Improving Survivability in Environment-driven Distributed Evolutionary Algorithms through Explicit Relative Fitness and Fitness Proportionate Communication

Emma Hart
School of Computing
Edinburgh Napier University
Edinburgh, Scotland, UK
e.hart@napier.ac.uk

Andreas Steyven
School of Computing
Edinburgh Napier University
Edinburgh, Scotland, UK
a.steyven@napier.ac.uk

Ben Paechter
School of Computing
Edinburgh Napier University
Edinburgh, Scotland, UK
b.paechter@napier.ac.uk

ABSTRACT

Ensuring the integrity of a robot swarm in terms of maintaining a stable population of functioning robots over long periods of time is a mandatory prerequisite for building more complex systems that achieve user-defined tasks. *mEDEA* is an environment-driven evolutionary algorithm that provides promising results using an implicit fitness function combined with a random genome selection operator. Motivated by the need to sustain a large population with sufficient spare energy to carry out user-defined tasks in the future, we develop an explicit fitness metric providing a measure of fitness that is relative to surrounding robots and examine two methods by which it can influence spread of genomes. Experimental results in simulation find that use of the fitness-function provides significant improvements over the original algorithm; in particular, a method that influences the frequency and range of broadcasting when combined with random selection has the potential to conserve energy whilst maintaining performance, a critical factor for physical robots.

Categories and Subject Descriptors

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

Keywords

Evolutionary Robotics; Environment-driven; On-line Evolution

1. INTRODUCTION

Recent advances in technology both in hardware and software [8] have fuelled visions of swarms of robots being sent to remote or hazardous environments in which they will need to survive over long periods of time. Environments will be unknown and potentially dynamic, requiring autonomous adaptation by the swarm. This has led to a number of recent efforts to study evolution within a swarm as a mechanism for driving adaptation, as opposed to a mechanism for optimising an explicit fitness function, as is common in much work within evolutionary robotics. Montanier [12] notes that this step is in fact a pre-requisite to studying any kind of user-driven task behaviours within a robotic swarm in an open-environment, as the former cannot be achieved if the integrity of the swarm is compromised.

This type of evolution is often referred to as *environment-driven evolution* [2]. Typical approaches such as [14] remove the need for any central control, resulting in algorithms that perform distributed and online evolution. An additional feature of environment-driven algorithms is that no explicit fitness function is defined: instead, mate selection and reproduction depend on selection pressure provided only by the environment yet need to lead to stable populations. A recent example of this is the mEDEA algorithm (minimal Environment driven Distributed Evolutionary Adaptation) [3, 4, 5]. mEDEA relies on an implicit fitness function that results from two potentially conflicting motivations for a robot: an *extrinsic* motivation to cope with environmental constraints in order to maximise survival ability and an *internal* motivation to spread its genomes across the population in order to survive. A complex trade-off exists in which behaviours that maximise mating opportunities might negatively impact survival efficiency, e.g. failing to maintain stable energy levels; as a result, mEDEA (or an environment-driven EA) must find some equilibrium between the two states.

The original version of mEDEA exploited a simple strategy in which a robot continuously broadcast its genome — this can be received and stored by any robot currently within communication range. At the end of a generation, each robot makes a *random* selection from its set of stored genomes, applies a mutation operator, and then replaces its current genome, exactly as in a (1,1) Evolution Strategy [1]. Although there is no selection pressure on an individual basis, from a global perspective, the most widely spread genomes will be selected more often on average. While this achieved success in evolving stable populations in an open-ended environment, it is of interest to attempt to improve both the size of the swarms maintained and their net energy levels, in order that complex user-defined tasks can be added in future. It is reasonable to assume that spare energy, over and above that required to survive can be exploited to achieve complex tasks, whilst a large swarm offers more potential in terms of the tasks that might be accomplished.

Within mEDEA, the evolutionary mechanism differs from natural evolution that also drives adaptation in a number of respects. Firstly, there is no form or crossover — variation

http://dx.doi.org/10.1145/2739480.2754688

is provided only by a mutation operator — and hence the emphasis is on the spreading of *genomes* rather than *genes* as proposed by the Selfish Gene paradigm of Dawkins [7]. Secondly, as all robots broadcast their genome continuously and with fixed radius, each robot has an equal opportunity to pass on its genome, regardless of its quality. This latter point is clearly not true in many natural systems. In nature, an individual's chance of reproduction is related to its fitness relative to other individuals in its vicinity. Additionally, individuals mate selectively, choosing partners based on some estimation of their quality. Some species broadcast their quality through visual or behavioural displays: a peacock displays its tail feathers, a bird of paradise 'dances'. Fitter individuals can attract the attention of a greater number of potential mates.

