Article:
Andrade, GR and Boyle, JH (2020) A minimal biologically-inspired algorithm for robots foraging energy in uncertain environments. Robotics and Autonomous Systems, 128. 103499. ISSN 0921-8890

https://doi.org/10.1016/j.robot.2020.103499

A minimal biologically-inspired algorithm for robots foraging energy in uncertain environments☆

Gabriela R. Andrade*[1] and Jordan H. Boyle[1]

Institute of Design, Robotics and Optimisation (iDRO), University of Leeds

☆ This work has been funded by the Government of Brazil under the project; Science Without Borders.

* Corresponding author.
E-mail addresses: gabrielarabelo@gmail.com; mngra@leeds.ac.uk (G. Andrade); j.h.boyle@leeds.ac.uk (J. Boyle).
[1] Authors are with the Institute of Design, Robotics and Optimisation, at the University of Leeds (UK).

Abstract

This work details the design and simulation results of a bioinspired minimalist algorithm based on *C. elegans*, using autonomous agents to forage for attractant energy sources. The robotic agents are energy-constrained and depend on the energy they forage to recharge their batteries, which is significant as the foraging task is one of the canonical testbeds for cooperative robotics.

The algorithm consists of 6 input parameters which were simulated and optimised in 9 unbounded environments of varying difficulty levels, containing attractant sources which robots would then have to forage from to maintain energy levels and survive the entirety of the simulation.

The robots running the algorithm were then optimised using Evolutionary Algorithms and the best solutions in all 9 environments were categorized with the use of clustering techniques. The clustering results highlighted the different strategies which emerged. Ultimately across the 9 environments, 6 different strategies have been identified. The results demonstrate the applicability of the proposed algorithm to localise attractant sources and harvest energy in different scenarios using the same core algorithm.

1. Introduction

The main goal of the work is to develop and test a bio-inspired reactive foraging algorithm to robots with limited sensing capabilities. The proposed algorithm was optimised in 9 different environments and the results were processed with statistical tools.

In sophisticated sensory systems for robotics, some issues emerge, such as; storing and handling an extensive amount of data, the weight of hardware components and the expense of these components, and dependency on high sensor precision. In a different approach, low-cost minimalistic robots, with limited and inexpensive sensing and processing capabilities, may be applied for solving similar problems either as a collective of independent individuals or as a swarm. Recent developments in the field of autonomous systems and swarm robotics, combined with miniaturized and low-cost electronic devices are setting the groundwork for a new class of robots: capable of performing unsupervised tasks, autonomously managing energy resources. When combined with the developments in the field of smart and biodegradable materials, some recent works even envision autonomous robots completing a life cycle and eventually, dying [1,2]. As a critical aspect on the path to autonomy, energy management usually requires balancing the pursuit of a robot's primary objective with the acquisition and maintenance of its energy supplies [3-7].

In this work, the problem of finding attractants in unknown environments by mobile robots with limited and inexpensive sensing abilities is addressed by a minimalist algorithm designed and based on *C. elegans'* chemotaxis. As in the living organism, the motions of the robots depend on limited sensory data and are governed by a simple set of rules for turns/reversals and for the speed, based on availability, variation and concentration of attractants [8-10]. The algorithm works with the information of one noisy single-pixel analogue sensors and a set of 6 input parameters controlling the sensitivity to attractants and tolerance to changes in its concentration, as well as a tolerance for sensor noise. A similar problem formulation was used by [11], consisting of a homogenous unbounded 2d environment with a number *n* of robots that are initially placed in arbitrary positions and facing arbitrary directions, they also have only one sensor, however whilst the robots in [11] were embodied, preventing overlap, key in measuring aggregation and dispersion, the robots in our simulations were not embodied.

A tailored program was developed to create environments, simulate multiple robots running the Foraging Algorithm, and to process the results. The third-party free program FEX-GODLIKE[1] was connected to the program as a module to perform the optimisation with Evolutionary Algorithms. As the robots were not embodied, there was no reason to expend computational resources with a simulation of physical aspects. Using the framework developed for this work we were able to optimise the use of computational resources, and that made it possible/feasible to run the 229,959 function calls in the optimisation step and the further 21,744 function calls during cross-trial and clustering steps, which although feasible still took a considerable amount of computational hours, 3,318 to be exact.

The foraging task was chosen as it is already considered a benchmark problem in cooperative robotics, consisting of several sub-tasks, including efficient exploration, physical collection, transport, homing and depositing [12-15]. As we envisioned this algorithm possibly being applied to swarm robotics the foraging task seemed like a great equalizer to start with. Allowing the robots' behaviour to be evaluated on the completion of a task which could have varying levels of difficulty, permitting and even pushing the robots to evolve better strategies for the completion of this task. The foraging algorithm's performance was explored in 9 different environments of varying sizes and with varying concentrations of energy sources, some more challenging than others. The controlling parameters were independently optimised for each environment using evolutionary algorithms. The costs are obtained from a based on the performance of a population of identical simulated agents with the candidate genome.

As demonstrated by models, foraging energy in its essence encompasses a basic trade-off, in which energy is spent in an attempt to obtain more [16]. As the evolutionary survival of a species depends heavily on the ability to effectively forage resources, the most efficient foraging behaviours are expected to have been selected, regardless of the size and complexity of the animal [17,18]. *Caenorhabditis elegans*' is an ideal source of inspiration for developing bio-inspired low-cost robots as its foraging strategies had to evolve to make the most of its simple nervous system and anatomy. *C. elegans* is a free-living roundworm which, despite its remarkable simplicity, displays a considerable repertoire of behaviour including foraging, mating, feeding and threat avoidance, as well as rudimentary learning, memory and social behaviour [19-26].

The exploratory behaviour of *C. elegans* emerges from a close interaction of sensory and motor neurons, with a few interneuronal relays in between. The organism moves towards more favourable surroundings by chemotaxis, thermotaxis, and aerotaxis, and evades noxious or harmful conditions through avoiding or escaping behaviours. The exploratory behaviour of *C. elegans* can be decomposed

[1] Copyright © 2018, Rody Oldenhuis. Copyright © 2006, Joachim Vandekerckhove. Available at <https://github.com/rodyo/FEX-GODLIKE>. Access on 03/03/2019.

in simple steps as forward and backward movements, turns, and reversals, that can also be coupled into more complex movements, as short and long reversals, pirouettes, and omega turns (in which the animal bends its body to perform a turn of approximately 180˚) [19-21,27-30]. Several works investigating *C. elegans*' exploratory behaviour identified pattern variations according to: the presence or absence of food; its (short-term) memory about the environment; and its internal state of hunger [19-21,28,31,32]. Three distinct states are identified by [19]: Feeding (Dwelling), Local Search, and Dispersal. When exposed to food, the animals dwell within the peak of the source by performing frequent short reversals. The presence of food also induces a slowing response on *C. elegans* [19,31,32]. Experiments presented in [19] show that the feeding speed reaches about 10% of the animal's foraging speed. According to [31], food-deprived individuals presented an enhanced slowing response compared to well-fed animals. This shows that the food-induced slowing response relates to the internal state of the animal. This frequency immediately drops by a third after the extinction of the food source and continues to drop until almost approaching zero after 30 minutes. On the other hand, long reversals and omega turns are rare while the animals are feeding, increasing immediately after the sources extinguish, and dropping slowly to match the rate of short reversals [19].

The results were processed using a novel set of techniques developed by the authors, named Genetic Analysis. The best solutions evolved in each environment were then combined before being grouped into "species" using clustering techniques (*k-means*++) on the basis of genetic similarity. The results show a very strong relationship between genetic similarity and environmental similarity, as well as the emergence of several behavioural patterns adapted to different environments. In a further step, each species was tested in environments which it had not evolved in originally.

The results lead to two distinct conclusions, one practical and one more academic. From a practical perspective, the results suggest that, when suitably tuned, minimalist *C. elegans*-inspired foraging algorithms can lead to effective navigation to unknown targets even in the presence of repellents and under the influence of a significant sensor noise. From an academic perspective, the work demonstrates that even simple models can serve as an interesting and informative testbed for exploring fundamental evolutionary principles.

This algorithm would not have been possible without the direct influence of evolutionary robotics, which allowed the algorithm to develop independently. The core idea behind evolutionary robotics is that the controllers for robotic systems can be evolved and optimised in simulated environments, using principles and techniques inspired by concepts of darwinian Evolution and of the survival (and reproduction) of the fittest. An initial population is randomly created and tested under certain environmental conditions, each individual being a vector of controlling parameters. The fittest individuals are allowed to reproduce and this process is repeated until a solution that satisfies the performance criterion is found, or it hits the stopping criteria [33,34].

The key to the application of these algorithms is in their simplicity, and adaptability, as I hope to make the code I wrote openly available, it is likely that further applications will arise organically. It is ideally suited for minimalist autonomous robots as it does not require complex hardware.

Overall, the results obtained up to present demonstrate a promising application of bio-inspired behavioural algorithms in the development of autonomous mobile robots. As the algorithm is inspired by a remarkably simple organism, and therefore would not require complex or costly technologies (such as geolocation modules or machine vision systems) these results may also contribute to the advance of energy-efficiency and low-cost autonomous robots. These robots would be affordable and able to remain in the field for long periods without human interference, ideally suited to such applications as agriculture, pest control or monitoring, water monitoring and the overall monitoring of natural systems such as freshwater, forests and the monitoring of invasive species.

Regardless of the application, most of the autonomous robots would potentially benefit from nature-based algorithms aimed at finding optimal strategies for dealing with trade-offs related to tasks and system maintenance.

The best 25 evolved solutions from each environment were then combined before being grouped into "species" using clustering techniques (*k-means*++) on the basis of genetic similarity. The results show a very strong relationship between genetic similarity and environmental similarity, as well as the emergence of several behavioural patterns adapted to different environments. In a further step, each species was tested in environments other than the one it evolved in.

The results demonstrate the applicability of the proposed algorithm to localise attractant sources and harvest energy in different scenarios using the same core algorithm.

Results indicate that due to the emergent pattern future applications may be applied to swarm robotics. Although in this instance they do not communicate or sense each other, it has been noted that in other works swarm robotics may also consist of very minimalist hardware and algorithms, which is interesting in the context of swarm robotics as complex communication devices are also expensive, and emerging patterns can definitely be observed in this algorithm already [11,35].

Furthermore, the work demonstrates a novel application of bio-inspired concepts in the context of the Evolutionary Robotics akin to speciation, using the same algorithm a range of patterns of emergent behaviour appear, similar to the evolution of different species from a common ancestor. The clustering and further characterization based on genetic similarity is, we believe, a novel approach and highly useful allowing to different strategies to emerge instead of a single best. Furthermore this approach is interesting particularly for biologically inspired algorithms as this links closely to the way a living organism may evolve, instead of showing a single best species it shows patterns of emergent behaviour allowing for comparisons between species and their performance in different environments instead of merely a one-size-fits-all approach to evolutionary algorithms.

