Taylor & Francis
Taylor & Francis Group

# The use of evolution in a central action selection model

doi:10.1080/11762320701470541

F. Montes-Gonzalez, P. Bautista-Cabrera and V. Escobar-Ruiz

*Facultad de Física e Inteligencia Artificial, Universidad Veracruzana Sebastián Camacho 5, Col. Centro, Xalapa, Veracruz, México*

**Abstract:** The use of effective central selection provides flexibility in design by offering modularity and extensibility. In earlier papers we have focused on the development of a simple centralized selection mechanism. Our current goal is to integrate evolutionary methods in the design of non-sequential behaviours and the tuning of specific parameters of the selection model. The foraging behaviour of an animal robot (*animat*) has been modelled in order to integrate the sensory information from the robot to perform selection that is nearly optimized by the use of genetic algorithms. In this paper we present how selection through optimization finally arranges the pattern of presented behaviours for the foraging task. Hence, the execution of specific parts in a behavioural pattern may be ruled out by the tuning of these parameters. Furthermore, the intensive use of colour segmentation from a colour camera for locating a cylinder sets a burden on the calculations carried out by the genetic algorithm.

**Key words:** Action selection, behaviour-based robotics, evolutionary robotics.

## INTRODUCTION

The study of animals in their natural environment is essential for the development of ethological models that try to explain how a task can be solved by the activation and inhibition – 'the action selection' of different behavioural patterns (Montes González 2005). In a similar fashion the action selection problem in robotics can be resumed as making the right decision at the right time (Maes 1989). However, the identification of the right decision in animals is only a matter of selecting those actions that facilitate their survival. If we take into account that some of the internal mechanisms for selection in animals may have been 'optimized by the pressure of evolution, we will still notice the existence of different modular structures capable of producing a similar behavioural output. Nevertheless, these internal mechanisms employ specialized centres for selection to produce the right combination of these behavioural patterns to solve a particular task. Following these ideas we are building our action selection model.

In this paper, we are not concerned with the possibility of redundancy in the development of behavioural modules; for selection to occur we are focusing on the organization of these modules instead. In other words, granularity is not our main concern here; if at a lower level these modules are duplicating the use and the existence of basic motor commands, as long as selection is rightly done, the final behaviour should not be affected. Next, we explain the work behind the experiments in this paper. Thus, a brief explanation of the selection model namely CASSF and the use of genetic algorithms will be provided in the second subsection. Evolution is employed for the optimization of the parameters related to the selection model and is presented in the third subsection. Next, the integration of evolution in the design of the behavioural modules to be selected is reviewed in the fourth subsection . The presentation of the experiments we have carried out is given in the fifth subsection. Finally, a discussion of our work summarizing the findings related to these experiments is provided in the last section.

### Central selection and genetic algorithms

The use of genetic algorithms (Holland 1992) for the development of robot control systems (Nordin and Banzhaf 1997; Yamada 1998) commonly relies on the use of neural networks (Nolfi and Floreano 2000). On this approach, a neural controller represents one of the individuals of a current population. Usually, a direct encoding scheme is used to copy the weights of a neural controller into the array that forms an individual of one specific population. The genetic operators, selection, crossover and mutation,

*Corresponding author:*
F. Montes-González
Facultad de Física e Inteligencia Artificial
Universidad Veracruzana Sebastián Camacho 5, Col. Centro
Xalapa, Veracruz, México
Tel.: +522288172957; Fax.: +52288172855, Email: fmontes@uv.mx

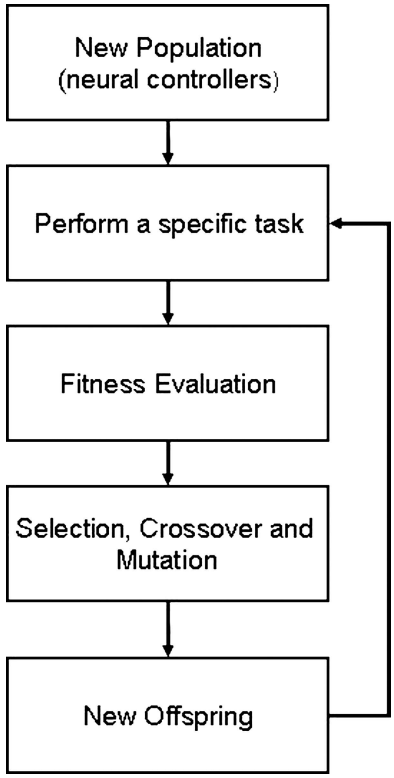


*Figure 1* The genetic algorithm iterates over generations of new individuals. A measure of their genetic value is given as 'fitness' in solving a particular task. The algorithm is stopped when an acceptable level of fitness is reached. Adapted from Santos and Duro (2005).

are applied to all the chromosomes of the entire population in order to spawn a new offspring. In turn, each chromosome forms the new weights of the neural controller that is evaluated for a limited period of time. A measure of performance in the solving of a specific task is calculated for all the individuals of a single population. Then, the performance represents the fitness to the environment and to the task that an individual is solving. The fittest individuals have more opportunity to share their genetic material into the next generation. The process of generating a new offspring, from the fittest parents of a previous population, is carried out until a satisfactory evaluation level is obtained (Figure 1).