Inspired by nature, we investigate the effect of introducing a relative fitness measure into mEDEA. A robot makes an estimate of its fitness to survive relative to those within its broadcast range, thereby maintaining the distributed nature of the algorithm. The value can be used in two ways:

- An individual robot can make an informed rather than random selection from the genomes it has received according to the relative fitness value
- The relative fitness value can be used to influence the broadcasting behaviour of a robot to provide a bias towards the spread of good genomes

The latter point changes the nature of the reproductive strategy used in mEDEA from a 'promiscuous' one in which there is indiscriminate broadcasting of a genome, to one in which the *spread* of a genome (and therefore the probability of it being collected) is dependent on its quality. Two novel methods for influencing broadcasting are introduced: the first causes the robot to adapt the probability with which it broadcasts based on its fitness. The second causes the robot to adapt the range over which it broadcasts its genome. As in the original version of mEDEA, robots still make a random selection from collected genomes; however, due to the biased broadcasting methods, on average good quality genomes are more likely to be collected than poorer ones.

Results for all approaches are compared to the original mEDEA algorithm. Experiments show that both methods perform equally well compared to the original algorithm. Note however that broadcasting in the physical world is an energy consuming operation; methods that reduce this energy in order to save battery by either reducing the range or frequency of broadcasting are likely to be of considerable benefit.

2. RELATED WORK

mEDEA was first proposed in [3]. The system was tested under two scenarios: the first evaluated mEDEA in an environment providing limited pressure in which energy is ignored and an agent survives as long as it collects at least one genome. In the second, environmental pressure is introduced by forcing 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. Furthermore, given its lightweight nature, it was predicted to be suitable for hardware and software setups that have limited computation.

Haasdijk *et al* [10] extended mEDEA so that in addition to surviving and operating reliably in an environment, a robot could also perform user-defined tasks. Survivor selection is driven by the environment, whereas parent selection is driven by task performance. Their new framework MONEE (Multi-Objective aNd open-Ended Evolution algorithm) showed that task-driven behaviour can be promoted without compromising environmental adaptation. Robots accumulate credit for accomplishing particular tasks — this credit value is transmitted along with a genome, and is utilised in a fitness function to select parents, replacing the random selection seen in mEDEA. The basic mEDEA setup is also altered by adding an 'egg' phase that occurs at periodic intervals: during this phase, the stationary robot collects genomes from passing robots — no genomes are collected whilst a robot is moving.

Perez *et al* [13] 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.

Watson *et al.* [15] proposed an completely decentralised algorithm for embodied evolution (EE). In their Probabilistic Gene Transfer Algorithm (PGTA) robots exchange randomly selected genes through short range communication. The algorithm differs from other approaches in that is doesn't have a dedicated variation and replacement steps. Each robot holds a single genome of which only individual genes are replaced at runtime. Transmission frequency and gene acceptance are based on the explicit fitness value of the respective robot, which reflects its performance on a task.

Another strand of work of work worth mentioning is that of Stanley et al. [11] who introduced the notion of *Novelty Search* that promote the discovery of solutions that differ from the ones already evolved. Solutions are selected for novelty rather than objective fitness hence do not require an explicit fitness function; however, the algorithm requires global knowledge to calculate novelty hence cannot be used as described in a purely distributed evolutionary algorithm.

Our proposal provides a novel contribution to the line of work on mEDEA that started in [3] in adding an explicit fitness function that is defined in relation to survival ability rather than task performance, and using this to providing two methods of influencing the rate of spread of good genomes through the population. By providing a more robust mechanism for maintaining swarm integrity than previous work, we pave the way towards future work in which it is possible to add user-defined tasks to a swarm operating in an unknown environment, thereby increasing its utility.

3. ALGORITHM DESCRIPTION

mEDEA 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 43 sensory inputs 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, the presence of walls and other robots, whether it belongs to the same group and the relative orientation between the two robots. An energy level input feeds the current level into the network. A distance and angle sensor give the direction to the nearest energy token. The RNN has 1 hidden layer with 8 nodes, thus 434 weights are defined by the genome. This setup is adapted from [3]. The original mEDEA algorithm is defined in algorithm 1 — only the single step *broadcast()* is modified in this paper, therefore the reader is referred to [3] for a detailed description of the concepts that underpin its design and the specification of each of the other methods.

```
genome.randomInitialise();
while forever do
    if genome.isNotEmpty() then
        agent.load(genome);
    end
    for iteration = 0 to lifetime do
        if agent.isAlive() and genome.isNotEmpty()
        then
            agent.move();
            broadcast(genome);
        end
    end
    genome.empty();
    if genomeList.size() > 0 then
        genome =
        applyVariation($select_{random}$(genomeList));
        if agent.isAlive() == false then
            agent.setAliveState(true);
        end
    end
    else
        agent.setAliveState(false);
    end
    genomeList.empty();
end
```

Algorithm 1: Pseudo code of the original mEDEA algorithm by Bredeche *et al.* [4]

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 selects a random genome from its list of collected genomes and applies a variation operator. This takes the form of a Gaussian random mutation operator, inspired from Evolution Strategies [1] which can be easily tuned through a σ parameter. Robots that have not collected any genomes become inactive, thus reducing the population size.

At the start of each generation, a robot is initialised with an energy E_0. Every time step, the energy value is decreased by one unit. Energy *tokens* are scattered in the environment. If a robot moves over a token, its energy is increased by an amount E_{token}. An robot with energy=0 remains stationary for the remainder of the generation.

The next section describes the proposed modifications to mEDEA algorithm that we dub $mEDEA_{rf}$ — mEDEA with relative fitness.

