Evolution in Robotic Islands

Optimising the design of autonomous robot controllers for navigation and exploration of unknown environments

Final Report

Authors: Angelo Cangelosi[1], Davide Marocco[1], Martin Peniak[1], Barry Bentley[1], Christos Ampatzis[2], Dario Izzo[2]

Affiliation: (1) University of Plymouth (UK)
(2) Advanced Concept Team, ESA

Date: 25 June 2010 (Draft Report)

Contacts:

Name: Angelo Cangelosi
Position: Professor of Artificial Intelligence and Cognition
Address: School of Computing and Mathematics, Drake Circus,
           PL4 8AA, Plymouth UK
Tel: +44 1752 586217
Fax: +44 1752 586300
e-mail: acangelosi@plymouth.ac.uk

Advanced Concepts Team
Fax: +31(0)715658018
e-mail: act@esa.int

Ariadna ID: 09-8301
Ariadna study type: Standard
Contract Number**:** 22708/09/NL/CBI

Available on the ACT website
http://www.esa.int/act

Evolution in Robotic Islands: Draft Study Report

Table of Content

Contents

1 INTRODUCTION AND OBJECTIVES OF THE STUDY

The scientific and technological rationale of this project was to integrate the island model paradigm for parallelisation with the automatic controller design methodology of Evolutionary Robotics. This integration allows the investigation of the conditions and contexts in which the utilisation of the island algorithm can enhance the robustness, generality of autonomous robotics controllers.

1.1 Background and Motivation

In the past years the application of artificial evolution to the optimisation of neuro-controllers for autonomous robots has been gaining a lot of momentum. The approach called Evolutionary Robotics (ER) is, essentially, a methodological tool to automate the design of robots' controllers. It is typically based on the use of artificial evolution to find sets of parameters for artificial neural networks that guide the robots to the accomplishment of their task. With respect to other design methods, ER has the theoretical advantage that it does not require the designer to make strong assumptions concerning what behavioural and communication mechanisms are needed by the robots.

However, to-date, the complexity of the tasks solved by agents controlled by evolved neuro-controllers is lower than the complexity achieved by other methods using hand-coded controllers driven by expert knowledge. Also, even if automatic techniques could in principle reduce the human effort required to design controllers, this is usually not the case. In other words, the complexity achieved by automatic approaches seems incommensurate with the effort expended in setting up and configuring the evolutionary system. Therefore, despite the theoretical advantages of automating the design problem for autonomous agents, the robotics control community cannot yet claim to having reaped its benefits. More effort should be put by researchers in reducing the computation time required until a solution to a problem at hand is obtained with these techniques, and at the same time in creating a framework that can generate more complex solutions without a significant effort overhead on the side of the experimenter. Various approaches in the literature have explored the possibility of enhancing the efficiency of automatic design tools, such as incremental evolution and symbiotic evolution, but the focus of such methods is not on creating a generic and simple design framework and certainly not on the algorithmic side.

Typically, ER researchers launch a number of independent evolutionary runs, each differentiated by a different initial random seed. Subsequently, for each run, the best individual of the last generation, or alternatively the best individual encountered throughout evolution is identified and analysed (post-evaluated). Thus, some of the runs end up successful, and others not. By successful run we mean a run where the fitness obtained is the maximum, or one where a neuro-controller is produced that is able to drive the robots to the accomplishment of their task. Every evolutionary run is characterised by the prevalence of a certain genetic material that gets spread through recombination and survives through selection. Variation of genetic material within the population comes through random mutation only.

This project is based on the idea of adding the island model paradigm to the ER arsenal, thus using the island model to evolve artificial neural networks controlling robots. The theory of punctuated equilibria inspired Cohoon et al. in formulating the island migration model, a coarse-grained parallelisation approach to global optimisation (GO), and more specifically, genetic algorithms. Thus, the underlying mechanisms of this stochastic global optimization technique inspired by Darwinian evolution, namely recombination, mutation and selection, are complemented by

migration. Initially isolated populations, while evolving in parallel, exchange individuals at a certain rate, thus interacting with each other. As a result, not only do GA populations evolve faster, but their performance is also improved. The island model paradigm has also been applied to the parallelisation of other evolutionary algorithms, as differential evolution (DE) and applied on difficult and high dimensional global optimisation problems (see Izzo et al. 2009). In this case of ER, the experimenter will still be launching multi-start evolutionary runs, only this time, the dynamics of the stochastic search in a single run will not be fully determined by the initial random seed, but also co-determined by the migration among runs and information exchange. Populations will be invaded with genetic material shaped elsewhere. Migrants might represent solutions radically different than the solutions developed in a given population. The analogies between the biological observations and the effect of migration in GO still hold when the application is the ER methodology. In particular, exploration arises from migration and exploitation from isolated evolution in islands. Finally, while in GO the island model introduces new local minima and optimisers to a pool of solutions, in ER migration represents introducing new ways of solving the task in a pool of existing solutions. This is because a solution is, apart from a fitness number, the behavioural capabilities of autonomous, simulated or real agents that are evaluated in a given environment and with respect to a given task. This paves the way to addressing more complex ER scenaria, including evolving populations for different tasks in different islands, and then letting migration perform the mixing of genetic material and behavioural capabilities.

The project will build up on first and preliminary results by Ampatzis et al. (see Ampatzis et al. (2009), where the island model is applied to evolving neuro-controllers. In particular, the authors introduced the island model paradigm to the evolutionary robotics methodology and presented one test case where migration of genetic material across evolving populations not only accelerated evolution but also created better individuals, managing to obtain solutions to the problem at hand.

1.2   Aims

The main aim of the project was to study how the integration of the evolutionary robotics methodology with the island model can contribute to the design of robust and general controllers for navigation and exploration in unknown environment. Moreover, the project aimed at the investigation of how this integrated methodology facilitates the design of move advanced tasks, such as in the context of active vision. The project was able to address these aims by building on the previous space robotics research at both the University of Plymouth and the ACT Team. The robotics team at Plymouth had developed a prototype simulator of the Mars Rover to carry our ER simulation experiments on autonomous navigation (Penial et al. 2009a). In parallel, the ACT's expertise on GO and the island model had already indicated the potential of the island optimisation protocol to improve the performance of evolutionary optimisation studies. This strategic alliance of expertise and methodologies led to the definition of the following practical objectives for the whole study:

  (i)   To integrate current Mars Rover simulation software developed at Plymouth with the island model and PaGMO libraries for single and multi-agent robotic systems developed at ESA
  (ii) To carry out simulation experiments on single agent scenarios to investigate the contribution of the island model to the design of robust and general neural controllers for navigation and exploration in unknown environments. In particular, an island model is used here the population is split into separate islands to allow migration of single individuals, rathen thanallowing whole population to migrate.
  (iii) To carry out pilot simulation experiments on the contribution of the island model to potentiate the design of complex tasks such as the use of active vision strategies for the integration of local and distal information (e.g. landmarks) in navigation tasks.

In the report we will first describe the software development tasks as per first objective, that led to the successful extension of Plymouth Mars Rover simulation software, and its integration with the ESA PaGMO libraries to allow robotic experiments simulations using the island protocol.
Subsequently we will describe the experiments and main results of three experimental studies carried out during the project. The first two experimental studies (A on island comparison and B environment comparison) aimed at the evaluation of the island algorithm with respect to different population sizes and different environment configurations. The third study (C) focuses on the use of active vision strategies for complex navigation tasks.

2   SOFTWARE DEVELOPMENT

2.1   PaGMO

The Parallel Global Multiobjective Optimizer (PaGMO) is an open-source software developed by the Advanced Concepts Team of the European Space Agency to provide a flexible and friendly platform / library to perform massively parallel engineering optimization tasks (see Biscani et al. 2010). The software implements a 'generalized migration operator' between populations of solutions grouped in islands belonging to an archipelago. Tests made with maxi-archipelagos containing up to 50000 islands verified PaGMO reliability and thread efficiency. In PaGMO populations can be evolved (optimized) by means of any optimization algorithm and PaGMO will take care of parallelizing the computations by placing each population on a different operating system thread / process and of exchanging information between the various islands according to predefined migration strategies and routes. In the current PaGMO versions some popular algorithms are already implemented and provided to the user: A Simple Genetic, Differential Evolution, Particle Swarm Optimization, Adaptive Neighbourhood Simulated Annealing, Improved Harmony Search, Compass Search, Monotonic Basin Hopping, Multistart , Monte-Carlo, Bee Colony, Ant Colony, Firefly Algorithm, NSGA II. Interfaces to external optimization libraries are also provided and in particular: to the GSL library (includes Nelder-Mead, BFGS and more), to the NLOPT library to the IPOPT library and to the SNOPT library. In this particular project an early version of PaGMO is used (the current version is currently available from the original sourceforge repository but is not backcompatible). The simple genetic algorithm was selected as main algorithmic tool, modified and improved to make use of more sophisticated mutation and crossover techniques. The evolutionary robotics task was considered as a stochastic optimisation problem approximated via the Stochastic Sample Averaging technique (SSA) (see Ampatzis et al. 2009). A stochastic optimization problem is an optimisation problem where the objective function represents the expected value of some stochastic process. In the case of evolutionary robotics tasks the outcome **f(x,s)** of one simulation is typically stochastic as it depends on the sensory noise, the possible environment variability and the initial system conditions (all contained in **s**) plus, of course, the dependence from the chromosome **x** representing, in abstract terms, the artificial neural structure of an agent. In order to evolve good behaviours evolutionary robotic researchers typically demand to extremalize the expected value of **f(x,s)**:

$$f(x) = \mathrm{E}[f(x,s)] \approx \frac{1}{N} \sum_{i=1}^{N} f(x, s_i)$$

Approximated by its mean over a sample s_i. While apparently a simple relation, the equation above (called SSA) opens a number of far reaching questions typically overlooked by the majority of the ER community. What is the impact of the sample size on the resulting evolved behaviour? What is the impact of the chosen sample on the resulting evolved behaviour? In the island model, assuming each island maintains a different sample [s_i], when a migrated individual is inserted in a new island should he be compared to the 'indigenous' individuals using its original sample s?