The central model of action selection that we have developed builds a unified perception of the world using raw information from the sensors in the body of a table-top robot. The model of selection has been named CASSF because it employs central action selection with the use of sensory fused raw information (Montes Gonzalez and Marin Hernandez 2004; Montes-Gonzalez *et al.* 2006). In this model, at every step of the main control loop, sensor readings form perceptual variables which are used in the model to calculate the activation levels that represent the urgency of the behaviour to be executed. Selection occurs in a winner-takes-all fashion and the behaviour with the highest salience takes over the motors of the robot. In the case of behaviours having similar levels of activation, selection randomly occurs between them. The motors are controlled by motor commands sent from the behavioural modules. In turn these modules may contribute to the calculation of their own salience by sending a busy signal that represents a state where interruption should not occur (Figure 2).

In this paper we are modelling the foraging behaviour of an animat that has to collect cylinders set in the middle of a walled arena and put the cylinders near the outside walls of the arena. In our experiments we artificially build five basic behaviours which are: *cylinder-seek*, *cylinder-pickup*, *wall-seek*, *cylinder-deposit* and *look-around*.

The salience of these behaviours is computed as follows: the perceptual variables wall_detector ($e_w$), gripper_sensor ($e_g$), cylinder_detector ($e_c$), and turning_time ($e_t$) are formed in the context vector that is constructed as follows: ($\mathbf{e} = [e_w, e_g, e_c, e_t]$, $e_w, e_g, e_c, e_t \in \{1,0\}$). Secondly, the five different behaviours return a current busy-status ($c_i$) indicating that ongoing activities should not be interrupted.

Thirdly, the current busy-status vector is formed as described next, $c = [\ c_s,\ c_p,\ c_w,\ c_f,\ c_d\ ]$, $c_s,\ c_p,\ c_w,\ c_d,\ c_l \in \{1,0\}$, for cylinder-seek, cylinder-pickup, wall-seek, cylinder-deposit and look-around respectively. Finally, the salience ($\mathbf{s}_i$) or urgency is calculated by adding the weighted busy-status ($\mathbf{c}_i \cdot w_b$) to the weighted context vector ($\mathbf{e} \cdot [\mathbf{w}^e{}_j]^T$). Therefore, in equation (1) we have:

$$\mathbf{s}_i = \mathbf{c}_i \cdot w_b + \mathbf{e} \cdot \left(\mathbf{w}^e_j\right)^T . \tag{1}$$

The computation of the salience can be also considered as a decision neural network (highlighted in Figure 2), which is formed by an input layer of four neurons and an output layer of five neurons with the identity transfer function. The raw sensory information from the robot is fed into the neural network in the form of perceptual variables. Then, the input neurons distribute the variables to the output neurons. The behaviour with the highest salience level sends a busy signal to the output neurons. Next, the definition of the behavioural modules is explained. *Cylinder-seek* explores the arena searching for food while avoiding obstacles; *cylinder-pickup* clears the space for collecting cylinders; *wall-seek* locates walls while avoiding obstacles; *cylinder-deposit* lowers and opens an occupied gripper; finally *look-around* makes a full spin of the robot trying to locate the nearest perceptible cylinder.

The use of the genetic algorithm requires that initial random values be given to the weights $w_b$ and $\mathbf{w}^e{}_j$, which form the first generation of individuals that will be manipulated during the evolutionary process. Then, fitness is evaluated and the more adapted individuals are selected, their genetic material combined, and some of their individual contents may be mutated. However, the fitness function for evaluating the performance of these individual neural controllers has to be defined first. In the next section we provide the definition of this fitness function. Furthermore, we provide

 