3.1 mEDEA$_{rf}$

A robot calculates an estimate of its own fitness to survive based on the balance between energy lost and energy gained, δ_E: this term is initialised to 0 at $t = 0$ 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 its fitness relative to the 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 f'_i is defined in relation to the ability of the robot to *survive* in the environment; it records the net energy of a robot, accounting for energy expended and energy gained by locating tokens. It differs from the explicit task-driven fitness functions investigated by [13] that were concerned only with task-driven selection, i.e optimising performance on defined tasks; although one task investigated was a foraging task that involved collecting tokens in a similar manner to the one used here, the tokens did not influence robot survival in that they did not contribute to the energy of the robot and therefore its ability to stay alive.

The new explicit fitness function can be exploited in two ways: it can either be transmitted with a genome and used by an individual within a selection function or it can be used to influence the rate at which a genome is broadcast, thereby indirectly affecting its chances of being selected for reproduction. The two approaches are described below.

3.1.1 Explicit selection mechanisms

The $select_{random}$(*genomeList*) method in mEDEA can easily be replaced with an informed selection method that uses the relative fitness measure to discriminate between genomes. We investigate three well-known selection strategies:
tournament-selection, roulette-wheel selection and an elitist select-best strategy.

3.1.2 Biasing broadcasting of genomes

Alternatively, the spread of genomes can be biased by adapting the *broadcast()* step in algorithm 1. In mEDEA, robots make a random selection from their list of collected genomes at the end of each generation. We propose two new methods, both of which bias the spread of genomes throughout the population in favour of higher quality ones based on a robot's estimation of its fitness f' relative to those in its immediate surroundings.

- *broadcast_radius()* adapts the range at which a robot broadcasts depending on f'

- *broadcast_probability()* broadcast at a fixed range r with the probability depending on f'

Given f'_i, we define the probability of a robot broadcasting using equation 2 that simply describes a function that returns a probability 0 if f'_i is less than d_0, probability=1 if f'_i is greater than $d_{\max}$ standard deviations away from the mean, and linearly interpolated between 0 and 1 otherwise.

$$p_i(t) = \begin{cases} 0 & f'_i(t) <= d_0 \\ \frac{f'_i(t) - d_0}{d_{\max} - d_0} & d_0 \leq f'_i(t) < d_{\max} \\ 1 & f'_i \geq d_{\max} \end{cases} \tag{2}$$

For the *Broadcast_probability()* method, shown in $case1$ in algorithm 2, the probability $p_i(t)$ is used directly to determine whether a robot broadcasts. For *Broadcast_radius()*, the probability $p_i(t)$ is converted to a broadcasting range between 0 and a value $r_{\max}$ according to equation 3 — the higher the relative fitness, the greater the broadcast range.

Note that range increases with the square root of the probability in order to maintain a proportional increase in broadcast *area.*

$$r_i(t) = r_{\max} * \sqrt{p_i(t)} \tag{3}$$

Both methods result in robots that have higher relative fitness broadcasting their genome more than those with lower relative fitness, hence biasing the quality of genomes that a receiving robot collects. At the end of each generation, a random selection of genome is made from those collected as in mEDEA.

```
R ← all robots in simulation;
foreach robot $i$ in $R$ do
    $N_i$ ← getRobotsWithinMaxRadius(R);
    if $N_i > 0$ then
        $f_i'$ ← calculate fitness relative to $N_i$ ;    // eq. 1
        $p_i$ ← convert $f_i'$ to probability;             // eq. 2
    else
        $p_i = 0$
    end
    switch exp do
        // vary probability
        case 1
            if $p_i >$ rand() then
                broadcast($r_{\max}$, currentGenome, $\sigma$);
            end
        end
        // vary broadcast radius
        case 2
            $r_i$ ← adjustRadius($p_i$);                   // eq. 3
            foreach robot $j$ in $N_i$ do
                if distance($i$,$j$) < $r_i$ then
                    broadcast($r_i$, currentGenome, $\sigma$);
                end
            end
        end
    endsw
end
```

Algorithm 2: Pseudo code of the algorithm that is executed at every discrete time step of the simulation.

4. EXPERIMENTS

Three sets of experiments were undertaken, exploring the effects of using the explicit selection mechanism, biasing spread of genomes through altering the broadcasting mechanism, and finally biasing spread *and* using explicit selection.

Explicit selection mechanisms.

The first set of experiments investigates the hypothesis that replacing the random selection method in mEDEA with a selection method that selects based on the relative fitness value will increase both the average δ_E of the population and number of robots alive N_{alive} at the end of the final generation when compared to the original mEDEA algorithm. Three selection methods are investigated: binary tournament, roulette-wheel and an elitist select-best. These experiments are labelled E1 (mEDEA), E1+t, E1+rw, E1+b to denote the different selection methods.

Biased broadcasting of genomes.

Experiments were designed to evaluate the following hypotheses:

1. Biasing the spread of genomes via adapting the *probability* that a robot broadcasts based on its relative fitness will improve the average δ_E of the population and N_{alive} compared to the original mEDEA algorithm.