This formal description of the evolutionary robotics problem allows a straightforward interface to PaGMO (see Ampatzis et al. 2009). In Figure 1a we show a simple illustration of the PaGMO principle (also mirroring the C++ class hierarchy). An archipelago is a collection of islands connected via a topology. An island contains a population and an algorithm. The population is a set of individuals with an evaluation metric defined by the optimization problem they are set to solve. A single call to the archipelago method evolve() starts the algorithms in each island on a different thread. At the end of each algorithm (asynchronously) information may get exchanged (migrated) between connected islands and the optimization on proceeds accounting for the new incoming information.
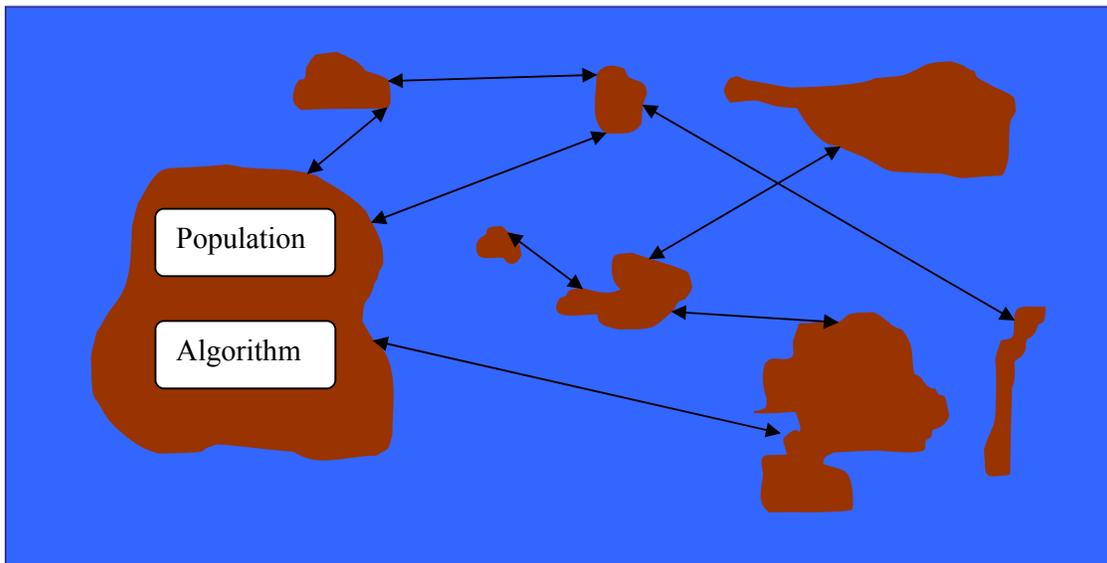

Figure 1a: Illustration of a PaGMO archipelago. Each island will be evolved in parallel. Arrows indicate the allowed migration routes as defined in the archipelago topology. The evolutionary robotics problem defines how population individuals are evaluated and thus the chance of their characteristic to be spread throughout the archipelago.

2.2 *Extension of Mars Rover Simulation Software and integration with PaGMO*

The integration of the Mars Rover Simulator, previously developed at Plymouth, with ACT's PagMO required a restructuring of its software architecture. In particular, the following main developments were carried out:

- Implementation of general controller that incorporates the functionality of island evolution through the PaGMO libraries as well as handling the configuration of simulation parameters,

such as the neural architecture, terrain, sensor configuration, graphics and environmental properties. This allows the parallelisation of the simulation with independent parallel processes corresponding to the test of one individual and its fitness calculation (Figure 1)

- Physics simulation module responsible for the execution of one behavioural test. The physics simulator executes the evaluation of neural controllers (i.e. genotypes) by deploying them in the environment and returning a floating-point number to the controller representing the achieved fitness of that controller, as defined by a fitness function. This module is based on the Open Dynamics Physics engine (www.ode.org). An optional OpenGL graphics rendering engine permits the visualisation of the robot and it's environment (Figure 2)

- Design of GUI (Figure 3) for the systematic execution of simulation experiments and the setting and control of the parameters for the island protocol (e.g. number of islands and migration strategies), genetic algorithm (e.g. population size, mutation rate), rover configuration and neural control systems (e.g. input/output nodes, connection sets) and environment (e.g. type/number of environments). This GUI also plots the fitness charts during simulation execution.
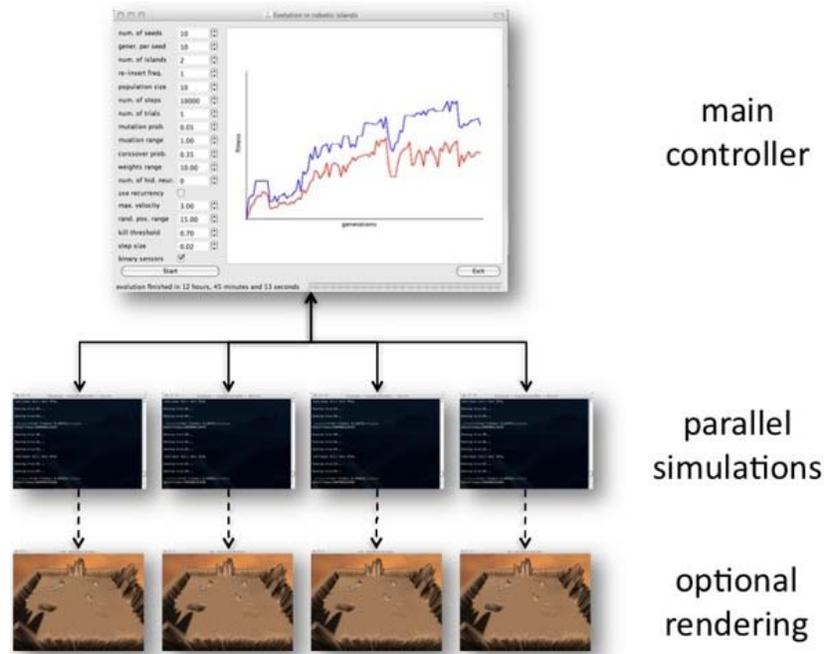

Figure 1b: Architectural diagram showing the relationship between the controller and physics simulator
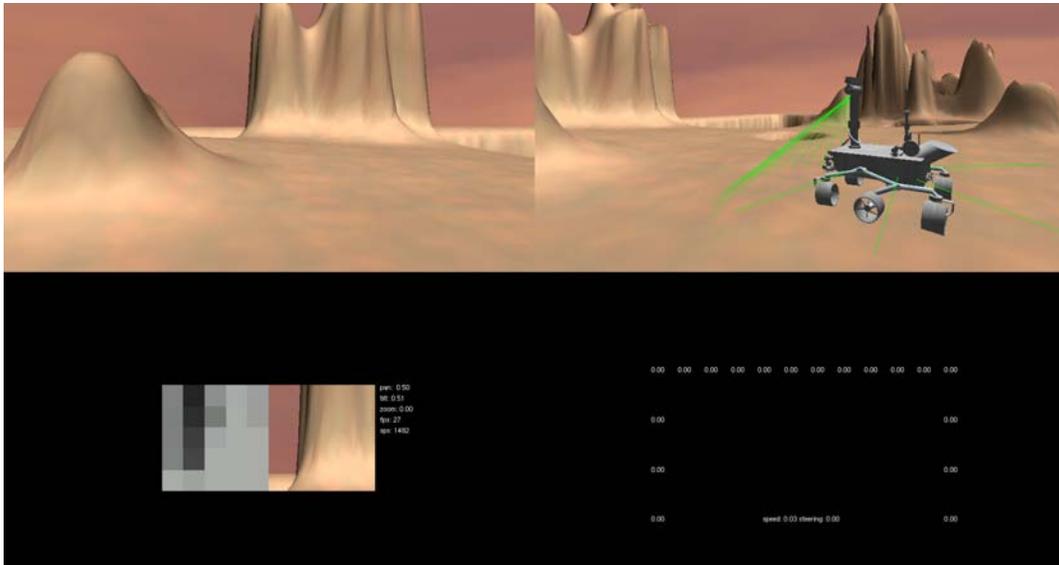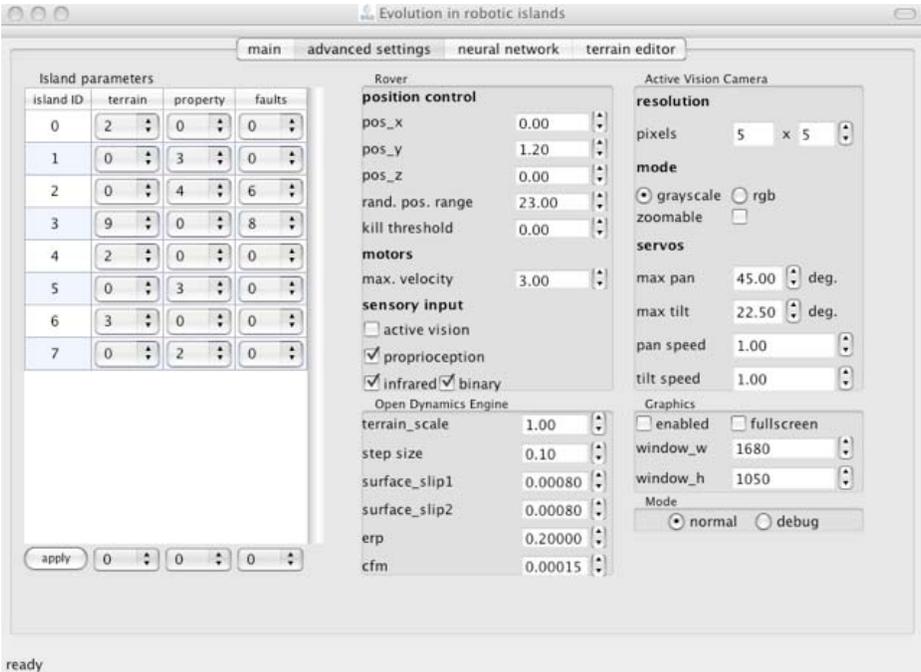
Fig. 2. Physics simulation of the Mars rover. The right section shows the user-controlled camera, the rover and the sensor inputs. The left section shows the rover's field of vision and information from the active vision system when in use.
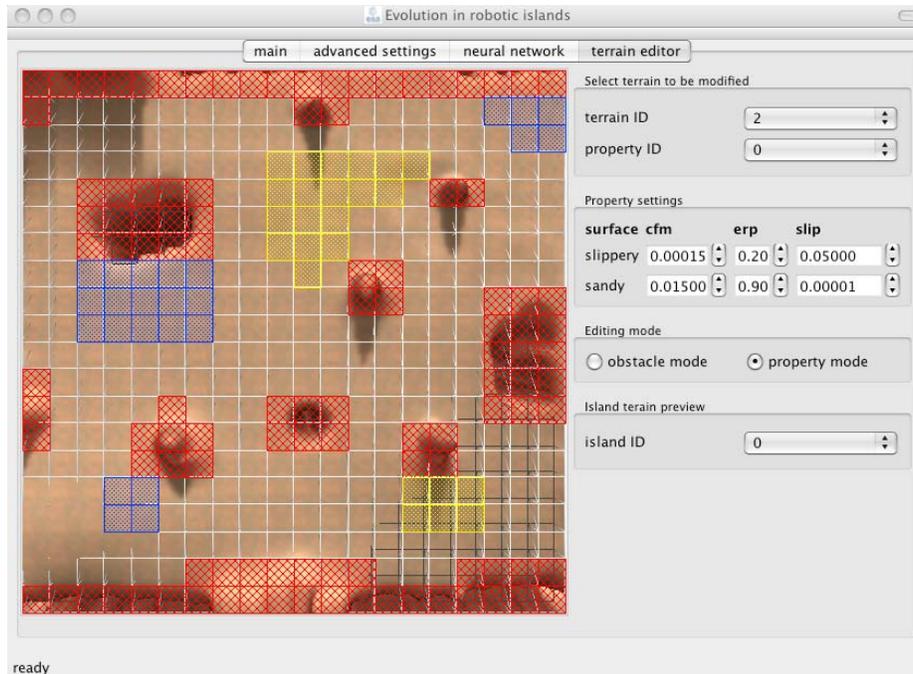
Figure 3: Example of screenshots of the GUI to control and start simulation experiments (top) and to modify the environment (bottom).