These low-cost autonomous robots, working as individuals, solve the problem of locating attractants in unknown environments by using limited and inexpensive sensors and a minimalist algorithm designed based on *C. elegans* chemotaxis.

2. Chemotaxis and Reactive Algorithms

Reactive algorithms are ideal for minimalistic, low-cost or disposable robots as they do not demand complex or expensive hardware resources. As highlighted by [36], reactive chemotaxis algorithms when suitable for a given application provide fast, simple and cost-effective solutions. Several authors emphasize the importance of energy-efficient systems for exploration robots which may be used for overland or underground, sea, air, or for space exploration [36-39]. Since their systems require the optimal use of energy resources, when it comes to using robots for long term exploration, it is important that they will have a certain degree of autonomy. Remote access for control is costly both in terms of battery use and the price of the equipment required, and oftentimes can be prone to delays, faults, or interruptions.

Reactive algorithms can also offer a cheaper alternative for disposable robots to be utilised in harmful or infeasible environments for humans or animals and that could destroy or require the robots to be disposed of afterwards. Other applications are environments that are too widespread for other more complex robots to be deemed cost-effective or in long-term autonomous robots operating in unstructured environments [39,40]. Some applications for these algorithms include: spotting light sources which would enable them to use solar panels to recharge the batteries; detecting pollutants or hazardous gases in the air and identifying the source; discovering toxins or hazardous chemicals in water; spotting radiation, fire and overheating.

Many authors investigate reactive chemotaxis algorithms and robot applications to spot chemicals in the air and odour plumes

[37,38,41-48]. Reference [36] presents two different strategies for detecting odour plumes: passive and active. In the passive approach, the robots adopt a stationary monitoring state in the absence of any sensory cues, only breaking into movement and searching for the odour source once chemicals are detected. In the active approach, robots can either perform a linear search or a random walk, actively looking for chemicals in the search field. Once an odour source is discovered, gradient-based algorithms are applied to follow the chemical trail. Some of the algorithms tested by the authors are bio-inspired by organisms such as the Silkworm Moth, the Dung Beetle, and the bacterium *E. coli*. These are compared with one another and with a gradient-based algorithm. All of these algorithms require a stereo chemical sensor and an air-flow sensor (this latter due to the peculiarities of the odour plumes), with the exception of the *E. coli* algorithm which requires only a single chemical sensor.

In this work we present a minimalist reactive foraging algorithm for autonomous mobile robots based on *C. elegans*' chemotaxis and food-induced slowing response, which is then evolutionarily optimised under different environmental conditions.

3. Algorithm Design

The foraging algorithm inspired by *C. elegans*' chemotaxis is composed of two key behaviours: *runs* and *turns*, controlled by a set of six parameters ($g_{1:6}$), also referred to in this work as the genome, as shown in Table 1:

Table 1 - Set of input parameters for the *C. elegans*' bio-inspired minimalist algorithm (genome)

GENE	PARAMETER	APPLICATION
g_1	**VarAngle**	controls the variability of the angle of a turn;
g_2	**NoiseTol**	controls the tolerance of the robot to the variation in sensed light
g_3	**BaseProb**	sets the base probability of turning, in the absence of any change in sensed light level
g_4	**ProbMult**	sets the multiplier (divisor) of the base turning probability when light decreases (increases)
g_5	**Sigm$_\alpha$**	sets the steepness of the sigmoid curve that controls speed according to the sensor reading
g_6	**Sigm$_\beta$**	defines the offset of the sigmoid curve that controls speed according to the sensor reading

At each time step (*t*), each Robot (*R*) adjusts its speed and turn probability according to its sensor reading and possibly makes a turn. The agent's battery level is updated according to the intensity of light it is currently exposed to (i^{in}), as well as how much was spent on moving (i^{out}) and running basic systems (i_{BMR}):

$$Bat_t^R = Bat_{t-1}^R + (Light_t^{x,y} * i_{max}^{in}) - Speed_t^R * i_{max}^{out} - i_{BMR} \ (1)$$

Also at each time step, the simulation program updates robots' positions, checks for extinguished light spots (replacing them with new ones if necessary) and checks which robots are 'alive' – a robot permanently 'dies' if its battery is depleted (Fig.1).

The reasoning process on each time step (*t*) starts when the agent acquires the sensor reading (*SensVal*^R_t) for light intensity (*Light*) at its current position [*x*,*y*], which includes some sensor noise (2). *ΔSensorVal*^R_t is then obtained from the current and previous sensor readings (3) in order to calculate the probability of initiating a turn (*ProbTurn*^R_t).

$$SensVal_t^R = Light_t^{x,y} + random * SensNoise \ (2)$$

$$\Delta SensVal_t^R = SensVal_t^R - SensVal_{t-1}^R \ (3)$$

If Δ*SensorVal*^R_t is sufficiently positive, *ProbTurn*^R_t decreases, whereas if it is sufficiently negative, *ProbTurn*^R_t increases. If the current and the previous values are sufficiently similar (as determined by *NoiseTol*), then *ProbTurn*^R_t maintains the value of *BaseProb* (Fig.1).

Once *ProbTurn*^R_t is set, a random number (0 to 1) is generated and, if it is less than or equal to *ProbTurn*^R_t, the robot will perform a turn (Fig.1). When performing a turn, the yaw (*Δθ*) will be calculated using another random number (between -1 and 1), according to:

$$\Delta\theta_t^R = 180^0 + (VarAngle * random) \ (4)$$

Also, speed is modulated by an inverse logistic function, controlled by the combination of the current sensor reading and the input parameters *Sigm*$_\alpha$ and *Sigm*$_\beta$, according to:

$$Speed_t^R = 1 - (1 + (e^{-Sigm_\alpha * (SensVal_t^R - Sigm_\beta)}))^{-1} * V_{max} \ (5)$$

As the behaviour of the agent is modulated without reference to any internal state variables, this is classed as a reflex-agent model.

We conducted some early experiments with noiseless sensors and found that the agents were able to survive the entire simulation time in all 9 environments, making for rather incomparable results as all of the robots managed to reduce their costs to 0. We therefore included substantial sensor noise in all further experiments, which is also more realistic since all real sensors (particularly low-cost sensors) are subject to noise.

4. Simulation and Experiment Design

The foraging algorithm was implemented and tested in robots simulated in virtual environments, as a requirement for the optimization with evolutionary algorithms (Fig.2). Virtual environments are preferable for a number reasons, as mentioned by [14]; limitation to the robots themselves (battery life, sensors, memory, and localization) and their environments (size, and complexity), furthermore, physical experiments take much longer to run their course allowing for only a few runs per treatment. For these reasons it was decided that a virtual simulation would be more appropriate. The virtual robots are capable of moving forward at variable speed and turning to any direction (0-360˚), and are equipped with a single light sensor, a battery, and a solar panel. Aiming at future application on physical platforms, all the robot's parameters are grounded in real hardware (Table 2).

Table 2 - List of robot parameters, their respective functions and standard simulation values

ROBOT PARAMETER	STANDARD VALUE	APPLICATION
V_{max}	0.5m/s	Maximum locomotion velocity
i_{max}^{in}	1.0A	Optimum solar current
i_{max}^{out}	1.4A	Motor draw at maximum power
BMR	0.2A	Basal Metabolic Rate: cost of running systems when not moving
Bat_{max}	6Ah	Battery capacity
$BatCharge$	20%	Initial battery charge
$SensNoise$	1%	Sensor Noise

3.1. Simulation

The virtual environments consist of simulated 2D fields populated with gaussian light spots of varying peak intensities, which the robots must use to recharge their batteries (Fig.2). The position of the light spots is unknown by the robots and changes randomly throughout the simulation, so they must rely on the temporal gradient of local light intensity to tell whether they are moving in an advantageous or disadvantageous direction. The included sensor noise means that only relatively steep gradients can be reliably sensed, effectively limiting the detection radius of these light spots.

The behaviour of the robots (agents) is a result of the interaction between the environmental conditions and the combination of the input parameters ($g_{1:6}$) tuning the foraging algorithm, which we called the robot's genome. The algorithm controls two basic movement components - runs and turns - by regulating the speed (and hence energy cost) of locomotion and the probability of turning. Speed regulation responds to instantaneous light value, while turn probability responds to the change in light value over time. The intensity of light perceived by the robots is contaminated by normally-distributed sensor noise with variance equal to 1% of the maximum light intensity (aimed at representing the imperfection of the dispersion of chemical cues in water or air, as well as the inherent noise of real sensor hardware).

Each simulation includes 100 identical robots (with the same genome) foraging light sources in a virtual environment over a certain period of time. Each environment is a square field with wrapping boundaries populated with 6 light spots of varying intensity and duration (Fig.2 and Fig.3). For this work, a set of 3 field sizes and 3 light intensity ranges were tested (Fig.4). As the simulation starts, the position of each light spot is randomly assigned, along with its intensity (within the range for that environment) and duration. The light intensity at a distance from the centre decreases according to a Gaussian curve. Each extinguished light spot is replaced by a new random instance so the number of light spots is constant during the whole simulation. For use in later analysis, the average amount of light available per square meter was calculated at each time step and averaged over the total simulation time to obtain the Light Index of each environment.

The total simulation time (*T=24 hours*) is divided into time steps (*ts*), in which the control algorithm is applied. Although each simulation runs with 100 identical robots, there is no interaction between them and their reasoning processes are independent of each other. Besides making behavioural patterns more visible in the graphical plots, another reason for simulating many identical robots is to improve the assessment of a candidate genome by averaging individual's scores (Fig.2 and Fig.3).

3.2. Behavioural Evolution and Optimisation

The algorithm's input parameters ($g_{1:6}$) were optimised for 9 different environments, with varying field sizes and quality of resources (Fig.4). As a preliminary measure of the level of difficulty, we calculated a Light Index for each environment (total light available averaged over time and space). The resulting ranking (from easiest to hardest) is SR, SA, MR, MA, LR, SP, LA, MP, LP. The 25 best individuals evolved in each environment (225 individuals in total) were later merged and classified into 6 different species using clustering techniques (Fig.5).

For each environment, a population of solutions was obtained using multi-objective optimisation with Evolutionary Algorithms [49-53] and the Matlab Program FEX-GODLIKE[2]. Differential Evolution (DE) was chosen after showing the best results when compared to Genetic Algorithms (GA) and Particle Swarm Optimisation (PSO) in preliminary tests. DE was able to find lower costs than the other EAs without

[2] Copyright © 2018, Rody Oldenhuis. Copyright © 2006, Joachim Vandekerckhove. Available at <https://github.com/rodyo/FEX-GODLIKE>. Access on 03/03/2019.

taking much longer than GA (quickest). The number of individuals per generation was set to 200, as this number achieved better results in most of the calibration experiments when compared to 100, 400 and 800. The maximum number of function evaluations was set to 100,000, although most optimisation cycles converged before 30,000. Differential Evolution parameters were set as 95% probability of a new genome to be inserted in the population and [-1.5 , 1.5] as lower and upper boundaries of the scaling parameter.