2. Biasing the spread of genomes via adapting the *range* over which a robot broadcasts based on its relative fitness will improve the average δ_E of the population compared to the original mEDEA algorithm.

Note however that the new methods *broadcast_probability()* and *broadcast_range()* introduce *two* adaptations compared to the original algorithm: (1) the broadcast probability (and therefore range) is variable across the population and (2) the broadcast probability (and therefore range) is determined by *relative fitness.* Thus in order to show that any improvement in average δ_E can be attributed to the effect of introducing the *relative fitness* term rather than simply a random variation, we perform additional control experiments as follows:

Rather than calculating the relative fitness of a robot according to equation 1 using its own $\delta_i(t)$, we simply replace it with x_i — a random number drawn from a normal distribution with mean $\Delta(t)$ and $sd\Delta(t)$, where the Δ terms refer to the mean and standard deviation of the fitness of the *global* population (the global fitness is used simply to ensure that the random value is drawn from an appropriate range). New methods *broadcast_randomProbability()* and *broadcast_randomRange()* then use equations 2 and 3 as previously described.These methods are introduced merely to perform rigorous control experiments: we do not suggest that this method would be used in practice as it requires the calculation of a global parameter, contrary to the distributed nature of the algorithm.

Five different experiments are performed, where E1-E3 are controls and E4 and E5 evaluate the new methods.

- E1 records the mean δ_E of the robot population and the number of active robots at the end of the final generation using only the original version of mEDEA

- E2 records the same metrics as above using *broadcast_randomProbability()*

- E3 records the same metrics as above using *broadcast_randomRange().*

- E4 records the same metrics as above using *broadcast_probability().*

- E5 records the same metrics as above using *broadcast_range().*

4.1 Methodology

All experiments use Roborobo! by Bredeche et al. from [6], as in the original simulations described with mEDEA. Roborobo! is a multi-platform, highly portable, robot simulator for large-scale collective robotics experiments. With respect to other robotic simulators, Roborobo! combines (pseudo-)realistic modelling with fast-paced simulation and thus falls somewhere in-between very realistic frameworks such as Player/Stage [9] that tend to be very slow and agent-based tool such as MASON that are extremely simplified with respect to the environment. It focuses solely on large-scale swarms of robots in a 2D environment and is based on a Khepera/ePuck model and has already been used in more than a dozen published research papers mainly concerned with evolutionary swarm robotics, including environment-driven self-adaptation and distributed evolutionary optimization, as well as online onboard embodied evolution and embodied morphogenesis.

All parameters used in the experiments are given in table 1. Simulation parameters are based on the original papers. Experimental parameters were chosen following limited empirical tuning. The maximum broadcasting range requires sensible selection and should be chosen proportional to the arena size.

Simulation parameters	
Arena size	1024 pixel by 1024 pixel
Number of robots	100
Robot lifetime	1500 iterations
Food regrow time	500 iterations
Sensor range	32 pixel
Chromosome length	434
Experimental parameters	
Number of runs	30
Maximum generations	500
Number of energy tokens	800
Energy value of token	100
Start energy	1200
Maximum range $r_{\max}$	64
d_0	0
$d_{\max}$	2