2.3 *Software repository*

All the software developed in the project has been made available open source under the Sourceforge,net page *MarsRoverSim*.
http://sourceforge.net/projects/marsroversim/
Details of all simulation experiments are available in the project wiki page http://marsroversim.wikkii.com/ .

3   EXPERIMENTAL STUDY A: ISLAND COMPARISON

The first experimental study directly addressed the evaluation of the the contribution of the island model to the design of robust and general neural controllers for navigation and exploration in unknown environments. This used a single-agent scenario within the simulated Mars rover setup. The study explicitly compares the performance of simulations using the island protocol versus other simulations using standard evolutionary computation protocols. Specifically, we decided to study the island model by splitting the population into separate islands to allow migration of single individuals. This is an extension of previous studies, where instead whole populations that were evolving separately were also allowed to migrate (Ampatzis et al., 2009).

3.1 *Description of experiments[i]*

Island model setup
In the study, an archipelago, i.e. the chain or cluster of islands with specific migration routes between them, was used consisting of 8 separate islands where each island contained 10 individuals. Every

island had the same environment (see Fig.7) but unique individuals, each having been initialised with a different random seed. Therefore, the overall size of the archipelago was 80 individuals. The islands from within the archipelago were evolved independently, receiving migrants only at certain intervals, as defined by the migration rate, at which point the best individual from each island move to a different island. In this study the migration rate was set to 5, which allowed individuals to migrate between the islands every 5th generation. The feasible migration paths were given by a particular topology. The island model framework supports a variety of such topologies including chain, ring, cartwheel, ladder, hypercube, lattice and broadcast topologies. For this
experiment the ring topology (see Fig.4) was utilised, simply because the number of islands was not high enough to experiment with the effects of topologies in this particular task.
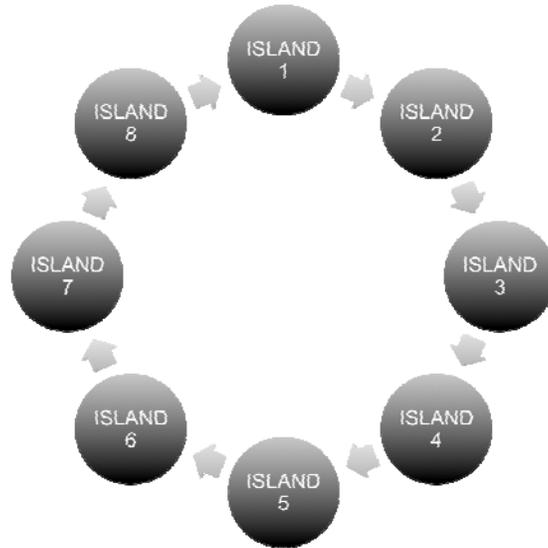

Figure 4: Ring topology for the island simulations

Each island was evolved in a separate thread and managed by a genetic algorithm. The population size of each island was set to 10 individuals. Only the best 2 individuals were allowed to produce 5 offspring each. Mutation and crossover operators subsequently acted on these offspring; a mutation occurred with the probability of 10% by adding to the original gene's value a quantity in the range [-1, 1], while crossover was exponential, happening with a probability of 95%. The best individual of the previous generation was retained unchanged and replaced the worst of the 10 offspring (often known as elitism). In this way it was possible to produce a new population of 10 individuals that inherited their genes from the best individuals of the previous generation.

The whole evolutionary process lasted 100 generations. In each generation, each control system was evaluated 10 times by deploying the rover in the environment (randomly positioned and rotated) and allowing it to act for up to 3000 sensory-motor cycles, that is to say, 3000 activations of the ANN. However, this was not always the case, as the evaluation of a particular genotype is terminated when a rover falls into a hole. 20 evolutionary runs were conducted, where each population was initialised with a different set of randomly generated individuals, spread across 8 separate islands.

The performance of each control system was evaluated according to the fitness function (see 2) that was carefully designed to shape the behaviour of the robot for effective and reliable exploration and obstacle avoidance behaviours:

$$F = \frac{1}{S \times T}(Sp \times St)$$

where the fitness F is a function of the measured speed Sp and steering angle St, where Sp and St are in the range [0,1]. Speed Sp is 1 when the rover goes at the maximum speed and 0 when it does not move or goes backward. Steering angle St is 1 when wheels are straight and 0 when they are turned over an angle of 30° from the centre. If for example the angle was 15° then St would be 0.5. T is the number of trials (10 in these experiments) and S is the number of sensorimotor cycles per trial (3000 in these experiments). The equation above shows how the fitness is calculated at every sensory-motor cycle. Thus, the GA has to maximise the fitness by increasing the value of Sp and St, which implies that a rover has to move at a maximum possible speed while steering only when necessary. If a rover goes forward at the maximum speed but keeps the steering angle over 30° then its final fitness will be 0. Similarly, if a rover goes backwards or does not move at all, its fitness will also be 0 regardless the steering angle. The maximum fitness contribution at each time step is therefore $1/S*T$. The final fitness of each individual is in the range [0, 1] and it is the sum of all contributions from all time steps of all trials.

In order to evolve a good controller, it was necessary to create a suitable environment to allow the robot to experience different surface conditions (see Fig.5). The environment that was modelled for this purpose is an arena of 60x60m and contains inclined and declined surfaces, three high and three small rocks, holes and rough areas. $111m^2$ of the terrain is covered by obstacles and hence not traversable.
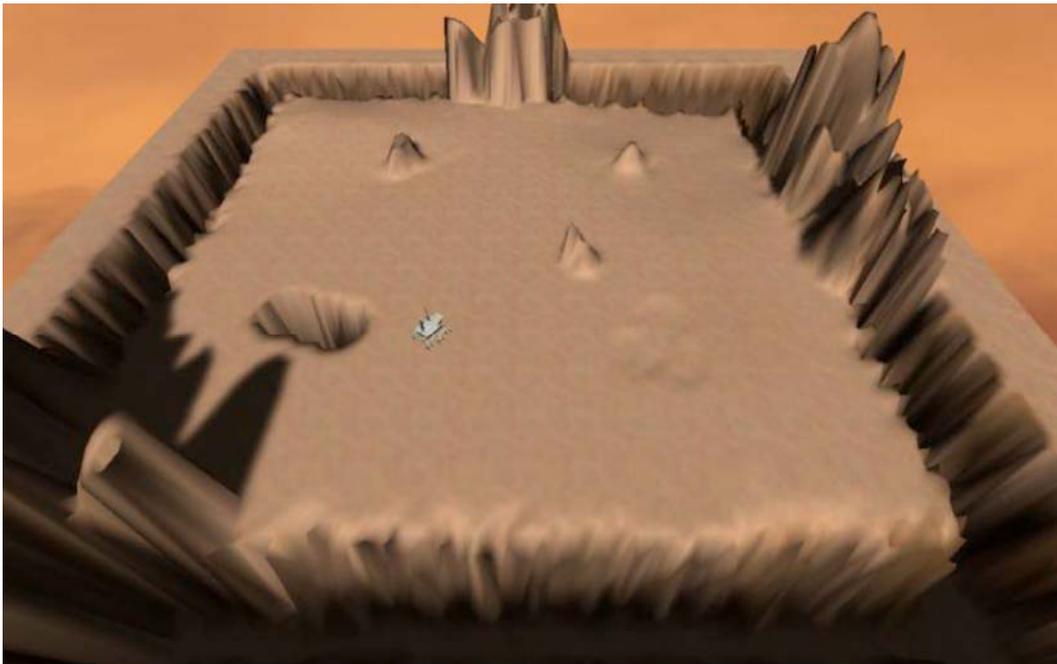

Fig. 5. Environment that was used during all evolutionary runs.

Rover Model
The robot used in this experimental study is a 3D simulation model of the MSL rover. The model cannot be considered an accurate or detailed representation of the actual rover, but only an approximate copy. This is mainly due to the lack of information on the rover's real dimensions, mass distribution and parts size, as well as many other details. According to the Centre National d'Etudes Spatiales (CNES 2009), the dimensions of the real rover are 2900Lx2700Wx2200H mm and its mass is about 775 kg. The physics model of the rover was therefore built considering these details, and modelled on the several diagrams and pictures that were available. These limitations are not crucial to the study at this stage as the focus is to demonstrate the application of the ER approach in developing a suitable controller capable of performing complex obstacle avoidance tasks in unknown rough terrains.

The motor system of the rover model (see Fig.6) consists of six wheels where the two front and the two rear wheels are able turn up to 90° to either side. The rover is capable of overcoming obstacles that are approximately the same size as its wheels. This is possible thanks to a rockerbogie suspension system. This advanced suspension system is designed to operate at low speeds and consists of two pivoted joints connecting two bogies with two rockers (Miller & Lee, 2002). These rockers are connected together via a differential join. This means the left and right part of the rocker-bogie system can move independently while keeping the main body level.

The rover is equipped with the sensory apparatus to process 18 infrared sensors, which are used to provide information about the surrounding environment. Two different sets of sensors are used to accommodate obstacle detection. The first set consists of six lateral sensors, which provide extra safety when the robot approaches obstacles from the side. These sensors have a range of three meters and are not able to detect holes. The lateral sensors cover an area of approximately 200° around the rover, leaving the front area deliberately uncovered. These sensors return either 0 (no obstacle) or 1 (obstacle present) when activated by the presence of an object within the activation range of the sensors.