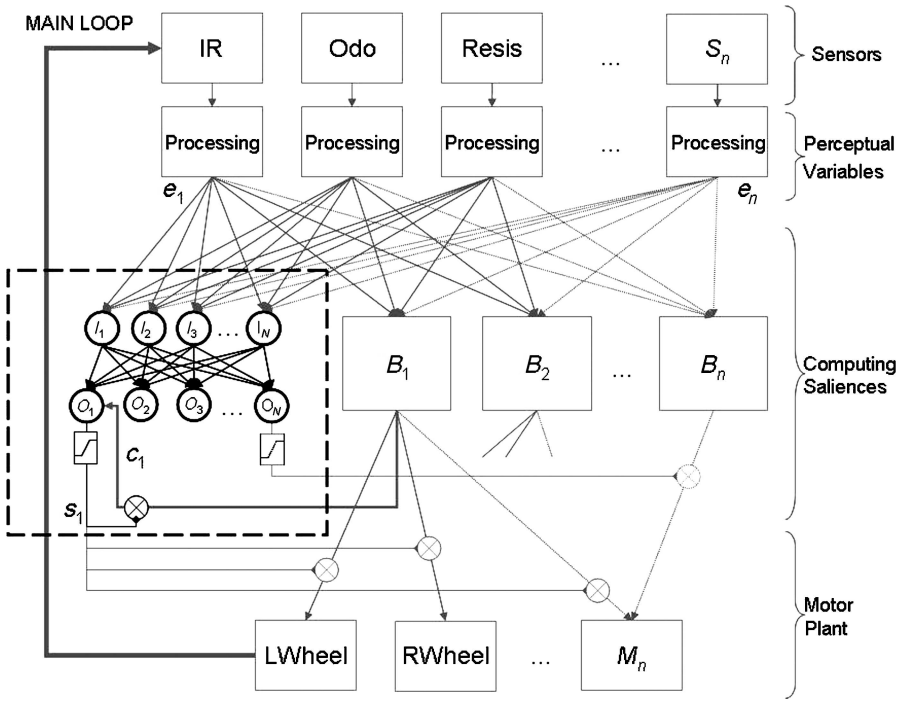


*Figure 2* In the CASSF model, perceptual variables ($e_i$) form the input to the decision neural network. The output of the selected behaviour with the highest salience ($s_i$) is gated to the motors of the Khepera. The busy-status signal ($c_1$) from behaviour $B_1$ to the output neuron $O_1$ should be noticed.

a concise explanation of the use of the genetic operators in the evolution of action selection and behavioural modules.

### The use of evolution in CASSF

The weights of the decision network in CASSF have to be properly set in order to obtain selection. The use of the genetic algorithm optimizes these weights by performing a gradient ascent search from initial random weights. The use of evolution in the setting of the weights requires some initial considerations for the genetic algorithm to work and these are next explained.

The evolution of the decision neural network makes use of a direct encoding for a chromosome **c** of 45 weights. Random values are set for the initial population $G_0$ of $n = 100$ neural controllers. The two more fitted individuals of one generation pass intact into the next generation as a form of *elitism*. The use of local competitions between random selected parents, with the fittest parents winning the competitions, is employed to select the procreators of the next offspring. The local competitions are part of a *tournament selection* that provides the genetic material of the most fitted pairs of parents, and their genetic material contents may be crossed over at one or several random points. Here, a new individual is spawned using a single random point with a *crossover* probability of 0.5. Individuals of a new offspring suffer *mutation* with a probability of 0.01. Finally, their fitness is evaluated for about 25 seconds.

The fitness formula for evaluating the overall behaviour of a robot provides the resultant action selection in CASSF. This formula takes into account the number of collected cylinders and their deposit near the outside wall of the arena. Therefore, the fitness formula supervises the completion of the foraging behaviour for the collection of cylinders in the arena. The fitness formula (equation (2)) for the foraging task that provides the final adjustment of the weights in the decision network was

$$f_{c1} = (K_1 \cdot laf) + (K_2 \cdot pickf) + (K_3 \cdot relsf). \qquad (2)$$

The evolution of the weights of the decision neural network were obtained using the constants $K_1$, $K_2$, and $K_3$ with $K_1 < K_2 < K_3$ for selection of the five basic behaviours in the fitness formula ($f_{c1}$). A formula like this should favour individuals that execute the look-around routine *(laf)*, and collect and deliver cylinders. Therefore, the fitness formula highly rewards the collection of cylinders inside the arena (*pickf*), and their release close the outside walls (*relsf*). The average fitness of a population, after only forty generations, and its maximum individual fitness are shown in Figure 3.

### Evolution and the design of robotic behaviour

In our model we have employed two kinds of behaviours; behaviours for exploring the arena (*wall-seek*, and *cylinder-seek*) that were evolved, and sequential behaviours (*cylinder-pickup*, *cylinder-deposit*, and *look-around*) designed as program routines. The behaviours wall-seek and cylinder-seek were evolved in a similar manner. Once the behaviours were properly evolved and designed, they were transferred to the robot for a final adjustment.