The performance of each genome (parameter set) was measured according to 3 objectives and averaged by the number of samples (100 robots).

Table 3 - List of Optimisation Objectives (***J***$_{1:3}$)

Objective 1 (J_1)	percentage of total simulation time that each robot survived.
Objective 2 (J_2)	residual energy stored in the batteries at the end of the simulation.
Objective 3 (J_3)	percentage of the field explored by each robot.

The first objective is the most important of the three, representing the capacity of robots to survive in the environments. The second objective is complementary to the first and was introduced as a means of ranking solutions which survive the entire simulation. The third objective was introduced to encourage exploration and penalize genomes whose original placement happened to be more favourable.

The corresponding Costs ($J_{1:3}$) are obtained by subtracting the normalized objective from one. Although Objective 1 takes precedence over the others in this study, all the three were treated equally in the optimisation step. The multi-objective approach was favoured over combining objectives into a single fitness function because it improves the diversity of the solutions and preserves the resolution of the results for the further steps of the study. To do so, the method of Non-Dominated Sorting (NDS) [52] was applied to rank and compare solutions (genomes) in each generation.

As the algorithm and simulated environment are both stochastic, some measures were adopted to compensate for this randomness. Each simulated instance of an environment is created with the same random seed, meaning that the position, intensity and duration of every light spot will behave the exact same way every time a different genome is simulated in that environment. Also, the score of each genome is calculated by averaging the scores of 100 identical robots, which start the simulation in different random positions (*x* and *y*) in the field. The exact behaviour of individual robots will differ due to the stochastic nature of the turning behaviour.

At the end of the optimisation step, all the individuals (genomes) were ranked according to Objective 1 (with objectives 2 and 3 used to break ties) and the best 25 from each environment were selected for further analysis.

3.3. Characterization of Evolved Genomes

This step explored the similarities and differences among the best solutions optimised for each environment by using clustering techniques to find patterns in the combination of the input parameters (genomes). We used the *k-means*++ algorithm [54] to cluster datasets composed by the best 25, 50, 75, and 100 genomes from each environment (all grouped together).

We also tested 4 distance metrics: squared euclidean distance; sum of absolute differences; one minus the cosine of the included angle between points (treated as vectors); one minus the sample correlation between points (treated as sequences of values). For each of these methods combined, we tested k values (number of clusters) from 1 to 20. The best fit for the clustering was found with 25 of each environment (225 genomes), squared euclidean distance, 20 replications, and up to 500 iterations.

When applying any clustering method, the number of clusters is a key parameter, and using too many or too few will make the results uninformative. The quality of any given clustering was quantified in terms of the accumulated sum of point-to-centroid distances. When this was plotted against cluster number (Fig.6), there was no clear single “elbow” on which to base the choice of optimal cluster number. As such, the representativeness of the clusters was further quantified based on a comparison between the performance of the individual cluster members and the cluster centroid in all 9 environments (Section 3.3.1).

Each application of *k-means*++ ran with up to 500 iterations and 10 repetitions. On top of that, we ran *k-means*++ clustering 10 times for 3 to 10 clusters (values greater or less than this were clearly unsuitable). Ultimately six clusters (K=6) were chosen for this analysis as it provided a balance between optimal accuracy and readability of results. In order to facilitate the discussion that follows, each cluster will be referred to as a Species.

3.3.1. Cross-Trials of Evolved Genomes and Centroids

In this step, the best 25 individuals from each environment (a total of 225) were tested outside their native environment. Each simulated instance of the 9 environments was exactly the same as the one used in the optimisation step.

The same procedure was adopted for each cluster centroid and the clustering scheme was chosen based on the accuracy of the results obtained for the cluster centroid and all its members (Fig.7)

5. Experimental Results and Discussion

4.1. Parameter Optimisation for Different Environments

In this step, the algorithm's input parameters ($g_{1:6}$) were optimised in 9 different environments. To keep the results consistent, all the simulated instances of a certain environment were exactly the same. All the optimisation cycles converged between 100 and 200 generations, before the maximum number of function calls were reached (Table 4).

There is a strong correlation (-0.83) between the optimised Cost 1 values and Light Index for each environment. However, there is some discrepancy in the order (easiest to hardest) according to the two metrics. Specifically, the ranking of environments in terms of the optimised values of Cost 1 (low to high) is: SR, MR, SA, LR, MA, SP, LA, MP, LP (Fig.8, Table 4), while the ordering with respect to Light Index (total light available in the field, averaged by time) is: SR, SA, MR, SP, MA, LR, LA, MP, LP (Fig.9). In general, rich environments perform better in practice than their Light Index would predict. This is a logical outcome because there are no "traps" (light spots whose peak intensity is too low to provide a net positive battery current over and above the current consumption).

Furthermore, it must be acknowledged that the optimisation process is stochastic so some solutions may be local optima (we are investigating this with more experiments). More generally, the Light Index may not capture all relevant aspects of the spatio-temporal light distribution (we are investigating alternative metrics).

Table 4 - List of Environments, Experiment Numbers and Optimisation Results

ENVIRONMENT		**Function Eval.**	**Minim. Cost 1**	**Minim. Cost 2**	**Minim. Cost 3**	**Number of Non-Dominated Solutions**	**Light Index**
Field Size	**Quality of Resources**						
S (150)	R	10,636	0.000	0.673	0.673	75	40.752
	A	37,093	0.018	0.869	0.869	13	27.614
	P	9,534	0.523	1.000	1.000	8	13.807
M	R	37,699	0.000	0.809	0.809	13	18.338

(225)	A	38,256	0.426	0.986	0.986	6	12.422
	P	9,844	0.724	1.000	1.000	13	6.211
L (300)	R	32,246	0.253	0.952	0.952	12	10.241
	A	40,797	0.671	0.998	0.998	11	6.935
	P	12,854	0.719	1.000	1.000	32	3.468
	TOTAL:	**228,959**					

4.2. Cross-trials

In a broad view, we can distinguish two basic groups: one composed by species evolved in Rich and Average environments, and the other composed by species evolved in Poor environments:

- ❑ **Genomes evolved in Rich and Average environments** exhibit quite similar performance across all Rich and Average environments, generally achieving costs quite close to those of the native populations in these environments. They all performed significantly worse than the native populations and the other group in Poor Environments (Fig.10).
- ❑ **Genomes evolved in Poor environments** exhibit quite a similar performance in the Poor environments, generally achieving costs quite close to those of the native populations, but performing significantly worse than the native populations in Rich and Average environments. Surprisingly, the genomes evolved in LP outperform the native populations in SP and in MP (Fig.10).

4.3. Characterization of Evolved Genomes

The results of genotype clustering (Fig.11) reveal some striking patterns. Firstly, at most two species feature in each of the 9 environments, with a single species present in five of these. It is also noticeable at a first glance that some pairs of clusters are more similar to each others' than to the other clusters (1 and 2, 3 and 4, 5 and 6), which will also be discussed further ahead.

The two species from Large Poor (LP) environments (Sp$_5$ and Sp$_6$) appeared to have minimised sigmoid position (g_6) at the same time that they have maximised sigmoid steepness (g_5) (the resulting sigmoids are shown in Fig.13). Therefore they evolved to stop suddenly in partially favorable conditions which other species would not consider favourable enough to reduce speed or stop. That explains why they perform worse in the other environments because they fall into light spots with low intensity (traps), missing the better quality food sources available in Rich and Medium environments.

These two Species (Sp_5 and Sp_6) have also minimized the value of Base Probability of Turn (P0_turn/g_3). This, combined with relatively high values of Noise Tolerance (g_2), means they have evolved an unexpected strategy where they virtually never turn, instead moving straight until they hit a light spot and stop, exploiting the wrapping boundaries of the world. It can be observed that in these two species native to LP, the values of Probability Multiplier (g_4) and Turn Variation (g_1) are distributed across a broad range (particularly when considering Species 5 and 6 together). This makes sense because their evolved strategy means that these genes are irrelevant (because they don't turn). In fact, observation of the simulated behaviour suggests that Species 5 and 6 actually exhibit virtually identical behaviour. Given that only genes 1 and 4 are functionally irrelevant, they are subject to random genetic drift. The clustering algorithm (which is concerned with gene values rather than behaviour) appears to have separated these into two species mainly on the basis of high / low values of g_4.

The Large Poor (LP) environment was the only one in which a strategy using no turns at all evolved, and as a result the genetic profiles of Species 5 and 6 are strikingly different from all others (see Fig.12). Focussing on Species 1 to 4, the differences are more subtle but still significant. Most notably, when comparing Species 3 and 4 (evolved in Poor environments SP and MP) to Species 1 and 2 (evolved in Rich and Average environments), there is a striking difference in the sigmoid genes (g_5 and g_6). Comparing the resulting shapes of the sigmoids (Fig.13) shows that Species 3 and 4 will stop much more suddenly and in lower quality light sources than Species 1 and 2. This is a necessary adaptation to the low-quality energy sources available in Poor environments.

When comparing Species 3 and 4 to each other, the only somewhat significant difference in again in the sigmoid genes (g_5 and g_6). The light intensity required to bring robots to very low speeds is similar, but Species 4 has a sharper transition (Fig.13). The behavioural relevance of this is unclear, and the fact that these two species perform very similarly in cross-trials (Fig. 6) and are both native to MP (although only Species 3 is native to SP) suggests that it is minimal.

Finally, when comparing Species 1 and 2 the only significant genetic differences are in g_4 and g_6 (see Table 5). The difference in g_6 means that Species 1 (found primarily in Rich environments) slows down at somewhat higher light levels than Species 2 (found primarily in Average environments), which is a subtle but logical adaptation to higher consistency of good quality energy sources found in Rich environments. The fact that Species 1 has a somewhat higher value of ProbMult (g_4) is consistent with the fact that Rich environments will, on average, have higher intensity light spots and hence steeper light gradients. This means that a sensed change in light intensity is more likely to be due to the underlying gradient (as opposed to sensor noise), and therefore warrants a stronger effect on turning probability.

Table 5 - Decoded Centroid Genome of each species and respective standard deviation

		GENE					
		g_1	**g_2**	**g_3**	**g_4**	**g_5**	**g_6**
		VarAngle	**NoiseTol**	**BaseProb**	**ProbMult**	**Sigm$_\alpha$**	**Sigm$_\beta$**
SPECIES	**1**	109° ± 33°	1.64 ± 0.17	0.003 ± 0.001	180.7 ± 29.9	11.8 ± 4.52	0.27 ± 0.07
	2	112° ± 31°	1.59 ± 0.12	0.003 ± 0.001	160.4 ± 38.7	11.8 ± 3.49	0.16 ± 0.05
	3	92° ± 33°	1.35 ± 0.11	0.004 ± 0.004	150.3 ± 61.2	36.7 ± 26.60	0.08 ± 0.04
	4	85° ± 29°	1.41 ± 0.17	0.004 ± 0.001	179.7 ± 19.8	73.7 ± 24.41	0.11 ± 0.03
	5	236° ± 110°	2.95 ± 1.01	0.000 ± 0	169.0 ± 56.8	91.1 ± 7.06	0.00 ± 0
	6	298° ± 52°	3.62 ± 0.73	0.000 ± 0	35.2 ± 27.5	94.1 ± 4.22	0.00 ± 0

6. Conclusions

The work presented here leads to two distinct conclusions, one practical and one more academic. From a practical perspective, our results suggest that, when suitably tuned, a minimalist *C. elegans*-inspired foraging algorithm can lead to effective navigation to unknown targets even in the presence of significant sensor noise. Furthermore, it suggests that this approach could be used to achieve long-term autonomy by allowing robots to seek out energy sources in their environments. Our simulations used realistic physical parameters (speed, current, battery capacity, etc), but these results would benefit from validation on real hardware.