The second set consists of 12 infrared sensors with a maximum range of five and half meters. These infrared sensors, which shall be referred to as ground sensors, are positioned on the rover's camera mast and point downward at a 45° angle, reaching the ground approximately three meters in front of the rover. The twelve sensors are positioned and directed to ensure the range extends to around 400 mm beyond ground level. Ground sensors constantly scan the distance from the surface and are able to detect both rocks and holes. Each of these sensors returns a floating point value from 0 (no feedback) to 1 (strongest feedback). Holes or cliffs can be detected by the rover when it loses sensory feedback from the ground (i.e. ground sensor returns 0). The same sensors allow the robot to detect dangerous rocks or excessively rough terrain. This is achieved thanks to a particular threshold. When the activation of a sensor reaches that threshold it indicates that the robot is facing an insurmountable rock or a potentially dangerous terrain roughness. If a sensor's output goes over this threshold (a rock) or returns 0 (a hole) then its output value is changed from 0 (not active) to 1 (active). On the other hand, if the returned value stays within a certain boundary, which is given by the threshold, then the sensor returns 0. From this perspective a 0 activation can be seen as a safe zone and 1 as an obstacle. To model the lateral sensors and the ground sensors the researchers aimed to simulate the existing infrared sensors Sharp 3A003 and Sharp 0A700, respectively. In previous experiments (Peniak et al. 2009a) the threshold, which can be in a range [0,1] was co-evolved with the neural network weights to a near-optimal value. In addition to the above sensors, the rover is provided with an active vision system (not used in the experiments reported in this study) and two internal sensors measuring its speed and steering angle.
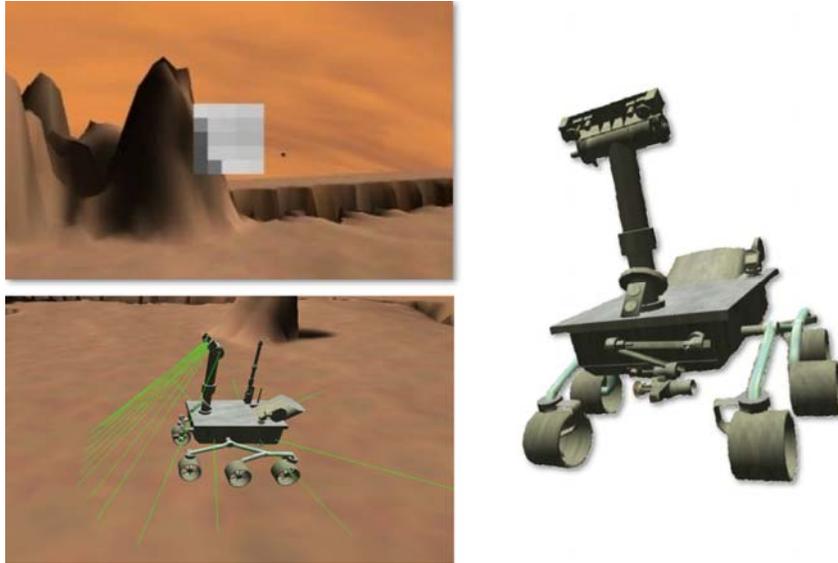
Figure 6: 3D physics model of the rover showing the different parts of the rocker-bogie suspension system (right) as well as the vision system (top-left) and the position and orientation of the 18 infrared sensors (bottom-left). The vision system consists of a 5x5 matrix of foveal cells whose receptive fields receive input from a greyscale image of a limited area (100x100 pixels) of the whole image.

Neural controller

The control system is a fully-connected, discrete time, feedforward ANN (perceptron) with evolvable bias (see Fig.7). A set of 18 sensory neurons receive activation from the 18 infrared sensors of the rover, while an additional set of 2 proprioceptive neurons encode the value returned by the internal sensors, providing information about the speed and the position of the wheels. The 20 sensory neurons are fully connected to 2 motor neurons that modulate the level of the force, which is applied to the actuators, directly responsible for rover's speed and steering. Motor neurons have the sigmoid activation function:

$$f(x) = \frac{1}{1 + e^{-x}}$$

in the range [0, 1], where x is the weighted sum of the inputs minus the bias. Biases are implemented as a weight from an input neuron with an activation value set to -1. The ANN does not have a hidden layer as the authors' previous experiments showed that it was redundant and did not help to achieve higher fitness. This simple architecture greatly reduces the computational demand of the control system, which is an important asset when considering a planetary rover, where the computational resources have to be kept to a functional minimum.

The rover's actions depend on the values of the synaptic weights of the ANN. Each weight must be set to an appropriate value to produce a desired output and, as mentioned previously, a genetic algorithm is used to evolve these. The free parameters that constitute the genotype of the control system and that are subject to evolution consist of: 42 synaptic weights (the 40 synaptic weights that connect the 20 sensory neurons to the 2 motors neurons, plus the 2 biases) and a single gene which encodes the threshold applied to the ground sensors. Weights and biases are encoded as floating point values in the range [-1, 1] and the threshold in the range [0, 1].
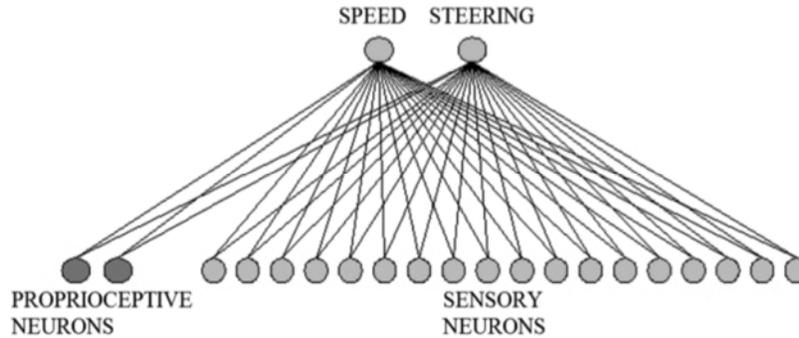
Fig. 7. Feed-forward neural network used as a control systems for the rover in the evolutionary experiments.

3.2   *Main Results*

The experimental setup involved using both the island model as well as the standard sequential approach in order to evaluate the quality of the evolved solutions produced by each model. Twenty evolutionary runs were conducted for each model and the results showed that an effective behaviour emerged throughout all the runs. In particular, due to the general behaviour optimised by the fitness function, the obtained controllers were able to navigate the environment with a certain degree of efficacy, capable of avoiding obstacles of different types and dealing with rough terrain.

The chart in Fig.8 shows the average results of the twenty evolutionary runs for each model. The graph was created by averaging values from all of the twenty runs. The black lines in the graph represent the standard approach and the grey lines the island model. The full lines show the maximum fitness obtained by the best individuals, while the dashed lines show the average fitness for all the populations. By looking at the graph it can be noticed that the island model achieved slightly higher fitness that the standard model. However, the main advantage in this particular case is the significant time reduction in completing the evolutionary process. In particular, this time decrease is proportional to the number of processors employed by the island model.

In these experiments eight separate islands were running in parallel and hence the overall time necessary for finishing 100 generations (approximately 5 hours) was 8 times less than using the sequential approach (almost two days). This way it is possible to split populations across any number of processors and reduce the time significantly while not losing anything in the quality of the solutions.

Using both models the behaviour that emerged was very similar and quite simple. Since the fitness function requires the rover to go as fast as possible while keeping steering at a minimum, the rover evolves a strategy whereby it preferentially travels in straight lines, turning only when encountering obstacles. If the rover detects a rock, for example, it will immediately turn around and continue in a straight trajectory, however, the rover will only steer as much as is necessary to circumnavigate the obstacle.
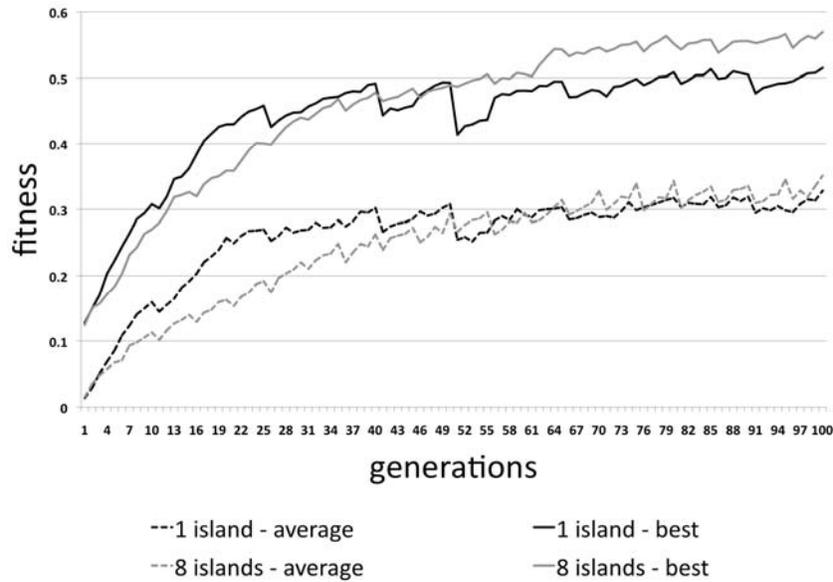
Fig. 8. Averaged fitness values from all the twenty evolutionary runs for each model. The black lines in the graph represent the standard approach and the grey lines the island model. The full lines show the maximum fitness obtained by the best individuals and the dashed lines show the average fitness for all the populations.

3.3   *Additional experiments*

The experiments described above report the main island comparison study, which was presented at the International Joint Conference on Neural Networks (IJCNN-WCCI 2010), held in Barcelona on 18-23 July 2010. The project however included a series of additional experiments to test various comparison conditions and population sizes, all of which are reported in detail in the project wiki page. Here we give an overview of these additional experiments.