Table 1: Simulation and Experimental Parameters for all experiments

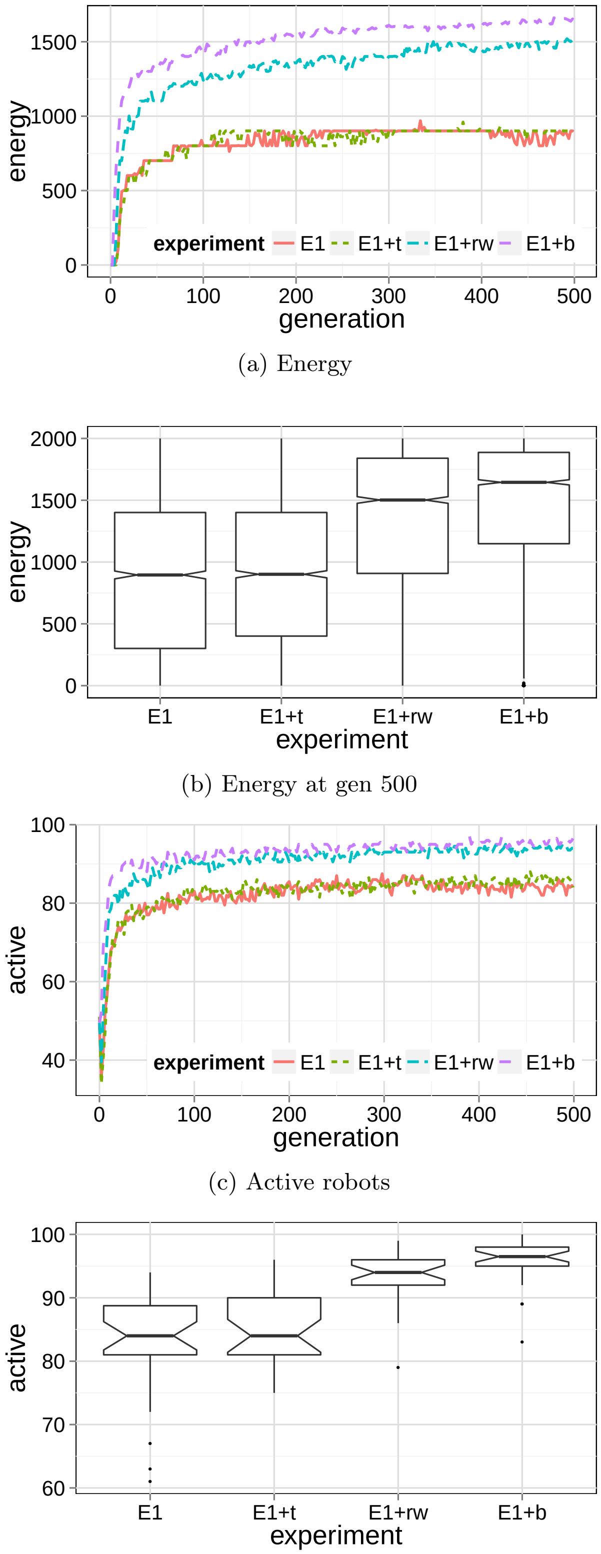
5. RESULTS AND ANALYSIS

Explicit selection by individuals.

Results from the experiments E1, E1+t, E1+rw, E1+b in which the $select_{random}(genome\_list)$ method in algorithm 1 is replaced with a selection method are shown in figure 1, which compares the median[1] energy and agents alive over 30 repeated runs for each of the four experiments listed. Adding an explicit selection method based on a relative fitness value relating to the ability of another robot to survive over the generation has significant effect in the case of *roulette wheel* and *best* selection when compared to mEDEA. Both of these methods exert high selection pressure. In contrast, the low-pressure *tournament* selection method shows little difference to the random selection method of mEDEA. Wilcoxon rank-sum tests confirm that the roulette-wheel and best methods provide significantly different results for both energy and N_{alive}, while no significant difference is observed with the tournament selection method for either metric. The highest pressure selection method *best'* outperforms *roulette-wheel* with statistically significant results at the 0.05 significance level.

Biasing genome spread.

The next set of experiments examines the results of using the two new broadcasting methods, comparing results to the original mEDEA algorithm. Figure 3 clearly shows that experiments E4 and E5 that introduce the new broadcasting methods outperform both the original mEDEA algorithm and the two control experiments. A Wilcoxon rank-sum test with significance level $\alpha = 0.05$ showed that the difference in final energy at generation 500 for both E4 and E5 was statistically different to E1, E2 and E3, but that there is no statistical difference between E4 and E5. The fact that E4



(a) Energy

(b) Energy at gen 500

(c) Active robots

(d) Active robots at gen 500

Figure 1: Explicit selection added to the mEDEA algorithm

[1] as a Shapiro-Wilk test showed that the results were not normally distributed

and E5 differ significantly from controls E2 and E3 show that the differences in performance are not simply attributable to varying the broadcast rate or range, but must be related to the fact that the broadcast rate and range are adjusted according to the estimate of fitness f' calculated by each robot. A corresponding pattern is observed when examining the number of active robots. Plots (e) and (f) within figure 3 clearly show that the number of genomes broadcast significantly decreases with respect to the original methods, but that is compensated for using the higher environmental pressure achieved by adapting what is broadcast based on the quality estimate f'.

In the original mEDEA, as all robots broadcast indiscriminately at the same fixed range, a very weak selection pressure is created that results in genomes that have spread more widely having more chance of being selected if we consider the population as a whole. Behaviours that lead to a robot coming in contact with more robots will result more spreading of genomes and thus on average, a higher probability of generating future offspring. In contrast, the more discriminate methods of broadcasting proposed in this paper create higher selection-pressure: genomes that have higher relative fitness have more chance of being received by other robots than lower fitness ones and thus are more likely to be randomly selected.

Combining explicit selection with biased broadcasting.

Finally, we investigate the effect of combining explicit selection within an individual with biased broadcasting, testing each of the three selection methods in combination with E4 and E5. Figure 2 shows boxplots of the results obtained from using the two $mEDEA_{rf}$ variants and mEDEA. Each of the $mEDEA_{rf}$ variants is significantly better in terms of energy level and active robots compared to standard mEDEA using an explicit selection method, confirmed using a Wilcoxon Rank-Sum test with a significance level $\alpha = 0.05$.