From an academic perspective, the work demonstrates that simple models can serve as an interesting and informative test bed for exploring fundamental evolutionary principles. Despite having only six free parameters in the behavioural model and two in the environment model, we have achieved results representative of the evolutionary differentiation of species. The work replicates, in a very simplified form, the way that modest changes in environmental conditions lead to evolutionary adjustments (see differences between Species 1 to 4, or between a pigeon and a dove) while sufficiently extreme changes can result in the same building blocks being used in fundamentally different ways (see differences between Species 5/6 and 1 to 4, or a penguin and a dove). Furthermore, our approach of grouping solutions evolved in different environments and then analysing these on the basis of genetic (as opposed to behavioural) similarity is novel, to the best of our knowledge, and yielded significant results that aligned very well with behavioural outcomes.

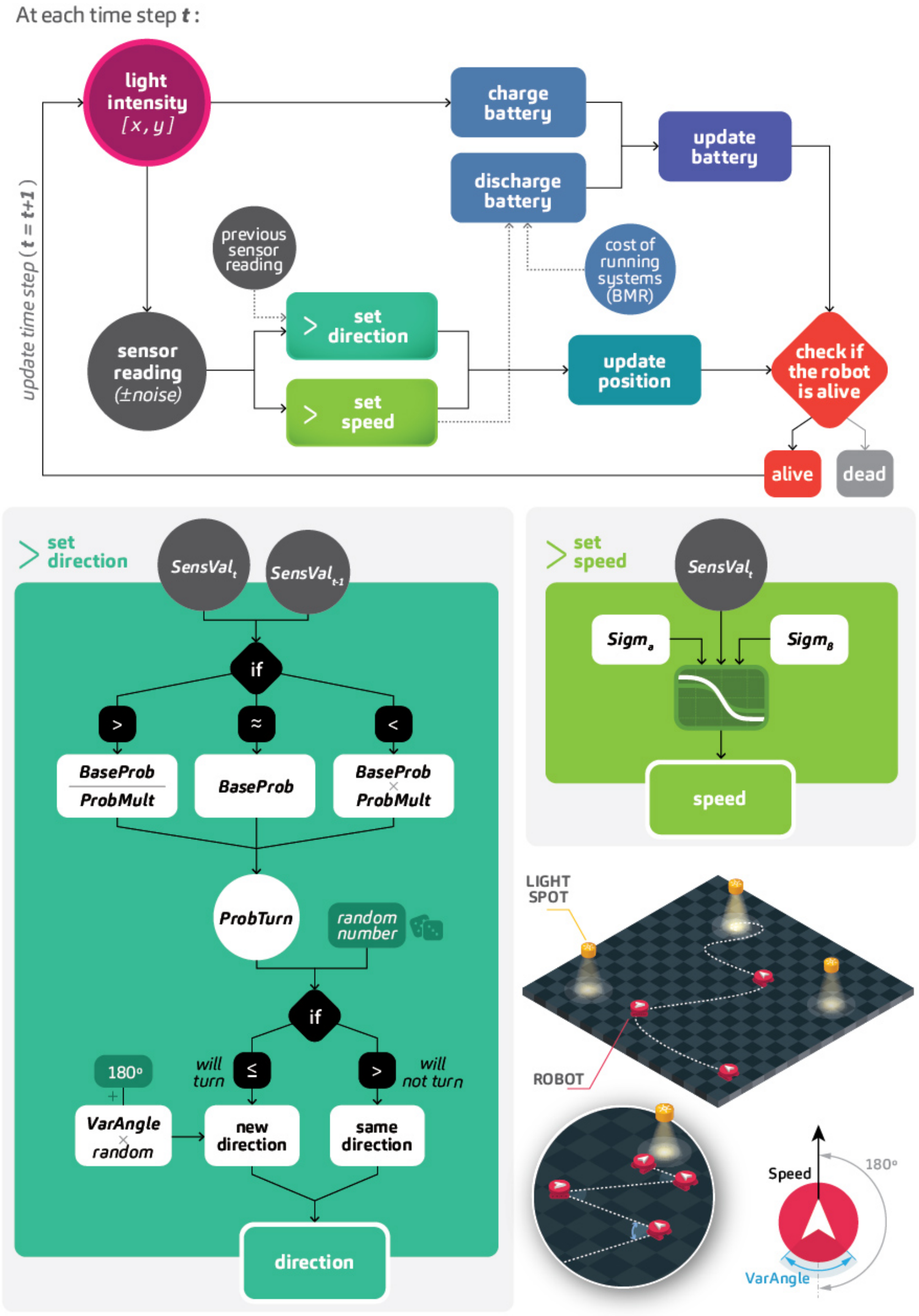


Figure 1 - Flowchart of the control algorithm applied at each time step of a simulation. Top: overview. Bottom: detailed flowchart of processes of setting direction (left) and setting speed (mid-right). Bottom-right: conceptual illustration of the simulation and of the robot.

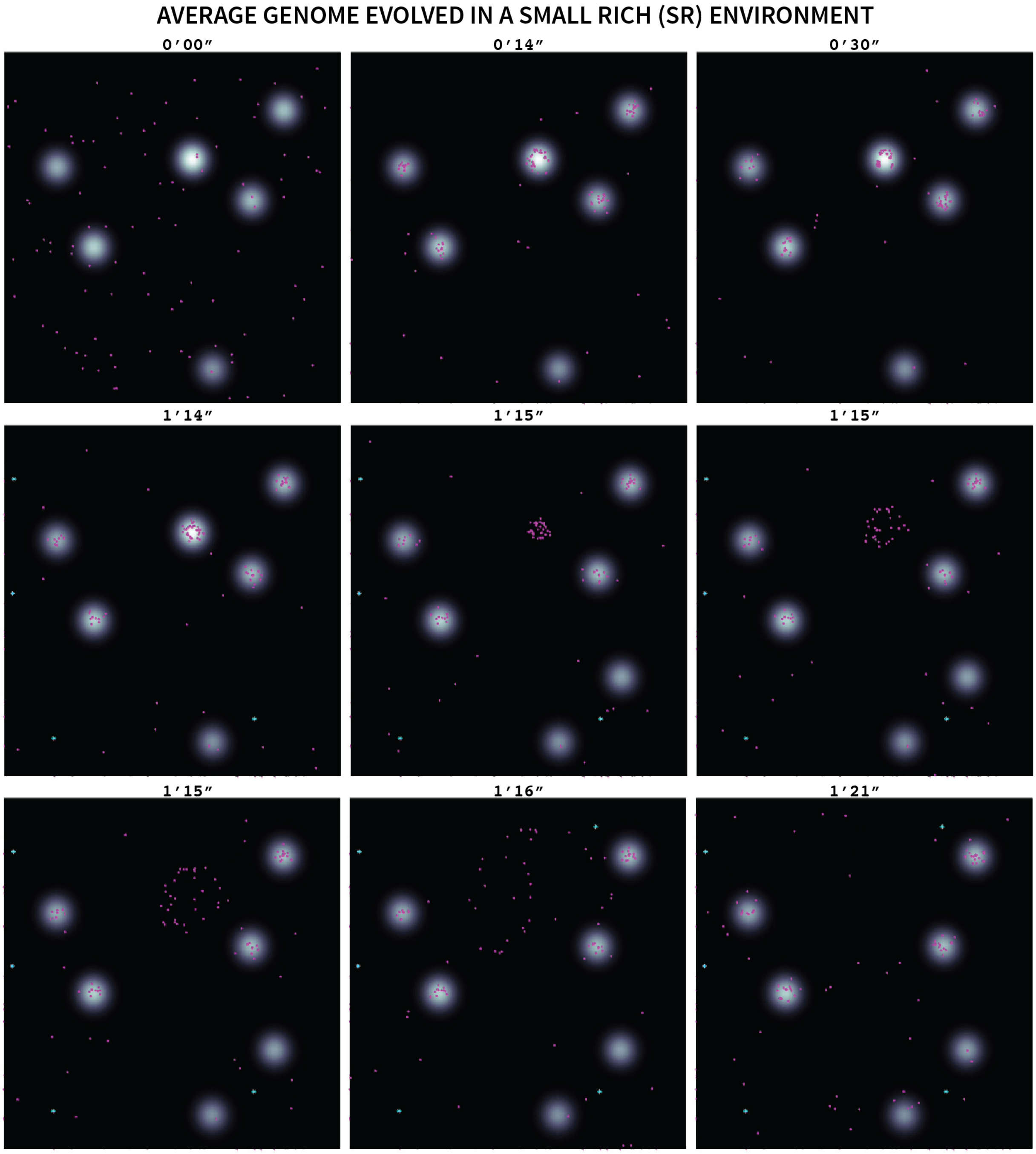


Figure 2 - Graphical plots of the simulation of an average genome evolved in a small environment, rich in resources (SR). White spots represent attractant resources. Pink dots represent each clone robot running the foraging algorithm with the same genome. Asterisk-shaped cyan marks represent dead robots. At the beginning of the simulation (0'00), some robots are found to have been fortunate to be placed right in a spot of light. At 0'14 to 0'30, many robots are observed to have found light spots. In 1'14, some of the clone robots started to run out of battery (and die). Right after 1'14, a light spot at the top-centre of the screen extinguishes, being instantly replaced by another one, in the bottom right. From 1'15 to 1'21, the dispersion of the robots that were feeding in the extinguished light spot can be observed.