One initial pilot study investigated variations of the evolutionary algorithm parameter set. For example, the chart in Figure 9 gives an overview of performance when varying the selection method (roulette wheel and elitism with 20% best selection) and mutation method (Gaussian, bounded and strong mutation – where the mutation value is unbounded). Following this pilot study, all of the follow-up experiments where based on the Elitism selection strategy. The majority of experiments used the bounded mutation method, whilst for a few it was necessary to use the Gaussian mutation (active vision study)
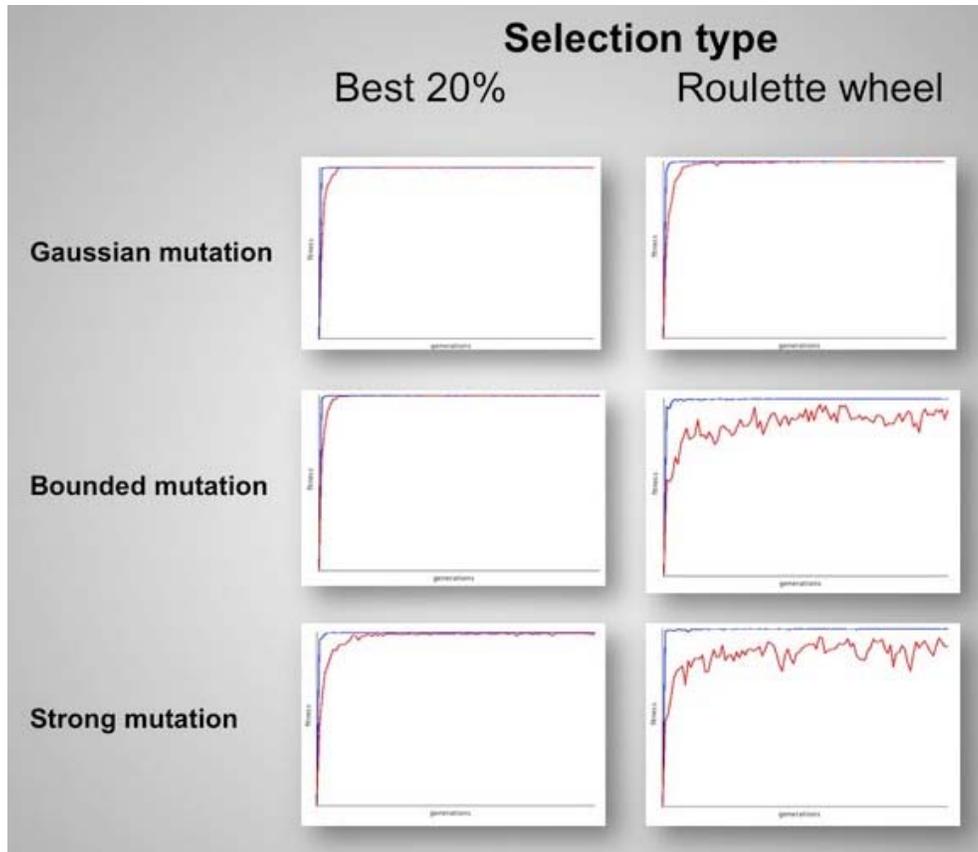
Figure 9: Pilot experiments to test effects of selection different strategies and mutation methods.

The initial island comparison study focussed on a maximum population size of 80 individuals in three different configurations: one island of 80 individuals, 8 10-individual islands with no migration and 8 10-individual islands with migration. Results in Figure 10 show that for the maximum fitness the standard island migration approach (red lines) leads to higher fitness performance.
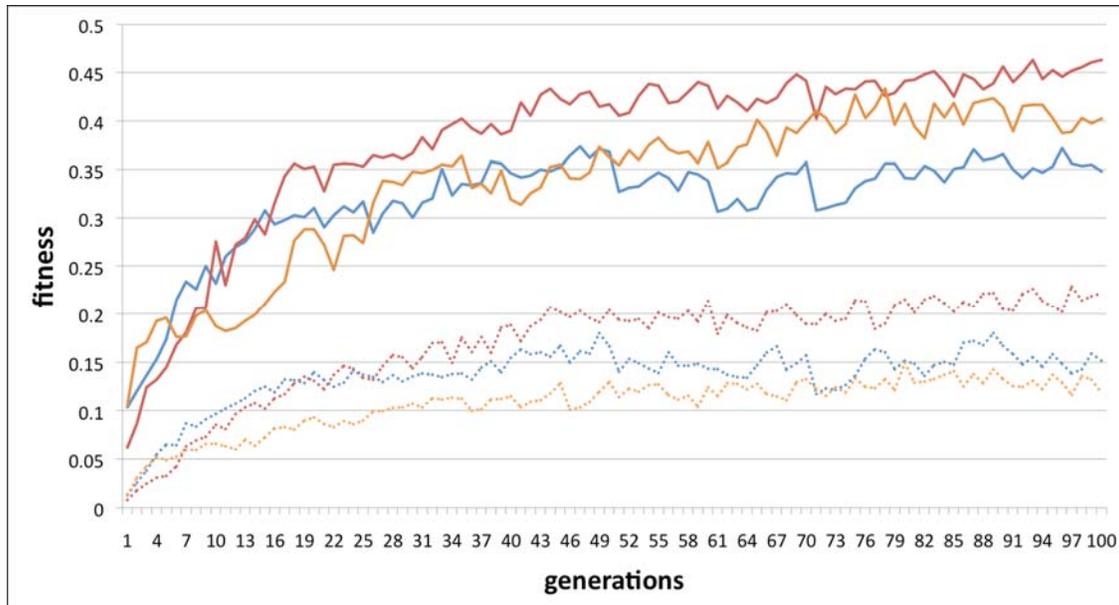
Figure 10 Average of 4 replications for the maximum values obtained with the same settings with no migration (purple) and one island of 80 individuals (green). Blue lines = 1 island with 80 individuals; Red lines = 8 islands with 10 individuals each; orange lines = no migration, 8 islands with 10 individuals each. Dashed lines show the average and full lines show the maximum fitness.

To understand better the performance of the island model with different population sizes, the same experiments where carried out with three maximum population sizes:

- 80 individuals respectively as one island of 80 rovers, 8 islands of 10 rovers each with no migration and 8 islands of 10 rovers each with migration
- 160 individuals respectively as one island of 160 rovers, 8 islands of 20 rovers each with no migration and 8 islands of 20 rovers each with migration
- 240 individuals respectively as one island of 240 rovers, 8 islands of 30 rovers each with no migration and 8 islands of 30 rovers each with migration

Figure 11 gives a boxplot overview of the results comparing these 9 conditions. Statistical comparisons of the difference between the three conditions in the varying population sizes, through the ANOVA test, indicate that: (i) For the 80 individuals experiment, the average fitness of one island and 8 islands with migration have statistical significant higher fitness than the islands with no migration; instead for the best fitness, the 8 islands with migration are better than both 1 island and 8 islands no migration. (ii) For the 160 individuals experiments, the average fitness of the 8 islands with migration is better than 1 island, which is then better than 8 islands no migration; the best fitness comparison shows that the 8 islands with migration is better than 8 islands no migration, which is then better than 1 island; (iii) in the 240 individuals case, the average fitness shows no statistical difference between any of the conditions, although 8 islands with migration has a tendency to significance over 1 island (p 0.052), and the best fitness show that the 8 islands with migration and 1 island are both better than 8 islands no migration.

Overall, this study demonstrate that the comparison of the average (thick middle line) data show that the advantage of the standard island experiment with migration is higher that the other two conditions consistently for the 80 and 160 population size and that the island migration advantage reduced with larger population sizes.
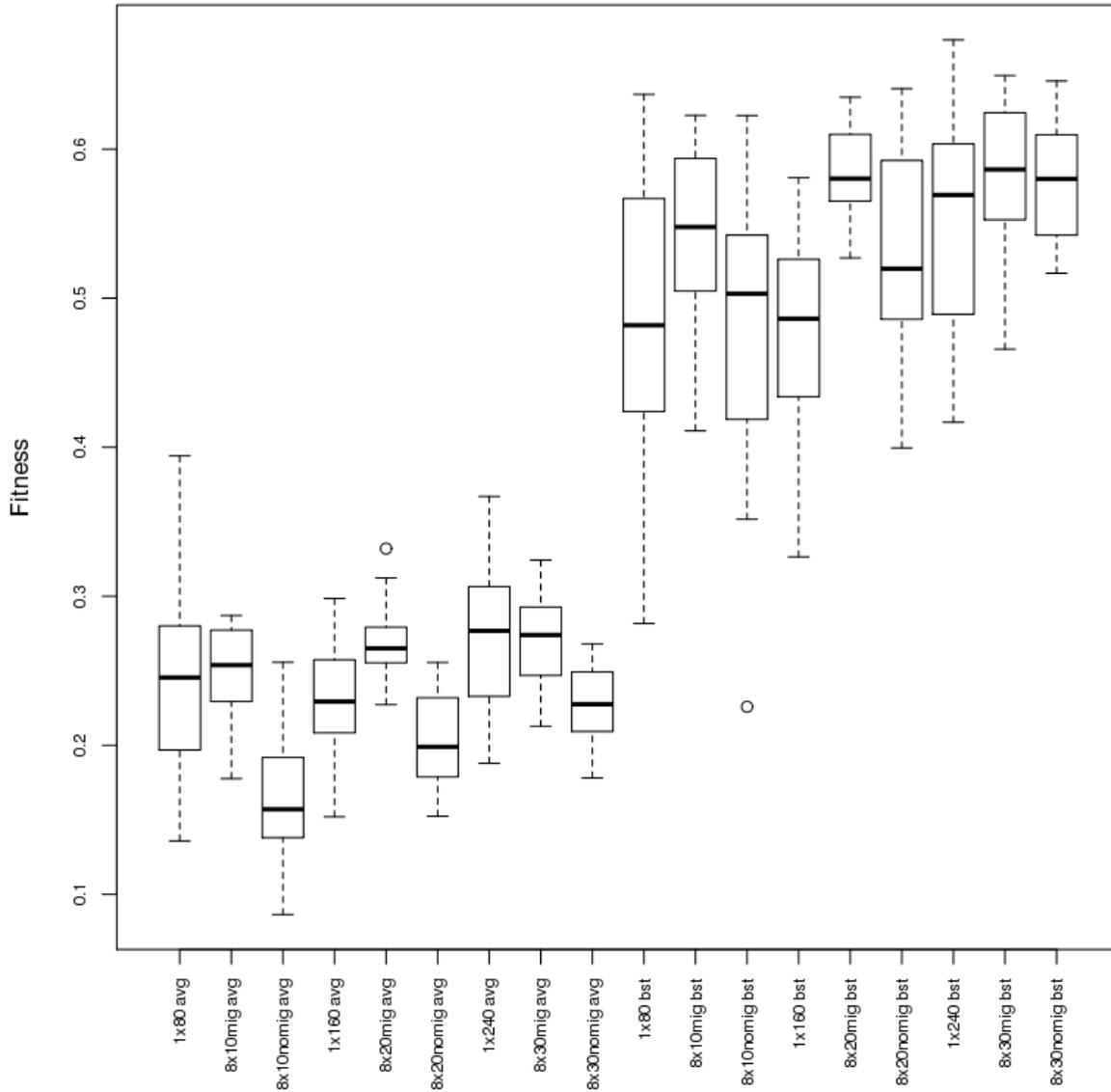
Figure 11: Results for the comparison of max population sizes of 80, 160 and 240 overview for the 9 conditions. Each boxplot indicates the average value (thick horizontal line), the max and min range and the 50% central percentile.