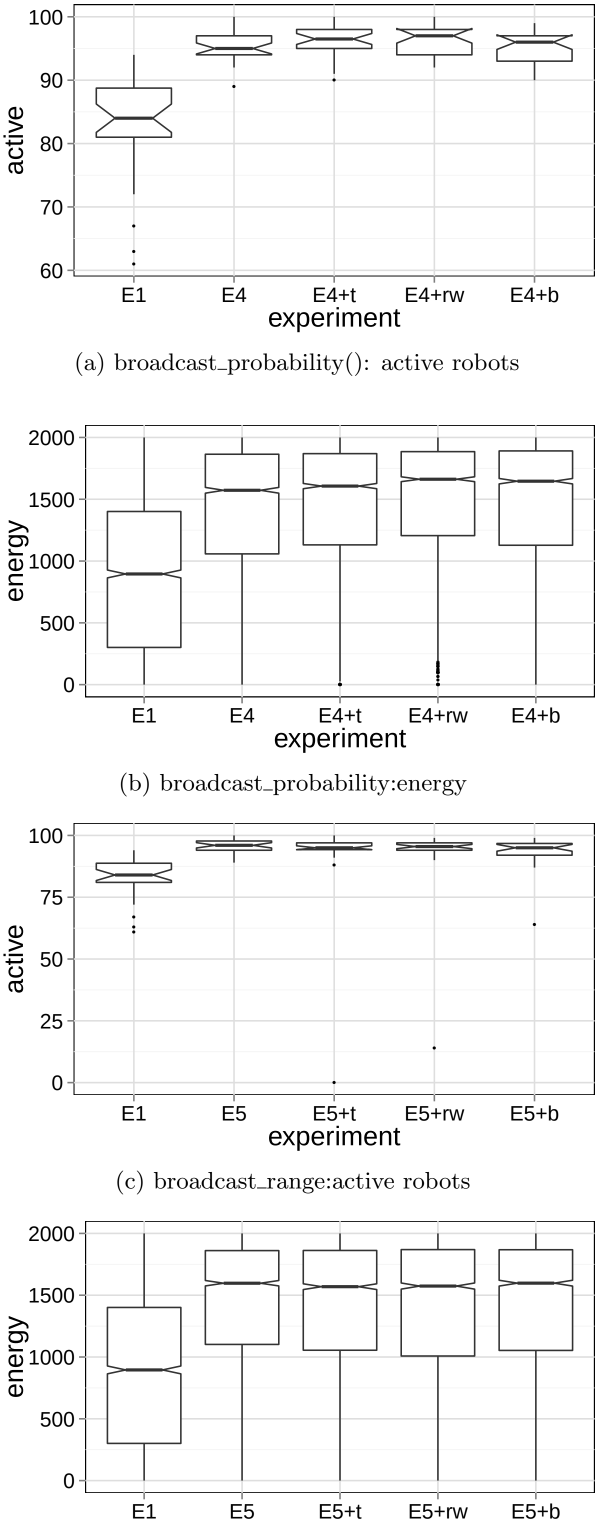
In order to easily contrast the new methods to the original algorithm, figure 4 compares mEDEA, mEDEA+rw, and the two new broadcasting methods combined with roulette wheel selection and in table 2 we use the Wilcoxon test to compare pairs of experiments with and without explicit selection (indicated by $Ei + s\ Ei$ respectively). Statistically significant results are shown in bold.

The following comments can be made that summarise all experiments. Where claims are made, they are evidenced by data that is statistically significant as show in the table.

- Coupling the standard mEDEA algorithm with a high-pressure explicit selection method results in a more robust and sustainable population (higher energy and more alive robots) than the standard mEDEA. However, using a low-pressure explicit selection method does not result in any statistical difference.

- The new methods of biasing the spread of genomes based on relative fitness combined with a random selection method by individiual robts (E4, E5) result in a more robust and sustainable population than mEDEA (higher energy and more alive robots). However there are no discernible differences between the two new methods.

- Coupling the methods for biasing spread of genomes (E4, E5) with an explicit selection method by individuals robots improves on the standard mEDEA but in most cases does not provide any significant advantage over biasing the spread and using random selection,



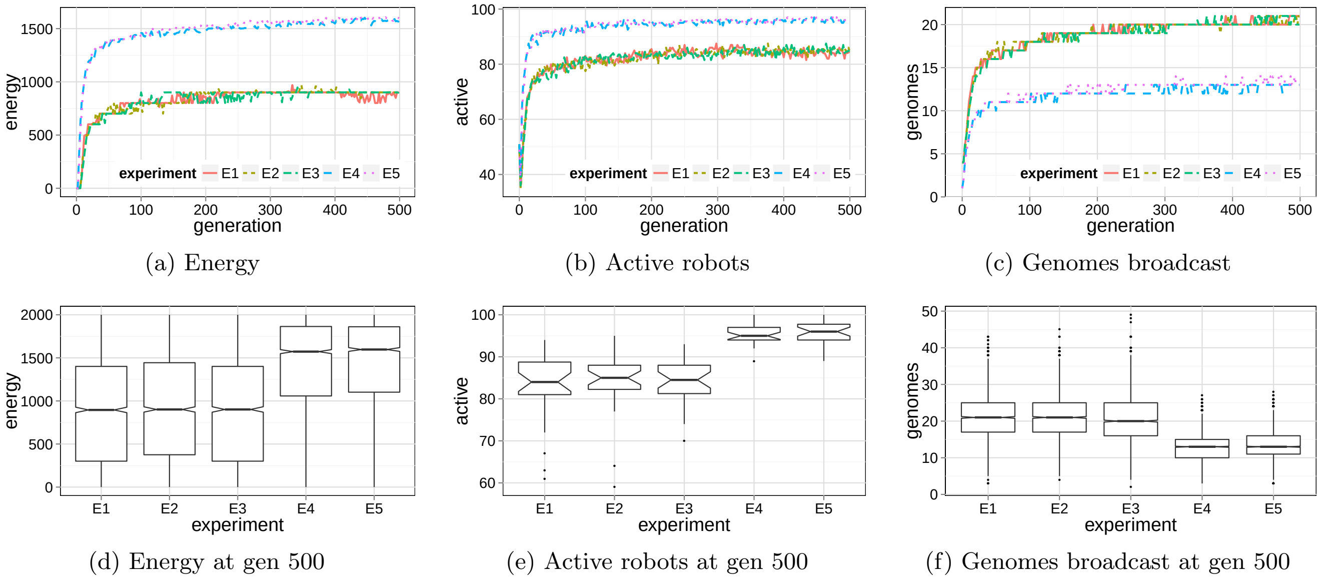
(a) broadcast_probability(): active robots