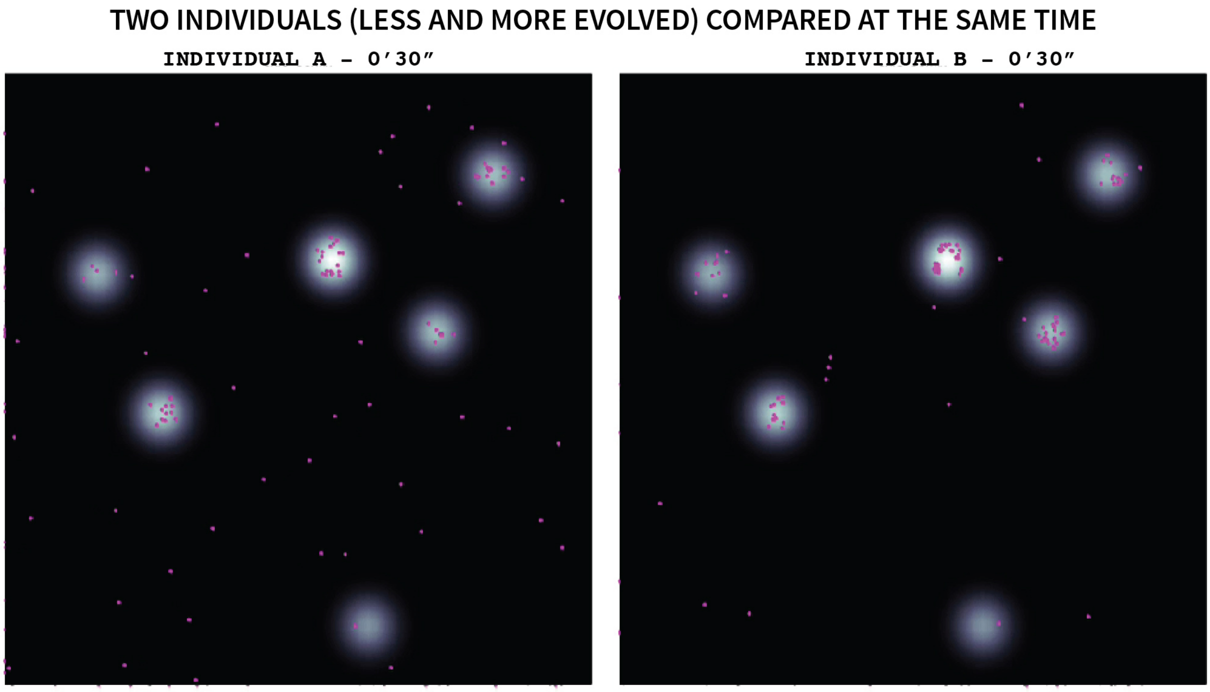


Figure 3 - Comparison of two individuals/genomes evolved in a small environment, rich in resources (SR), observed at time 0'00". Genome A (left) is less evolved than B (right). White spots represent attractant resources. Pink dots represent each clone robot running the foraging algorithm with the same genome.

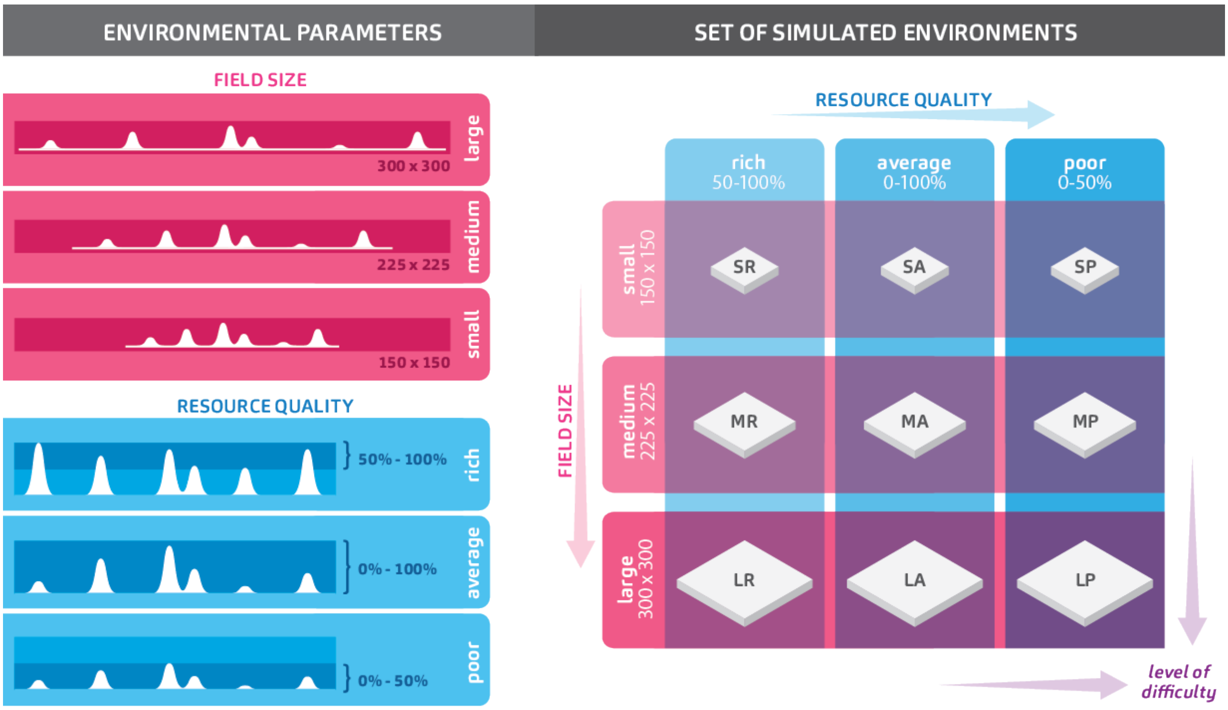


Figure 4 - Simulated Environments. Left: environmental parameters Field Size (small, medium, large) and Quality of Resources (poor, average, rich). Right: Set of 9 environments resulting from the combination (3-by-3) of the environmental parameters.

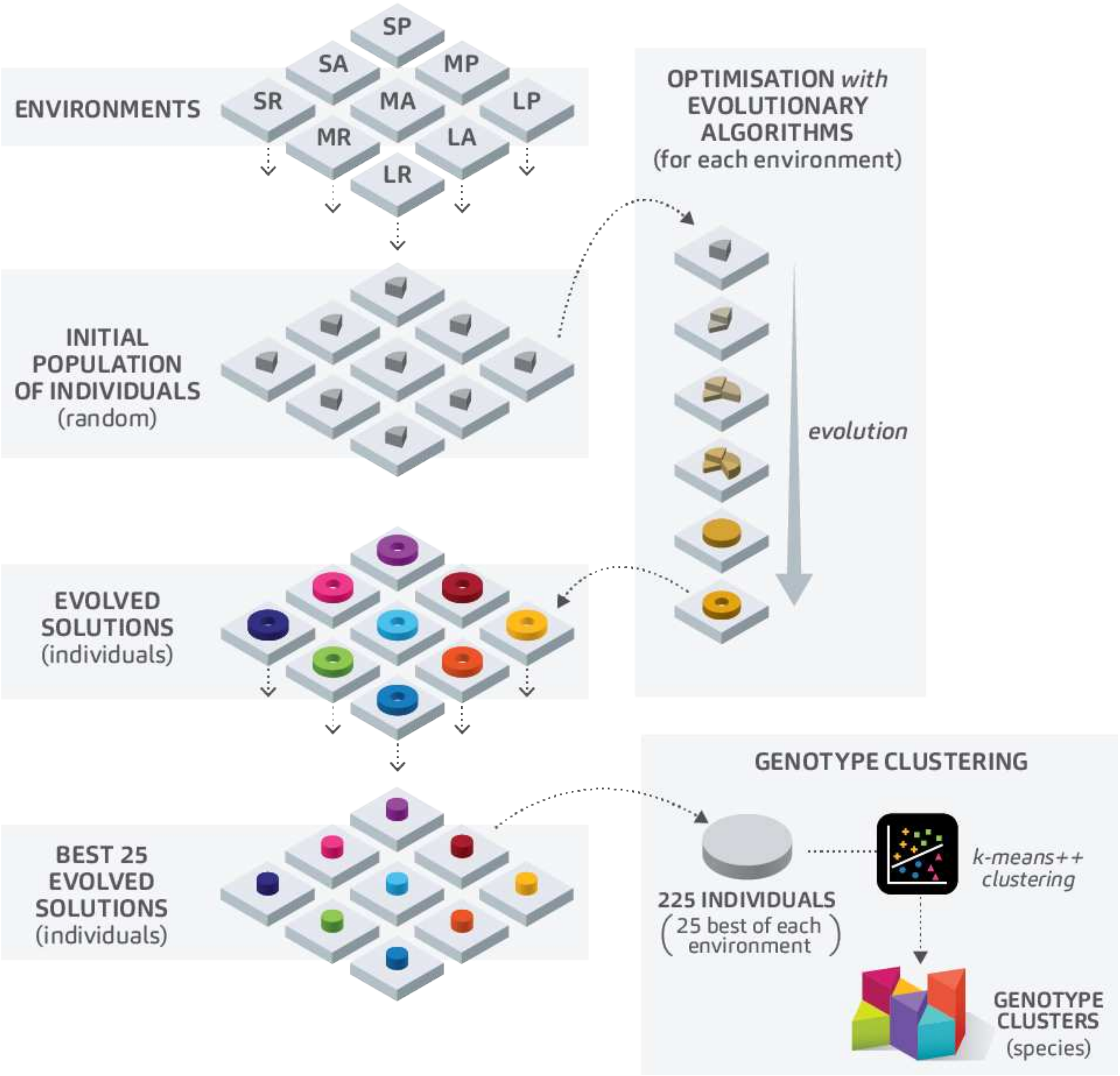


Figure 5 - Diagram of the methods of virtual evolution of individuals and classification into species.

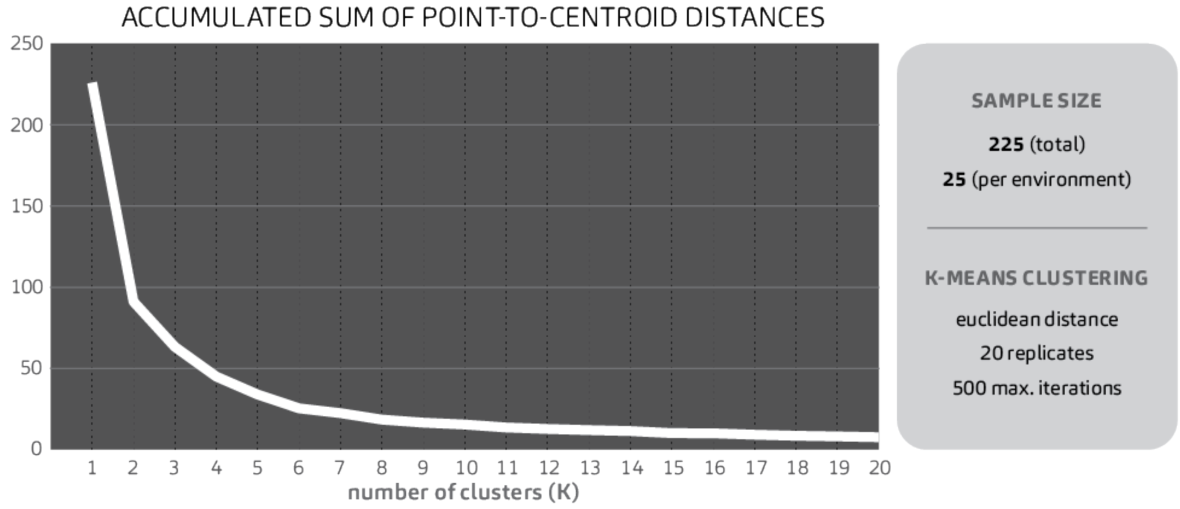


Figure 6 - The Accumulated sum of Point-to-centroid distances for different cluster numbers.