As a general conclusion of this first set of experiments in section 3, the comparisons of the island vs no island conditions show that although we observe some advantages (e.g. higher average/max fitness) of the migration protocol with respect to standard GA setups, these improvements in performance are not systematic. For example, the comparison of Figure 11 indicated that the fitness improvement for the max fitness disappears for higher population sizes. This could be explained by the fact that when there are larger populations, a standard GA benefits from the search of good solutions by trying a high number of genotypes. The island model instead compensates the smaller population size search by allowing migration of genotypes between islands.

4 EXPERIMENTAL STUDY B: ENVIRONMENT COMPARISON

4.1 *Description of experiments*

This experimental study specifically aims at the testing of the island model for the generality and robustness of navigation strategies in different environment configurations.

The island model is based on virtual archipelagoes, defined as chains or clusters of islands (subpoupulations) with specific migration routes between them. In this study an archipelago was constructed consisting of 9 islands, each with 10 randomly initialised individuals (every island having a different seed), giving a total population of 90 individuals, all of which were evaluated in the same environment containing 2 rocks and 2 holes (Figure 12 top). Within PaGMO, feasible migration paths are given by particular topologies, examples of which include chain, ring, cartwheel, ladder, hypercube, lattice and broadcast topologies. For this experiment the ring topology was utilised with migration commencing every 5th generation. (Figure 13).

All of the islands were evolved concurrently using a genetic algorithm, each island having a population size of 10 individuals with the best 2 individuals producing 5 offspring at each generation. Mutation was subsequently applied to these offspring with a 5% probability of adding a value to the original gene in the range [-1; 1]. The best individual of the previous generation was retained unchanged, replacing the worst of the 10 offspring (known as elitism). During each generation, all of the genotypes were evaluated 10 times for 3000 sensory-motor cycles (i.e. 3000 activations of the ANN), each time initialising the rover with a different starting position and orientation. This whole process was repeated for 100 generations, with 10 replications being conducted in total, starting each time with a different set of randomly generated individuals distributed across the 9 islands.

The performance of each control system was evaluated according to the fitness function, as in island comparison simulations, designed to shape the behaviour of the robot for effective, reliable exploration and obstacle avoidance behaviour.

To test the robustness and the generalisation capabilities of the final solution, two additional complex environments were used, the first featuring inclined and declined surfaces, three high and three small rocks, rough areas and holes. 111 m^2 of the terrain was covered by obstacles and hence not traversable (Figure 12, middle). The second environment consisted of a canyon with a cliff on either side. All environments used were 60 m^2 in area (Figure 12, bottom).
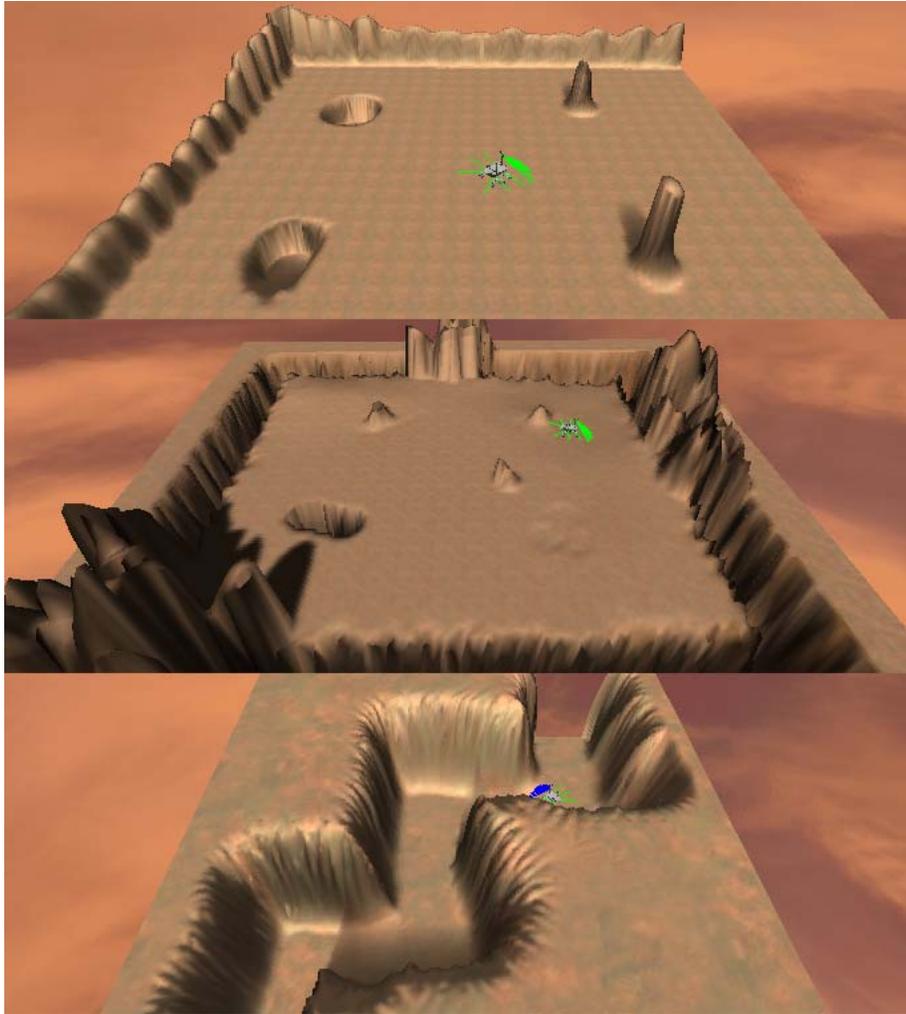
Figure 12: Bird's-eye view of the three environments: Environment used during evolution (top). Test Environment 1 (middle), Test Environment 2 (bottom).
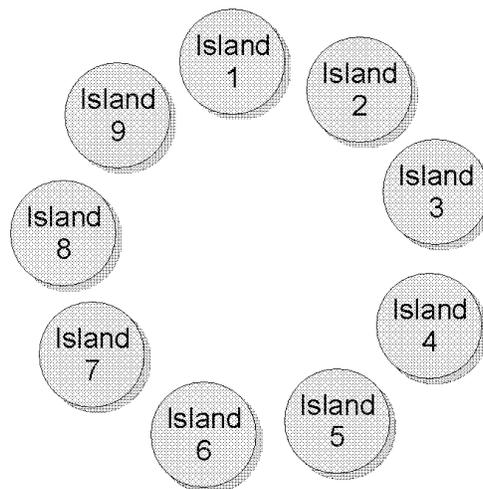

Figure 13 – Ring migration protocol for the 9 islands.

4.2   Main Results

A set of simulation experiments was conducted using the above set up to test the ability of the island model to produce suitable controllers for planetary rovers in different environments. As an experimental control, a set of optimisations were carried out using the traditional, single population approach. To ensure parity, both approaches used the same evolution parameters and evaluated the same number of individuals (90 individuals x 100 generations). It was found that both approaches produced controllers capable of navigating in unknown environments, avoiding obstacles of different types.

Ten replications were performed for both approaches. The averages of all replications can be seen in Figure 14, and show that the island model consistently achieves higher maximum fitness across all generations, with an ANOVA statistically significant difference when tested at the final generation ($p = 0.05$). The main advantage to the island model, besides from achieving higher maximum fitness, lies in its ability to parallelise the evolutionary process, with a time decrease directional proportional to the number of processors employed. In this case, the population was split over 9 islands with each island running on a separate core, effectively reducing the total optimisation time by a factor of 9. This result empirically proves that the evolution of neural controllers is capable of being parallelised using the coarse-grained approach of the island model and that the migration operator can improve the evolution process, confirming an important result in heuristics continuous optimisation. The behaviour of the controllers is consistent with what was expected from the fitness function — rovers travel in a straight trajectory, steering only when it is necessary to avoid obstacles.

To test the robustness of the solutions, the final individual from each replication was evaluated for an additional 100 trials in the evolution environment (see Figure 15) as well as two new environments, previously unseen by the controller (see respectively Figure 16 left and right). As is shown in the plots, the island model solutions on average outperform those from the single population, and perform much better relative to the traditional approach when presented with a new and more complex environment.

These simulations have empirically demonstrated that the island model can produce neural controllers for planetary rovers, capable of navigating and avoiding obstacles in unknown environments, while using a continuous value sensor system. It has been shown that this approach can achieve equivalent or better results than the classical evolutionary approach, while drastically reducing the time required for a solution.
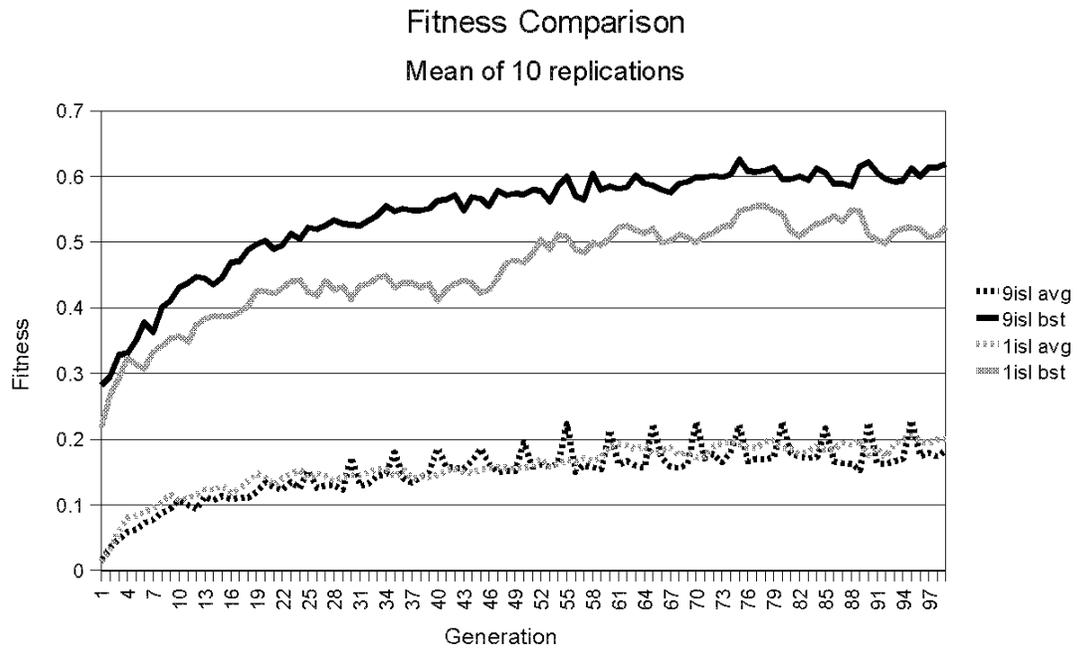
Figure 14. Average fitness during evolution. The island model is plotted in black and the standard approach in grey. Solid and dashed lines represent the mean of the maximum and average fitness, respectively.
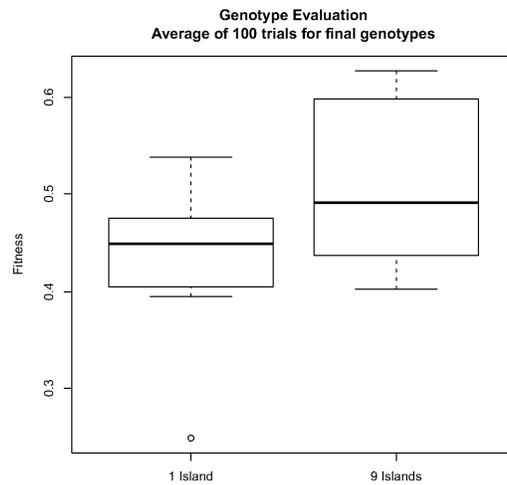

