}}$	Age	$\frac{\text{Age}}{\text{Genome}}$	$\frac{\text{Brodcasts}}{\text{Genome}}$
200	1150	*770*	*107.94*	*46.61*	*767.5*	*63.86*	*28.99*	*667*	*49.97*	*21.18*
500	500	*1026.5*	*86.12*	*39.11*	*1038.5*	*52.39*	*25.93*	*928.5*	*41.26*	*19.67*
1150	200	*1173.5*	*79.83*	*33.43*	*1093*	*47.39*	*21.95*	*1059*	*36.52*	*15.49*

ment. It is clear that the environment plays a key role in influencing what kind of behaviours emerge, in that it is not the total amount of energy available that matters but also the manner in which it is spread. Future work should be aimed at understanding the landscape in more detail, and in particular, explaining the ruggedness of some regions.

References

1. Bredeche, N., Montanier, J.M.: Environment-driven embodied evolution in a population of autonomous agents. In: Schaefer, R., Cotta, C., Koodziej, J., Rudolph, G. (eds.) PPSN XI. vol. 6239, pp. 290–299. Springer Berlin Heidelberg (2010)
2. Bredeche, N., Montanier, J.M., Weel, B., Haasdijk, E.: Roborobo! a fast robot simulator for swarm and collective robotics. CoRR abs/1304.2 (4 2013)
3. Fernández Pérez, I., Boumaza, A., Charpillet, F.: Comparison of selection methods in on-line distributed evolutionary robotics. In: ALife'14. pp. 282–289. MIT Press (2014)
4. Haasdijk, E.: Combining conflicting environmental and task requirements in evolutionary robotics. In: 2015 IEEE 9th International Conference on Self-Adaptive and Self-Organizing Systems. pp. 131–137. IEEE (9 2015)
5. Haasdijk, E., Smit, S.K., Eiben, A.E.: Exploratory analysis of an on-line evolutionary algorithm in simulated robots. Evolutionary Intelligence 5(4), 213–230 (2012)
6. Haasdijk, E., Weel, B., Eiben, A.E.: Right on the MONEE. In: Blum, C. (ed.) Proceedings of GECCO '13. pp. 207–214. ACM Press (2013)
7. Hart, E., Steyven, A., Paechter, B.: Improving survivability in environment-driven distributed evolutionary algorithms through explicit relative fitness and fitness proportionate communication. In: Silva, S. (ed.) Proceedings of GECCO '15. pp. 169–176. ACM Press (2015)
8. Heinerman, J., Rango, M., Eiben, A.E.: Evolution, individual learning, and social learning in a swarm of real robots. In: 2015 IEEE Symposium Series on Computational Intelligence. pp. 1055–1062. IEEE (2015)
9. Steyven, A.: Interactive 3D model of mEDEA_rf parameter sweep fitness landscape (2016), `http://research.steyven.de/conf/ppsn2016/`
10. Steyven, A., Hart, E., Paechter, B.: The cost of communication. In: Silva, S. (ed.) GECCO Companion '15. pp. 1239–1240. ACM Press (2015)