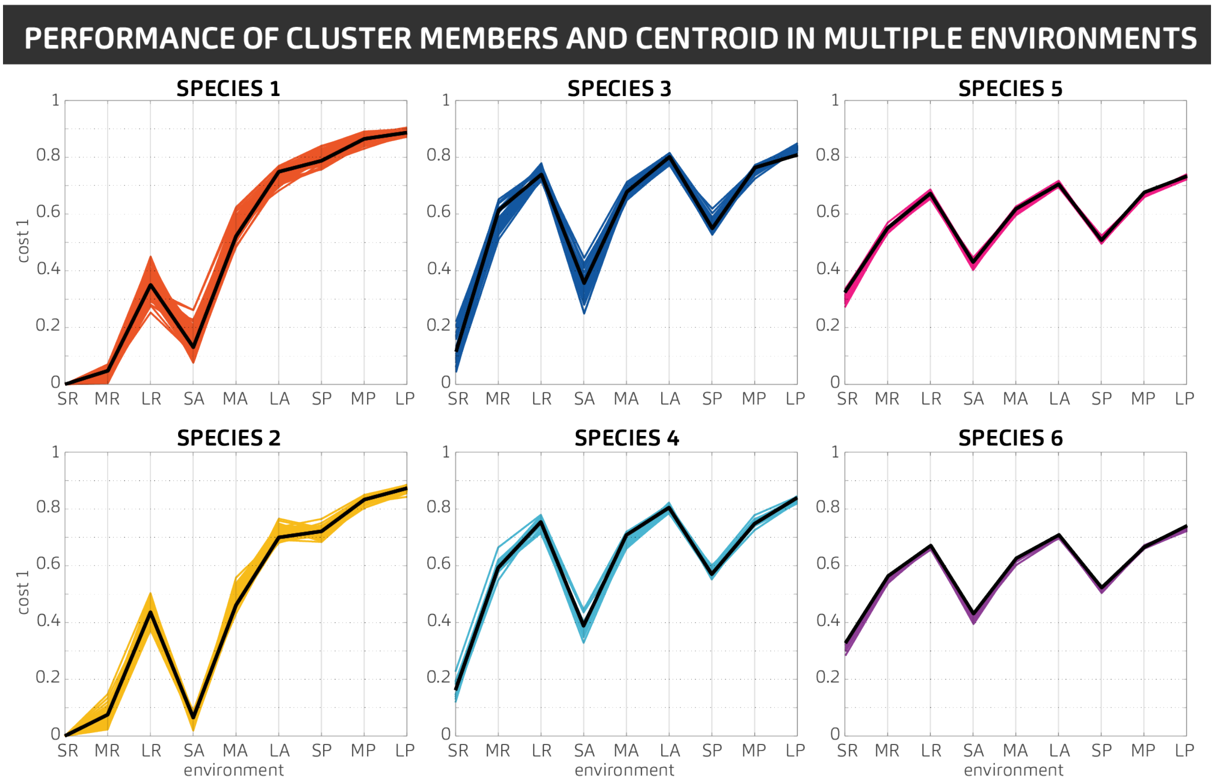


Figure 7 - Compared performance of cluster centroid (black line) and cluster members (colour lines), each tested in the 9 environments. Clusters obtained from the best 25 genomes evolved in each environments (225 in total). The tight grouping of the cluster members and cluster centroids indicate that the clustering was effective, with each cluster representing class of solution (species).

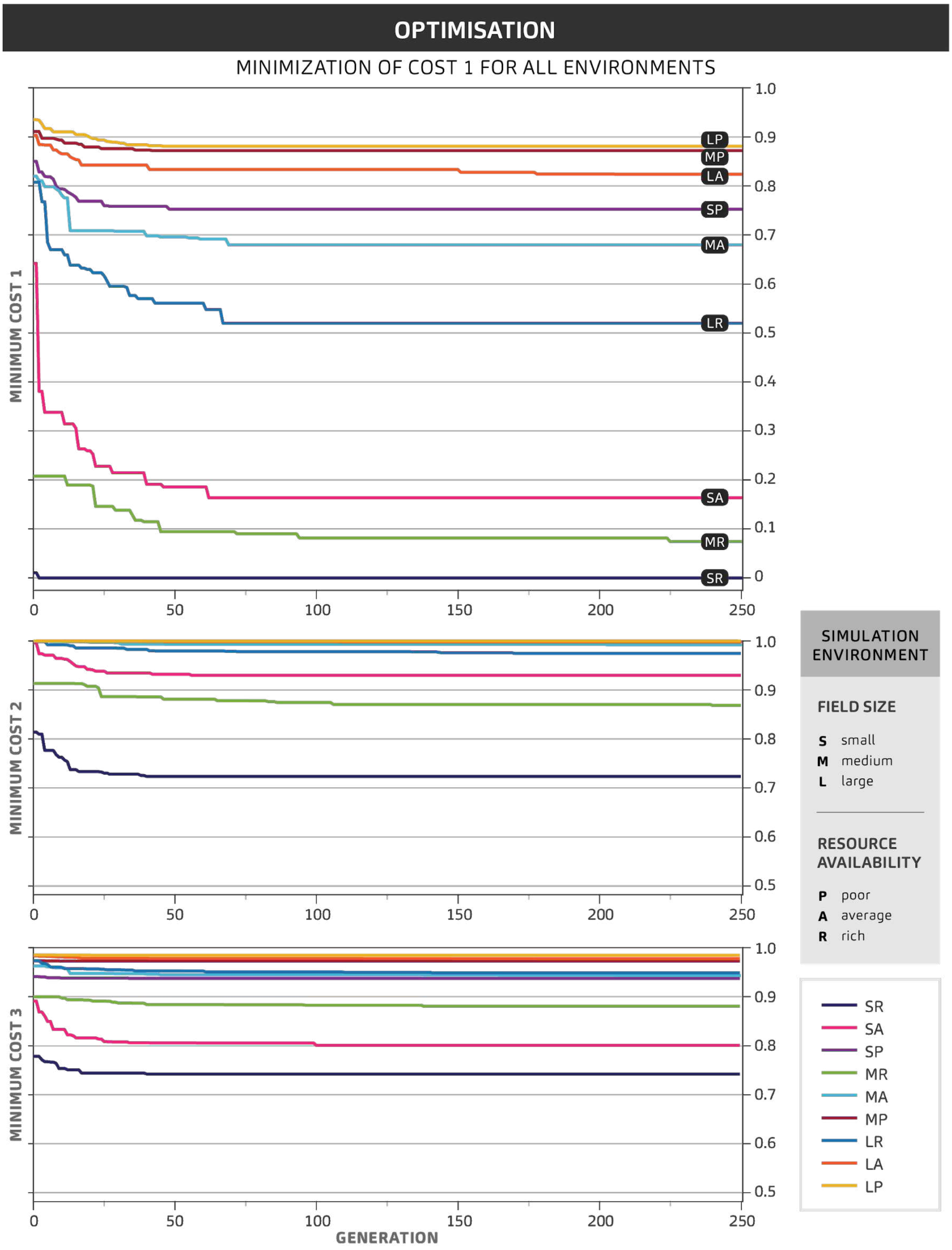


Figure 8 - Minimization of Costs 1, 2 and 3 for all the 9 environments. Each line represents the optimisation of parameters $g_{1:6}$ for a given environment. The model was optimised using the Differential Evolution algorithm with 200 individuals per generation.

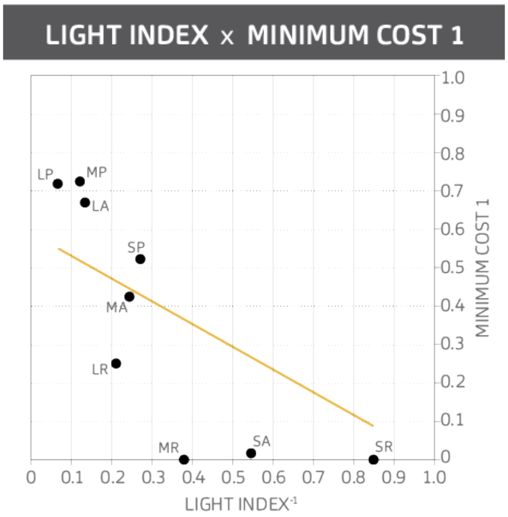


Figure 9 - Light Index versus Minimised value of Cost 1 in all Environments. Pearson Correlation Coefficient: -0.826; Spearman Correlation Coefficient: -0.9

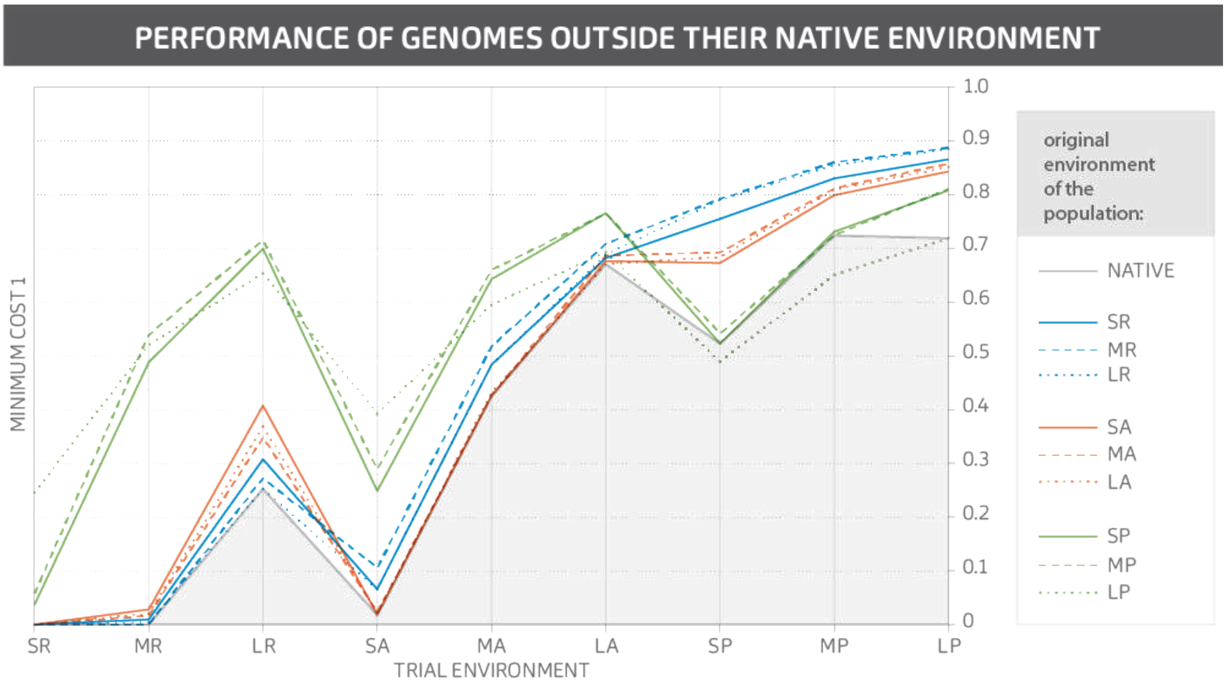


Figure 10 - Performance of the best genomes (25 of each environment) in all the 9 environments.