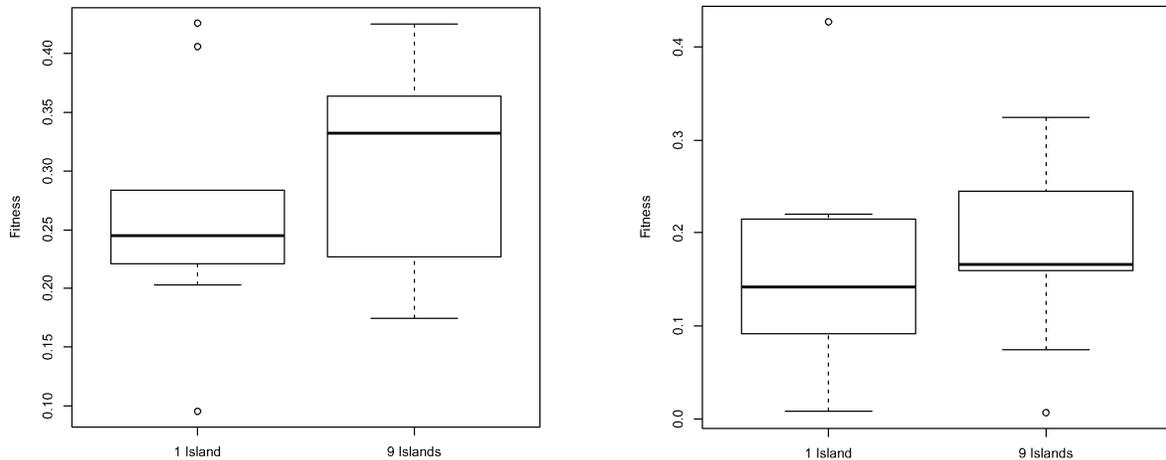Fig. 15. Evaluation of final genotypes, postevolution

Fig. 16. Evaluation of final genotypes in 1st (left) and 2nd (right) new environments

5 EXPERIMENTAL STUDY C. ACTIVE VISION

5.1 *Description of experiments*

Having successfully established the advantages of the island model in previous studies, the purpose of this study was to evaluate the model's scalability and potential when applied to more complex problems. To conduct this investigation, a task was designed requiring the interaction of several complex abilities, including active vision for obstacle avoidance and landmark navigation, and proprioception for terrain classification.

Active vision is based on the theory of active perception (Regan & Noe, 2001) and consists of exploiting the regularities arising from sensory-motor interactions with the environment, rather than constructing and analysing internal 3D representations of the environment. Previous work has demonstrated that this control strategy can be used successfully with the evolutionary robotics approach to evolve robots capable of performing simple obstacle avoidance (Peniak et al. 2009b) nce. To see whether this could be extended further with the aid of the island model, a preliminary experiment was devised. An environment was constructed consisting of three ravines, each with a different simulated surface property - one with the default terrain property, one with sand, and one with ice - with an illuminated landmark placed on one side (Fig. 17). The purpose was to see whether the island model could optimise controllers to utilise active vision strategies for landmark navigation and obstacle avoidance, while using proprioception to detect and avoid adverse terrains.
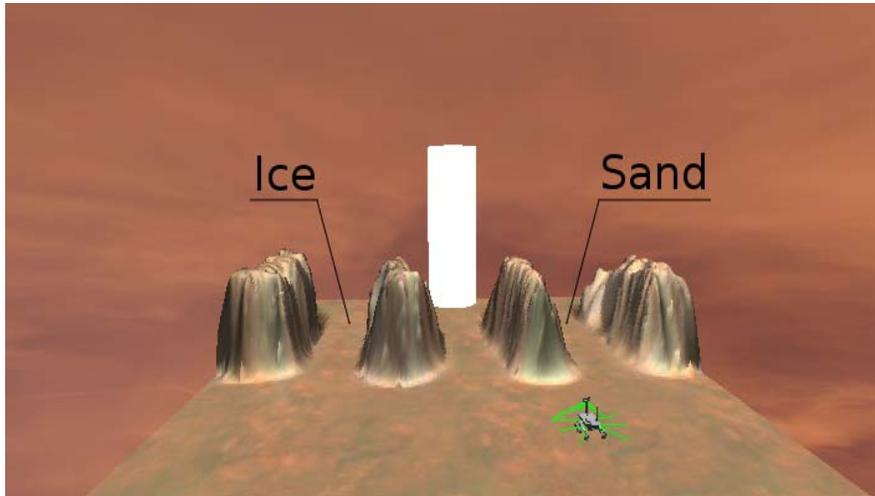
Figure 17: Environment used in the preliminary investigation of active vision, showing the three ravines and their terrain properties. The illuminated landmark can be seen in the middle.

The control system of the robot consisted of a multilayer continuous-time recurrent neural network (CTRNN) – a type of continuous neural network with a self-recurrent connection on each unit, modelled with the equation:

$$\frac{1}{\tau}\left(-y_i+\sum_{j=1}^{N} w_{ji}\,\sigma(y_j+\theta_j)+I_i\right)$$

CTRNN equation, see Beer (1995)

where y is the state of each neuron, τ is the time constant (in this case, fixed at 1000 for all hidden units), w is the connection weight from the ith to j^{th} neuron and θ is the bias, both in the range [-10; 10], I is a constant external input, and σ is the sigmoidal function:

$$\sigma(x) = 1 / (1+e^{-x})$$

The continuous, recurrent nature of this network affords a dynamical memory, allowing for the exploitation of temporal input components, such as those necessary to detect sudden changes, as might happen when moving onto a new terrain type.

The input to the network comprised of speed and steering proprioception and grey-scale visual input from a 5x5 pixel camera, statically mounted on the rover's mast (Fig. 18). These inputs were fully connected to a hidden layer of 5 neurons, which was in turn fully connected to 2 output neurons, directly responsible for controlling the rover's speed and steering (Fig. 19).

Figure 18: Robot's visual field with 5x5 foveal region (left)
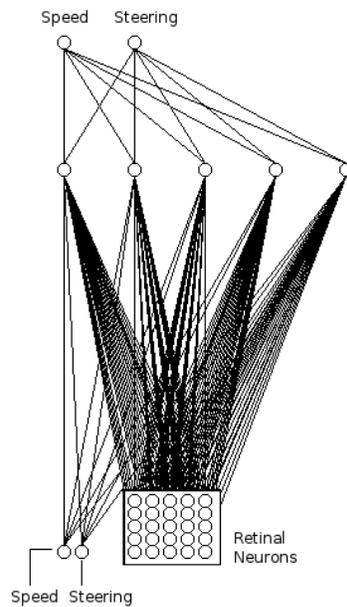and view of the robot's camera and mast (right).

Figure 19: Architecture of the CTRNN used for the rover controller systems, consisting 25 visual inputs, speed and steering proprioceptive input, five hidden units, and speed and steering output.

The island architecture used in the experiment contained 16 island connected in a ring topology, with migration occurring every 5th generation. Each subpopulation contained 20 individuals, giving a total population of 320 individuals, which was evolved for 100 generations. At every generation, the best 4 individuals produced 5 new offspring, with a mutation probability of 10%. The mutation range for each selected gene was based on a Gaussian distribution in the range [-1; 1]. The best individual of the previous generation was retained unchanged, replacing the worst of the offspring (known as elitism). Each evaluation lasted for 9000 sensory-motor cycles.

The experimental set up under which the controllers were evolved involved initialising the rover in a random orientation at one of six predefined starting positions on the side of the ravines opposite the landmark and evaluating their performance according to the fitness function:

$$F = (D \, / - D_{max}) + 1$$

where D_{max} is the maximum distance from the landmark possible for the terrain, and D is the rover's linear distance from the landmark ($D \geq 1$), as given by:

$$D = \sqrt{(x_r - x_l)^2 + (y_r - y_l)^2}$$

where x_r, y_r and x_l, y_l are the Cartesian coordinates for the rover and landmark respectively.

5.2 Main Results

Five replications of the above experiment were conducted, resulting in controllers which exhibited the desired behaviours of terrain and obstacle avoidance, as well as landmark navigation. The mean of the average and maximum fitness over all 5 replications is plotted in Fig. 20, along with the standard deviations.
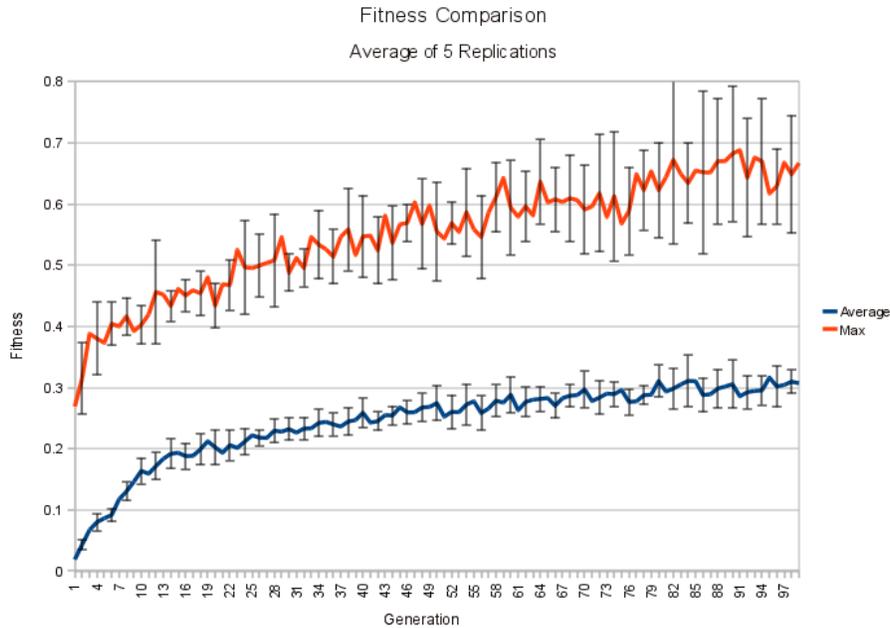

Figure 20: Mean of the average and maximum fitness accumulated from 5 replications, as shown for each generation during evolution. The red line (top) shows the maximum fitness, and blue line (bottom) the average. Error bars show the standard deviation across replications.