(b) broadcast_probability:energy

(c) broadcast_range:active robots

(d) broadcast_range():energy

Figure 2: Combining the biased broadcasting of genomes with explicit selection by individuals.



(a) Energy

(b) Active robots

(c) Genomes broadcast

(d) Energy at gen 500

(e) Active robots at gen 500

(f) Genomes broadcast at gen 500

Figure 3: mEDEA, control experiments and biased broadcasting: figures show the energy, number of active robots and genomes received for each of the experiments E1-E5

with the exception of improving energy levels in the case of $E4 + s$ when compared to E4 alone.

- Using roulette-wheel selection combined with standard mEDEA outperforms the two experiments in which the spread of genomes is biased but individuals apply random selection in terms of sustaining higher levels of energy within the population, but has no significant effect on the size of the sustained population. However, note that using the explicit selection method comes at a potentially high cost in terms of the number and range of broadcasts required to implement this when compared to the biased-broadcasting methods.

In summary, the results show that using mEDEA with a relative fitness function that either promotes spread of good genomes (via biasing what is transmitted) or promotes selection of genomes with high energy values (explicit selection) result in swarms that sustain high energy levels and high percentages of active robots when compared to the original version.

However, when considering real, physical robots, it should be clear that broadcasting comes with an overhead in terms of the energy required to communicate. Two factors influence the cost of broadcasting in energy terms — the number of broadcasts made and the broadcast range. For the explicit selection and *broadcast_range()* methods, the same number of broadcasts are made — however *broadcast_range()* results in a range of broadcast distances $\leq r_{\max}$, whereas the explicit selection method combined with mEDEA always broadcasts at $r_{\max}$, thus utilising greater energy. The *broadcast_probability()* method directly reduces the number of broadcasts made with respect to mEDEA as weaker robots broadcast less on average, thus saving energy. Hence, although both methods of influencing genome choice can provide similar results, the methods that modulate the broadcasting behaviours are preferable in reducing communication overhead. When considering real-robots this factor can have a significant impact on survival ability — in many real-life scenarios, the ability to prolong battery life by reducing energy usage might well be critical.

		E1	E4	E5
E4	Energy	**< 2.2e-16**		
	Alive	**3.079e-10**		
E5	Energy	**< 2.2e-16**	0.3521	
	Alive	**4.112e-10**	0.4885	
E1+s	Energy	**< 2.2e-16**	**2.13e-05**	**1.963e-07**
	Alive	**6.207e-08**	0.0684	0.1994
E4+s	Energy	**< 2.2e-16**	**1.227e-07**	**1.007e-05**
	Alive	**1.222e-10**	0.1076	0.2793
E5+s	Energy	**< 2.2e-16**	0.5267	0.1288
	Alive	**3.287e-09**	0.7715	0.7658

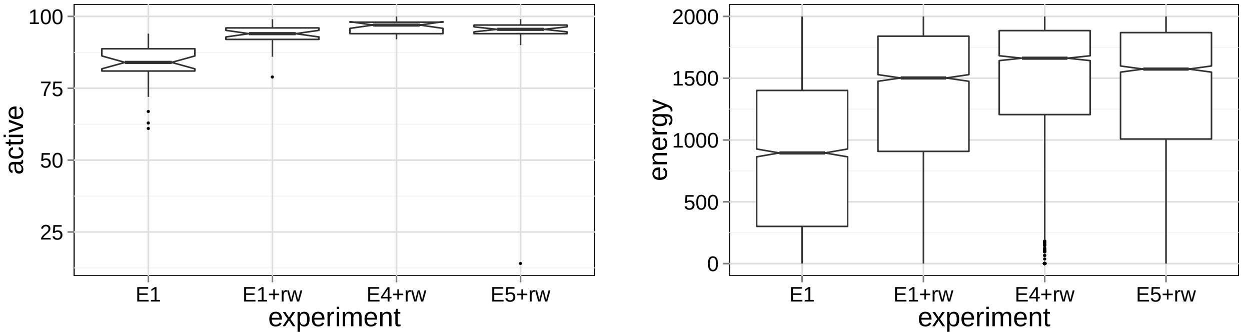
Table 2: p-values obtained from applying Wilcoxon's Rank Sum Test across pairs of experiments, including biased-broadcast only and biased broadcasting coupled with an explicit selection method

6. CONCLUSION AND FUTURE WORK

The paper has provided a number of extensions to an existing Environment Driven Evolutionary Adaptation algorithm — mEDEA. The goal of the work is to show that the integrity of the swarm can be maintained in a more robust manner than in the original work, while still retaining the original distributed and online flavour of the algorithm by using a fitness function that indicated fitness to survive. Having introduced the new fitness function, two new methods were described that adapted either the broadcast range or the probability of broadcasting of a robot, based on its estimate of its relative fitness.