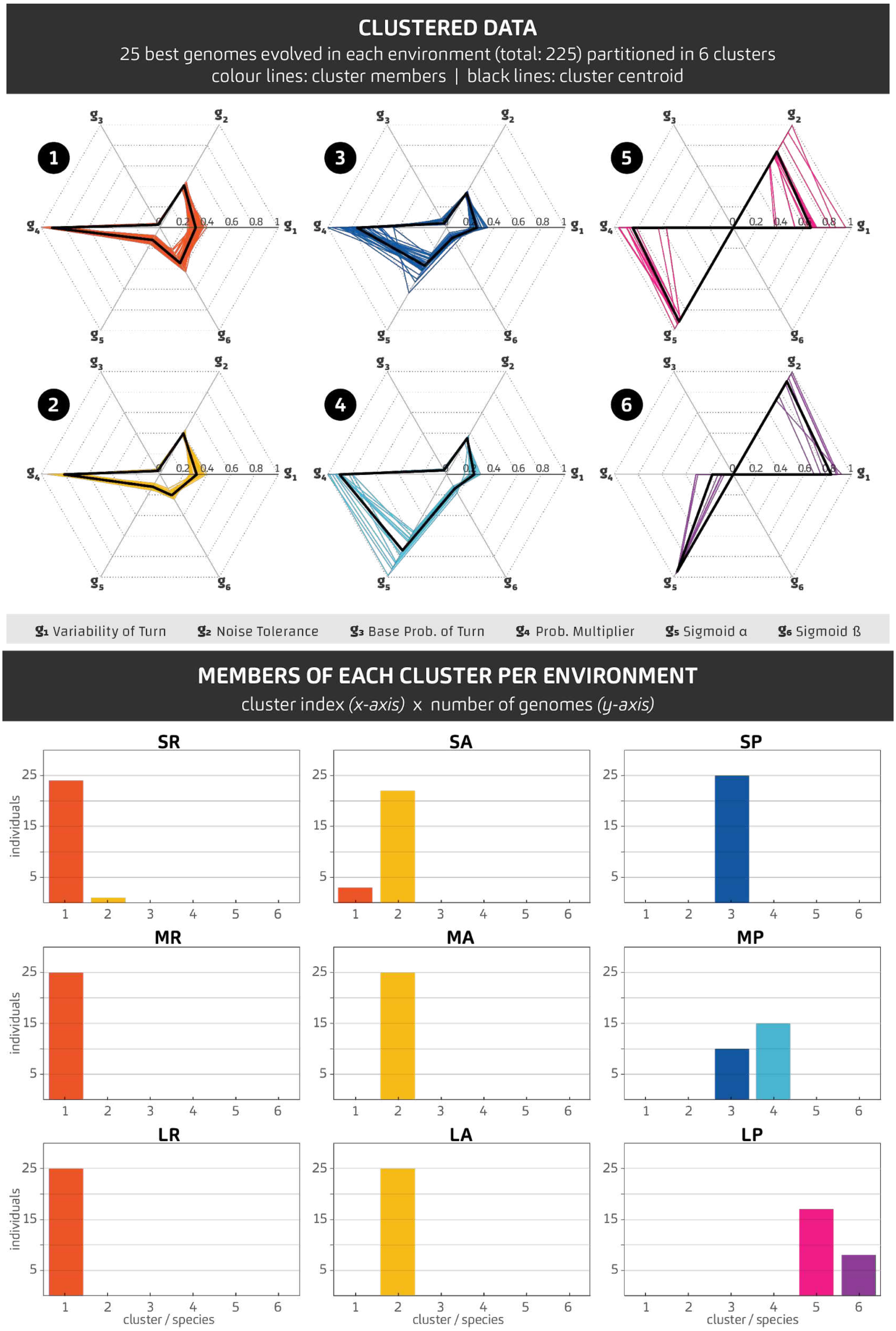


Figure 11 - Results of Genotype Clustering. Top: all members of the six clusters represented as spider plots (coloured lines) and the cluster centroid (black lines). Each radial axis of a spider plot represents a value ($g_{1:6}$) of the input vector. Bottom: Distribution of members of each cluster within the population of each environment.

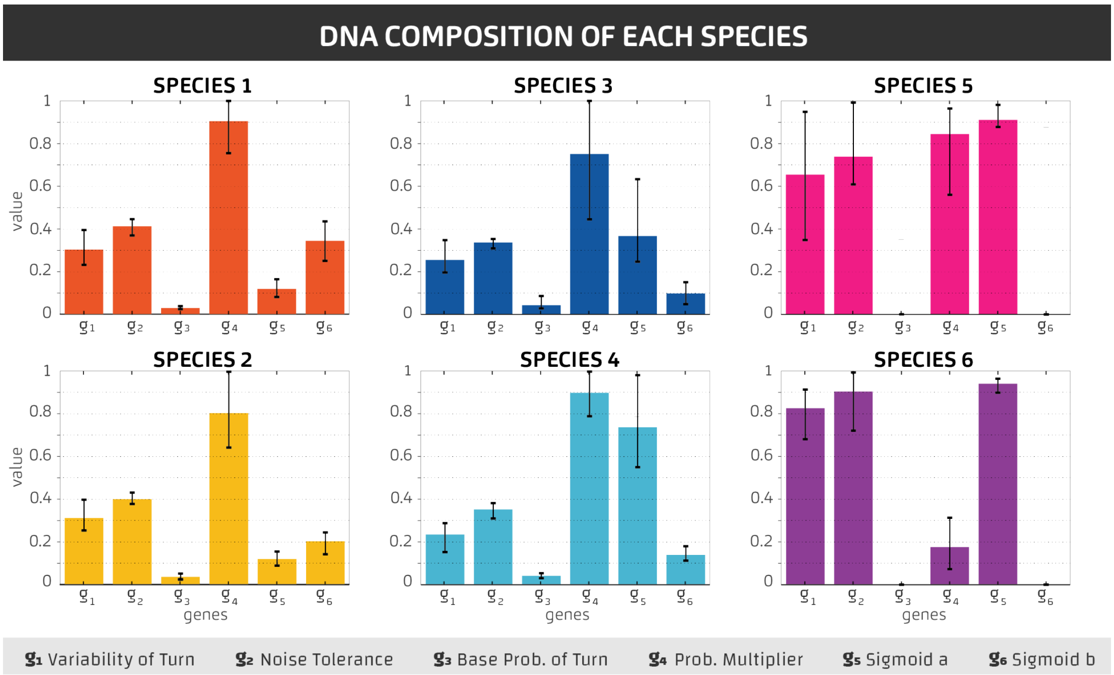
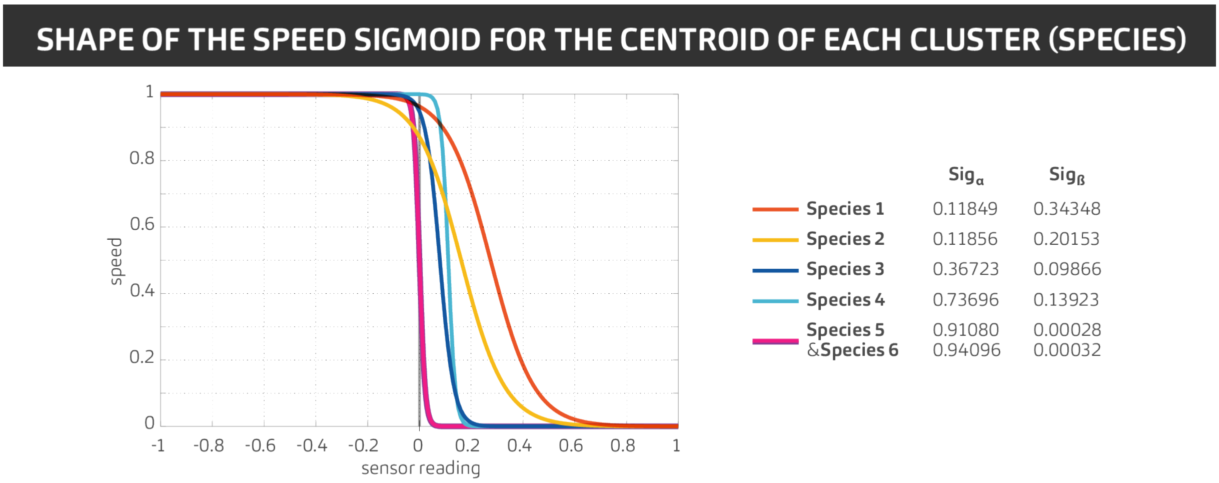


Figure 12 - Bar chart with DNA composition of each species.



	Sig_α	Sig_β
Species 1	0.11849	0.34348
Species 2	0.11856	0.20153
Species 3	0.36723	0.09866
Species 4	0.73696	0.13923
Species 5 &Species 6	0.91080 0.94096	0.00028 0.00032

Figure 13 - Variation in the shape of the speed sigmoid function for the centroid of each cluster.

References

[1] H. Philamore, J. Rossiter, A. Stinchcombe, I. Ieropoulos, Row-bot: An energetically autonomous artificial water boatman, in: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2015: pp. 3888–3893. doi:10.1109/IROS.2015.7353924.

[2] H.Q. Raja, Self-Sufficiency of an Autonomous Self-Reconfigurable Modular Robotic Organism, 2014.

[3] M. Milford, R. Schulz, Principles of goal-directed spatial robot navigation in biomimetic models, Philosophical Transactions of the Royal Society B: Biological Sciences. 369 (2014) 20130484–20130484. doi:10.1038/nn.2730.

[4] K. Berns, E. von Puttkamer, Autonomous Land Vehicles, Vieweg+Teubner, Wiesbaden, 2009. doi:10.1007/978-3-8348-9334-5.

[5] R. Siegwart, I.R. Nourbakhsh, D. Scaramuzza, Introduction to Autonomous Mobile Robots, MIT Press, 2011.

[6] A. Barrera, Advances in Robot Navigation, InTech, 2011. doi:10.5772/705.

[7] T. Barfoot, J. Kelly, G. Sibley, Special Issue on Long-Term Autonomy, The International Journal of Robotics Research. 32 (2014) 1609–1610. doi:10.1177/0278364913511182.

[8] J.T. Pierce-Shimomura, Analysis of the effects of turning bias on chemotaxis in C. elegans, 208 (2005) 4727–4733. doi:10.1242/jeb.01933.