As the fitness function explicitly rewarded rovers who were able to get nearest to the landmark, solutions emerged which had a propensity to avoid terrain features which hindered this aim (i.e. ice, sand and cliffs). It should be reiterated that this study was a preliminary one and given the complex nature of the problem the derived solutions often exhibited erratic and suboptimal behaviour, as can be seen from the path taken in one trial shown in Fig. 21; however the objectives of visual landmark navigation and obstacle avoidance were still achieved, highlighting potential in the approach and encouraging further investigation.

As the rovers used a static camera in this initial study, one explanation posited for the circular trajectories is that rovers continually turned in order to assess their environment. To test this hypothesis, the next planned investigation will involve repeating the experiment with an actuated camera to see whether controllers retain this behaviour, however including a term in the fitness function to reward linear trajectories should also resolve this.
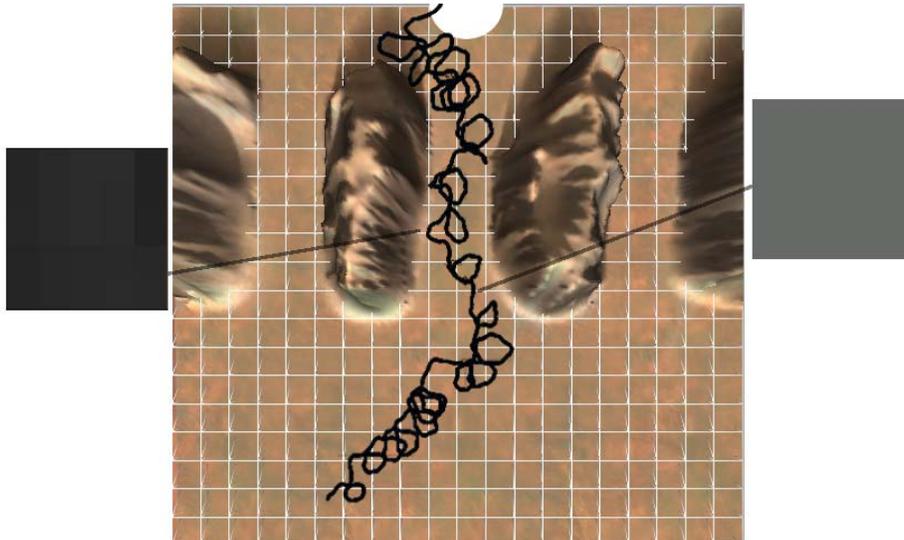

Figure 21: Example trajectory taken by one of the evolved individuals. The squares on either side illustrate the camera inputs at two of the points along the path – showing the visual input received for an avoidance manoeuvre (left), and landmark navigation behaviour (right).

Analysing the camera input at different points along the path travelled in Fig. 21, the emergent behaviour can be better understood. For example, in the left camera image, an obstacle avoidance manoeuvre can clearly be seen to occur when the visual input is mostly black, a result of the rover facing a wall. Likewise, in the left image when the rover is directly facing the landmark (which is engulfing the visual input), the rover travels towards the source.

Ongoing work is looking to tune the evolutionary parameters and fitness function to better optimise the resulting behaviour, as well as using environments with more complex ice and sand configurations to explicitly investigate the role of proprioception.

6   PROJECT MANAGEMENT, STAFF AND RESOURCES

The Plymouth project supervision staff were Angelo Cangelosi and Davide Marocco. Martin Peniak worked full time on the project for the first 4 months. Subsequently, Barry Bentley continued the experimental work. Frequent weekly meetings were held amongst staff of the Plymouth team. In addition, the Plymouth staff where in close contact (email and skype meetings) with the ACT team. Martin Peniak spent two weeks at the ACT site (January 2010) for joint work with the ACT staff on the simulator and PaGMO.

7   DISCUSSION AND PLAN FOR FUTURE WORK

Overall, the various experiments in this study demonstrates that the island model can produce solutions that are at least equally as good, or better in some conditions, than those evolved using standard sequential GAs. For example, in the comparison between maximum population sizes of 80, 160 and 240 individuals, the island with migration condition outperforms the other no-migration methods, especially for lower population sizes. Splitting the populations across multiple islands resulted in a significant reduction in the time required to achieve the same results through a standard GA. Evolved neural networks were found to be able to control the rover and deal appropriately with different types of obstacles.

The results also show that we do not observe a systematic advantage of the island model over all conditions, as for example this depends on the population size. This pattern of results might be explained by the fact that the rover navigation task used in most of the experiment is a relatively simple task of hole/rock obstacle. We would expect different stronger effects in experiments requiring the evolution of more complex behaviours. In particular, the island model might produce systematically superior results in the cases where the behaviour of the rover requires the composition/integration of modular skills (e.g. speed control, landmark tracking, obstacle avoidance). In this case, agents in different islands might specialise for different, individual skills and behaviours. The advantageous role of migration would come from the exchange and diffusion of various skills that can then be combined to improve the overall performance of the rover.

It is worth noting that with this approach the exploration and obstacle avoidance behaviours were not obtained through a pre-specified pattern of interaction between the rover and the environment. Rather, they are the emergent product of a fitness function working at the level of the whole behaviour of the robot. These behaviours are discovered autonomously through the evolutionary process and are functional to the optimisation of the global fitness used for the evolution. This research empirically verifies that this property of evolutionary robotics can be utilised with success in the design of a robust and computationally light controller, capable of dealing with the kinds of problems that future planetary robotic missions will face. As has been shown in this work, evolved neural network controllers can be extremely simple, require only minimal processing power and yet be very robust and reliable.

Results on the active vision experiments demonstrate the potential of such an approach for complex problems. The active vision experiments require the integration of local sensing information (e.g. type of slipper/stable terrain) and distal visual information about target and landmark. The system is capable to evolve minimal, optimal solutions that actively control the focus and location of the vision system to indentify important navigational information.

The results obtained in this study, and the methodological advances through the PaGMO/Rover simulator software tool, have the potential to lead to promising new research developments. One of the first directions of future work might regard the adaptation and testing of the autonomous navigation methodology to the physical rover robot platform. This ARIADNA study has focused on a simulation model of the rover robot, given the limited time scale, and the non direct access to the physical rover platform. For example, we hope that this might lead to a new collaboration with the Robotics division at ESA led by Dr Visentin, to extend the methodology for testing the solution evolvable through the simulation study on the on rover platform available at ESA Noordwijk. Another direction for future work is to continue the pilot experiments on the active vision configuration. In particular, one advantage of the active vision study is that the integration of local and distal sensing can be used to investigate autonomous goal oriented behaviour. The robot might

use distal vision information to identify and track a target or landmark, and use local sensing input to optimally navigate and deal with obstacle and the terrain. A further area of investigation with the PaGMO/Rover simulator methodology regards the design of dynamic and autonomous multi-agent role taking in collaborative tasks. Another interesting future research topic highlighted by the present study is the typology of the robot's control systems. In the pilot experiments on active vision we found the CTRNN producing very interesting result. A broader comparison between standard neural networks and CTRNN neural network types could lead to a great improvement in term of robustness and reliability of neural controllers. Moreover, the active vision task seems to benefit of the time integration capabilities provided by the CTRNN. Therefore, a better understanding of the role of stimuli time integration in the production of the robot's behaviour can help to produce better controllers, capable to autonomously cope with dynamical and unpredictable environments.

In addition to scientific advancements on our understanding of the island method in the evolutionary design of controller for the rover robot, this ARIADNA project has also had an important role in creating a new intense collaborative opportunity between the robotics team at the University of Plymouth and the ACT team at ESA. The collaboration with ESA staff in the past 8 months has been very fruitful and productive, and this will definitely continue after the conclusion of the study. In particular, this project has allowed the staff at the University of Plymouth's Centre for Robotics and Neural Systems to strengthen their involvement in space research. This is important as the UK government has recently initiate increased investment in space research, such as the establishment of an ESA UK site. Moreover, this is consistent with the University of Plymouth's strategic ambition to develop its space-related research, knowledge transfer and educational activities through Phase 2 of the Goonhilly Project, in partnership with Cornwall Council and Goonhilly Earth Station Ltd. The Plymouth robotic team is also involved with SciSys UK, a leading company in space research based in the southwest of England, on new ESA initiatives, such as their participation to the Start Tiger proposal in autonomous space robotics. We are confident that the fruitful collaboration initiated through the ARIADNA study will lead to a stronger partnership with the ESA ACT and robotics divisions.

8   REFERENCES

Ampatzis, C., Izzo, D., Rucinski, M., and Biscani, F., ALife in the Galapagos: migration effects on neuro-controller design, Proceedings of the European Conference on Artificial Life (ECAL 2009), 2009.

Beer, R. D. (1995). On the dynamics of small continuous-time recurrent neural networks. *Adaptive Behavior,* **3**:469-509

CNES (2010) MSL at a glance. [Online] Available: http://www.cnes.fr/web/5719-msl-09-at-a-glance.php

Izzo, D., Rucinski, M., and Ampatzis, C., Parallel global optimisation meta-heuristics using an asynchronous island-model, 2009 IEEE Congress on Evolutionary Computation (IEEE CEC 2009), Trondheim, Norway, May 18-21., 2009.

Miller, D. P., & Lee, T. L. (2002) High-speed traversal of rough terrain using a rocker-bogie mobility system. *In Proceedings of Robotics 2002: The 5th International Conference and Exposition on Robotics for Challenging Situations and Environments*.

O`Regan, J.K. & Noe, A. (2001). A sensorimotor account of vision and visual consciousness. *Behavior and Brain Sciences*, **24**(5), pp 939-1031

Peniak, M., Marocco, D., & Cangelosi, A. (2009a). Co-evolving controller and sensing abilities in a simulated Mars Rover explorer. *IEEE Congress on Evolutionary Computation (CEC) 2009.* Trondheim, Norway, 18th-21nd May

Peniak, M., Marocco, D., Ramirez-Contla, S., & Cangelosi, A. (2009b). An active vision system for navigating unknown environments: An evolutionary robotics approach for space research. *In Proceedings of IJCAI-09 Workshop on Artificial Intelligence in Space*, Pasadena, California, 17-18 July 2009.