This biases the spread of genomes through the population, with robots that are relatively fitter than their neighbours able to spread their genomes more effectively: individual robots perform a random selection from their store of (now biased) genomes. A thorough analysis of the experimental results shows that a considerable gain in performance is achieved, both in the number of active robots at the end of a fixed period of evaluation, and in the energy levels sustained by those robots.

The new fitness function was also evaluated within an explicit selection method. Experiments showed that this



(a) Active robots at gen 500

(b) Energy at gen 500

Figure 4: Comparison between vanilla mEDEA, mEDEA with explicit selection by individuals and $mEDEA_{rf}$ + explicit selection by individuals

also provided significant improvements over mEDEA, and slightly outperformed the biased broadcast methods in terms of energy sustained. However, as we described above, this comes with a higher cost than either of the biased broadcast methods in terms of the energy used in transmitting. This might be detrimental in a number of real-world scenarios and hence the lower energy-cost methods are preferred.

An obvious extension to this work will include accounting for the cost of broadcasting when calculating the net energy of a robot that is used by the fitness function, to test whether this will differentiate results from the two sets of experiments desribed in this paper. The results provide a robust platform for future experimentation in which user-defined tasks can be added to examine the effects of mixing environment and task driven evolution with a more robust swarm. Given the lightweight nature of the algorithm, an obvious way forward would also include testing on a real hardware platform in the near future.

Acknowledgements

The authors would like to thank Nicolas Bredeche for providing the improved version of Roborobo!, the base of the mEDEA algorithm implementation on which the experiments are based on and the inspirational discussions at the FOCAS summer school 2014.

7. REFERENCES

[1] H.-G. Beyer and H.-P. Schwefel. Evolution strategies – a comprehensive introduction. *Natural Computing*, 1:3–52, 2001.

[2] R. Bianco and S. Nolfi. Toward open-ended evolutionary robotics: evolving elementary robotic units able to self-assemble and self-reproduce. *Connection Science*, 16(4):227–248, 2004.

[3] N. Bredeche and J.-M. Montanier. Environment-driven embodied evolution in a population of autonomous agents. In R. Schaefer, C. Cotta, J. Kołodziej, and G. Rudolph, editors, *Parallel Problem Solving from Nature, PPSN XI*, volume 6239, pages 290–299, Krakov, Poland, 2010. Springer Berlin Heidelberg.

[4] N. Bredeche and J.-M. Montanier. Environment-driven Open-ended Evolution with a Population of Autonomous Robots. In *Evolving Physical Systems Workshop*, East Lansing, United States, 2012.

[5] N. Bredeche, J.-M. Montanier, W. Liu, and A. F. T. Winfield. Environment-driven distributed evolutionary adaptation in a population of autonomous robotic agents. *Mathematical and Computer Modelling of Dynamical Systems*, 18(1):101–129, Feb. 2012.

[6] N. Bredeche, J.-M. Montanier, B. Weel, and E. Haasdijk. Roborobo! a Fast Robot Simulator for Swarm and Collective Robotics, *CoRR*, abs/1304.2, Apr. 2013.

[7] R. Dawkins. *The Selfish Gene*. 1976.

[8] A. E. Eiben. Grand Challenges for Evolutionary Robotics. *Frontiers in Robotics and AI*, 1(4):74, June 2014.

[9] B. Gerkey, R. T. Vaughan, and A. Howard. The player/stage project: Tools for multi-robot and distributed sensor systems. In *Proceedings of the 11th International Conference on Advanced Robotics (ICAR 2003)*, number Icar 2003, pages 317–323, Coimbra, Portugal, 2003.

[10] E. Haasdijk, B. Weel, and A. E. Eiben. Right on the MONEE. In C. Blum, editor, *GECCO2015*, pages 207–214, Amsterdam, The Netherlands, 2013. ACM New York, NY, USA.

[11] J. Lehman and K. O. Stanley. Exploiting Open-Endedness to Solve Problems Through the Search for Novelty. In *Artificial Life XI*, pages 329–336. MIT Press, 2008.

[12] J.-M. Montanier. *Environment-driven Distributed Evolutionary Adaptation for Collective Robotic Systems*. PhD thesis, Université Paris Sud - Paris XI, Mar. 2013.

[13] I. n. F. Pérez, A. Boumaza, and F. Charpillet. Comparison of Selection Methods in On-line Distributed Evolutionary Robotics. In *ALife2014*, 2014.

[14] F. Silva, P. Urbano, S. Oliveira, and A. L. Christensen. odNEAT: An Algorithm for Distributed Online, Onboard Evolution of Robot Behaviours. In C. Adami, D. M. Bryson, C. Ofria, and R. T. Pennock, editors, *Artificial Life 13*, pages 251–258. MIT Press, July 2012.

[15] 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.