[9] S. Ward, Chemotaxis by the Nematode Caenorhabditis elegans: Identification of Attractants and Analysis of the Response by Use of Mutants, in: Proc. Natl. Acad. Sci. U.S.a, 1973: pp. 817–821. doi:10.1073/pnas.70.3.817.

[10] T.C. Ferrée, S.R. Lockery, Computational rules for chemotaxis in the nematode C. elegans, J Comput Neurosci. 6 (1999) 263–277.

[11] M. Gauci, J. Chen, W. Li, T.J. Dodd, R. Groß, Self-organized aggregation without computation, 33 (2014) 1145–1161. doi:10.1177/0278364914525244.

[12] O. Zedadra, H. Seridi, N. Jouandeau, G. Fortino, An Energy-Aware Algorithm for Large Scale Foraging Systems, Scpe. 16 (2016) 1–18. doi:10.12694/scpe.v16i4.1133.

[13] O. Zedadra, H. Seridi, N. Jouandeau, G. Fortino, A Cooperative Switching Algorithm for Multi-Agent Foraging, Engineering Applications of Artificial Intelligence. 50 (2016) 302–319. doi:10.1016/j.engappai.2016.01.025.

[14] O. Zedadra, N. Jouandeau, H. Seridi, G. Fortino, Multi-Agent Foraging: state-of-the-art and research challenges, Complex Adaptive Systems Modeling. (2017) 1–24. doi:10.1186/s40294-016-0041-8.

[15] R.A. Meyers, Encyclopedia of Complexity and Systems Science, Encyclopedia of Complexity and Systems Science:. (2009). doi:foraging.

[16] R.L. Bettinger, Hunter-gatherer Foraging: Five Simple Models, Cambridge University Press, 2009. doi:10.1017/S0002731600040907.

[17] A.I. Houston, J.M. McNamara, Models of Adaptive Behaviour, Cambridge University Press, 1999.

[18] G.M. Viswanathan, M. Da Luz, E.P. Raposo, H.E. Stanley, The physics of foraging: an introduction to random searches and biological encounters, 2011.

[19] J.M. Gray, J.J. Hill, C.I. Bargmann, A circuit for navigation in Caenorhabditis elegans, Proc. Natl. Acad. Sci. U.S.a. 102 (2005) 3184–3191. doi:10.1073/pnas.0409009101.

[20] M. de Bono, Molecular approaches to aggregation behavior and social attachment, J. Neurobiol. 54 (2002) 78–92. doi:10.1002/neu.10162.

[21] E.N. Allen, J. Ren, Y. Zhang, J. Alcedo, Sensory systems: their impact on C. elegans survival, Neuroscience. 296 (2015) 15–25. doi:10.1016/j.neuroscience.2014.06.054.

[22] C.I. Bargmann, Chemosensation in C. elegans, WormBook. (2006) 1–29. doi:10.1895/wormbook.1.123.1.

[23] Z.F. Altun, D.H. Hall, eds., WormAtlas, 2018. http://www.wormatlas.org.

[24] L. Frézal, M.-A. Félix, C. elegans outside the Petri dish, 4 (2015) e05849. doi:10.7554/eLife.05849.

[25] S.W. Emmons, The beginning of connectomics: a commentary on White et al. (1986) “The structure of the nervous system of the nematode Caenorhabditis elegans,” Philosophical Transactions of the Royal Society B: Biological Sciences. 370 (2015) 20140309–20140309. doi:10.1098/rstb.2014.0309.

[26] J.H. Boyle, S. Berri, N. Cohen, Gait Modulation in C. elegans: An Integrated Neuromechanical Model, Front. Comput. Neurosci. 6 (2012) 1–15. doi:10.3389/fncom.2012.00010.

[27] L.C.M. Salvador, F. Bartumeus, S.A. Levin, W.S. Ryu, Mechanistic analysis of the search behaviour of Caenorhabditis elegans, Journal of the Royal Society Interface. 11 (2014) 20131092–20131092. doi:10.1098/rsif.2013.1092.

[28] J.L. Donnelly, C.M. Clark, A.M. Leifer, J.K. Pirri, M. Haburcak, M.M. Francis, et al., Monoaminergic Orchestration of Motor Programs in a Complex C. elegans Behavior, PLoS Biol. 11 (2013) e1001529–17. doi:10.1371/journal.pbio.1001529.

[29] D. Kim, S. Park, L. Mahadevan, J.H. Shin, The shallow turn of a worm, Journal of Experimental Biology. 214 (2011) 1554–1559. doi:10.1242/jeb.052092.

[30] O.D. Broekmans, J.B. Rodgers, W.S. Ryu, G.J. Stephens, Resolving coiled shapes reveals new reorientation behaviors in C. elegans: figures and supplements, 5 (2016) 428–11. doi:10.7554/eLife.17227.

[31] E.R. Sawin, R. Ranganathan, H.R. Horvitz, C. elegans locomotory rate is modulated by the environment through a dopaminergic pathway and by experience through a serotonergic pathway, Neuron. 26 (2000) 619–631.

[32] B. Zhao, P. Khare, L. Feldman, J.A. Dent, Reversal Frequency in Caenorhabditis elegans Represens an Integrated Response to the State of the Animal and its Environment, Journal of Neuroscience. 23 (2003) 5319–5328. doi:10.1002/neu.480210403.

[33] S. Nolfi, D. Floreano, Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines, Bradford Book, 2004.

[34] T. Gomi, Evolutionary Robotics. From Intelligent Robotics to Artificial Life, Springer, 2003.

[35] S.O. Obute, M.R. Dogar, J.H. Boyle, Chemotaxis Based Virtual Fence for Swarm Robots in Unbounded Environments, in: N.T. Nguyen, R. Kowalczyk, S.-M. Chen (Eds.), Biomimetic and Biohybrid Systems, Springer International Publishing, Cham, 2019: pp. 216–227. doi:10.1007/978-3-030-24741-6_19.

[36] R.A. Russell, A. Bab-Hadiashar, R.L. Shepherd, G.G. Wallace, A comparison of reactive robot chemotaxis algorithms, Robotics and Autonomous Systems. 45 (2003) 83–97. doi:10.1016/S0921-8890(03)00120-9.

[37] X.-X. Chen, J. Huang, Odor source localization algorithms on mobile robots: A review and future outlook, 112 (2018) 123–136. doi:10.1016/j.robot.2018.11.014.

[38] Q. Lu, Q.-L. Han, S. Liu, A finite-time particle swarm optimization algorithm for odor source localization, Information Sciences. 277 (2014) 111–140. doi:10.1016/j.ins.2014.02.010.

[39] M. Moline, K. Benoit-Bird, Sensor Fusion and Autonomy as a Powerful Combination for Biological Assessment in the Marine Environment, Robotics. 5 (2016) 4–13. doi:10.3390/robotics5010004.

[40] S. Yu, C.S.G. Lee, Lifetime maximization in mobile sensor networks with energy harvesting, Icra. (2011).

[41] A. Lilienthal, A. Loutfi, T. Duckett, Airborne Chemical Sensing with Mobile Robots, Sensors. 6 (2006) 1616–1678. doi:10.3390/s6111616.

[42] W. Naeem, R. Sutton, J. Chudley, Chemical Plume Tracing and Odour Source Localisation by Autonomous Vehicles, J. Navigation. 60 (2007) 173–18. doi:10.1017/S0373463307004183.

[43] T. Lochmatter, X. Raemy, A. Martinoli, Odor source localization with mobile robots, (2007).

[44] A. Gongora, J. Gonzalez-Jimenez, Olfactory telerobotics. A feasible solution for teleoperated localization of gas sources? Robotics and Autonomous Systems. 113 (2018) 1–8. doi:10.1016/j.robot.2018.12.008.

[45] G. Ferri, E. Caselli, V. Mattoli, A. Mondini, B. Mazzolai, P. Dario, SPIRAL: A novel biologically-inspired algorithm for gas/odor source localization in an indoor environment with no strong airflow, Robotics and Autonomous Systems. 57 (2009) 393–402. doi:10.1016/j.robot.2008.07.004.

[46] Y. Kuwana, S. Nagasawa, I.S.B. and, 1999, Synthesis of the pheromone-oriented behaviour of silkworm moths by a mobile robot with moth antennae as pheromone sensors, Biosensors & Bioelectronics. (n.d.). doi:10.1016/s0956-5663(98)00106-7.

[47] T. Lochmatter, A. Martinoli, Theoretical analysis of three bio-inspired plume tracking algorithms, in: 2009 IEEE International Conference on Robotics and Automation, IEEE, 2009: pp. 2661–2668. doi:10.1109/robot.2009.5152686.

[48] W. Li, J.A. Farrell, R.T. Card, Tracking of Fluid-Advected Odor Plumes: Strategies Inspired by Insect Orientation to Pheromone, Adaptive Behavior. 9 (2001) 143–170. doi:10.1177/10597123010093003.

[49] C.A.C. Coello, G.B. Lamont, D.A. van Veldhuizen, Evolutionary Algorithms for Solving Multi-Objective Problems, Springer-Verlag New York, Inc, 2006.

[50] K. Deb, S. Agrawal, A. Pratap, T. Meyarivan, A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II, in: G. Kampis, I. Karsai, E. Szathmáry (Eds.), Advances in Artificial Life: Darwin Meets Von Neumann. 10th European Conference, ECAL 2009, Springer Berlin Heidelberg, Berlin, Heidelberg, 2000: pp. 849–858. doi:10.1007/3-540-45356-3_83.

[51] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Computat. 6 (2002) 182–197. doi:10.1109/4235.996017.

[52] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, 1989.

[53] N. Srinivas, K. Deb, Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms, Evolutionary Computation. 2 (1994) 221–248. doi:10.1162/evco.1994.2.3.221.

[54] D. Arthur, S. Vassilvitskii, k-means++: The Advantages of Careful Seeding, in: SODA Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, 2007: pp. 1027–1035.

Authors' Biography

Gabriela R. Andrade

PhD researcher at the Institute of Design, Robotics and Optimization (iDRO) in the Department of Mechanical Engineering at the University of Leeds (UK) working on the topic of Bio-Inspired Algorithms for Autonomous Robots. Masters in Design, Materials, Processes and Technology at UEMG (Brazil). Graduated with a bachelor degree in Industrial Design from UEMG in 2010. Her main research interests also include biomimetics, algorithm design and ecology.

Jordan H. Boyle

Received the B.Sc. (Hons.) and M.Sc. degrees in electrical engineering from the University of Cape Town, Cape Town, South Africa, in 2002 and 2005, respectively, and the Ph.D. degree from the School of Computing, University of Leeds, Leeds, U.K., in 2010. His Ph.D. research involved deciphering and modeling the neuro-mechanical basis of locomotion in the nematode *C. elegans*, in collaboration with experimental biologists. He is currently a Lecturer of Engineering Systems in the School of Mechanical Engineering at the University of Leeds. His primary research interests include bio-inspired robotics, with a specific focus on the control and mechanics of locomotion.

Authors' Photos

Gabriela R. Andrade	**Jordan H. Boyle**