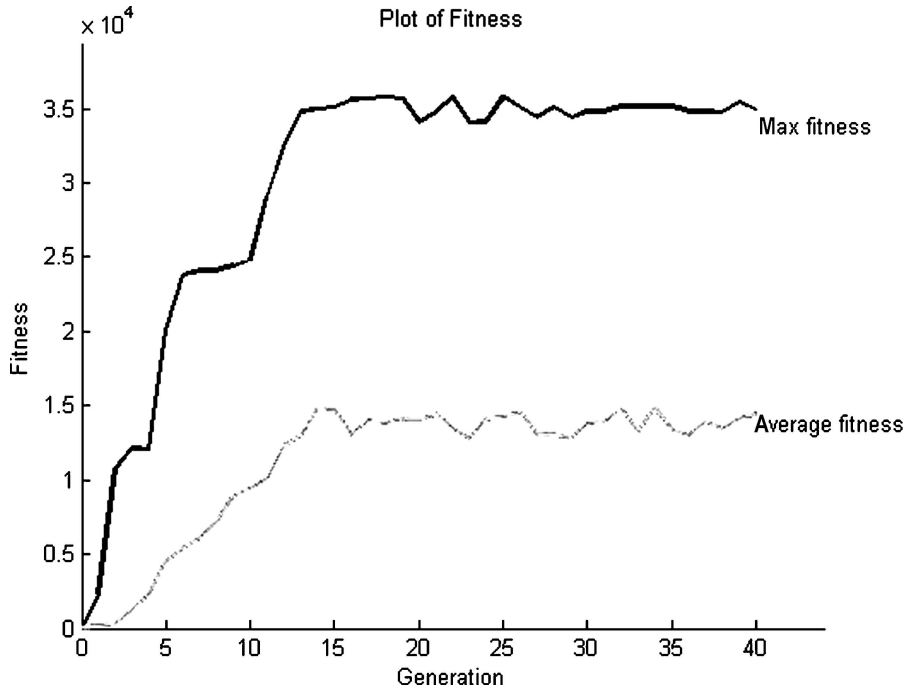


*Figure 3* Plot of the fitness for the decision network over 40 generations.

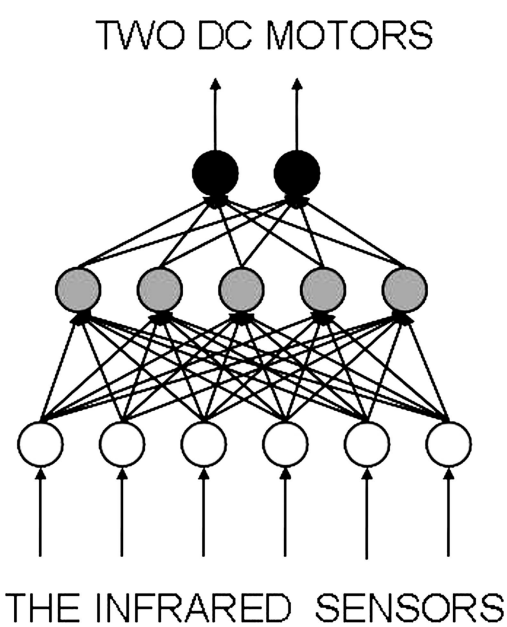


*Figure 4* The behaviours wall-seek, and cylinder-seek share the same neural network topology.

The development of the exploring behaviours required the use of the same neural network topology. Thus, a standard fully connected feedforward multilayer-perceptron with no recurrent connections was employed for the development of the exploring behaviours (Figure 4).

In this fully connected network, the input layer consists of six neurons that are connected to a hidden layer of five neurons. In turn, these hidden neurons send projections to the two neurons in the output layer. The infrared sensors values of the Khepera range from nothing detected (approximately 0) to something very close (approximately 1023). Then, the readings of the six frontal sensors are made binary with a collision threshold $t_{hc} = 800$. The use of binary inputs for the neural network facilitates the transference of the controller to the robot by adapting the threshold to the readings of the real sensors. The output of the neural network is scaled to $\pm$ 20 values for the DC motors.

For the behaviours wall-seek and can-seek, the genetic algorithm employs a direct encoding for the weights of the neural network which forms a vector $\mathbf{c}$ of 40 elements. Random initial values are generated for this vector $\mathbf{c}_i$, $-1 < \mathbf{c}_i < 1$, and $n = 100$ neural controllers form the initial population $G_0$. The two best individuals of a generation are copied as a form of *elitism*. Tournament selection, for each of the $(n/2)$ - 1 local competitions, produces two parents for breeding a new individual using a single random *crossover* point with a probability of 0.5. The new off-spring is affected with a *mutation* probability of 0.01. Individuals in the new offspring are evaluated for about 25 seconds in the robot simulator.

The development of the behaviour for locating a wall (*wall-seek*) can be seen as a form of obstacle avoidance because the robot has to look for a wall after fetching a cylinder. The robot has to run in a straight line until it bumps into a wall; however, obstacles have to be avoided while looking for the wall. The selection mechanism decides to stop this behaviour when a wall has been reached. The fitness plot for this behaviour is shown in Figure 5. The

 

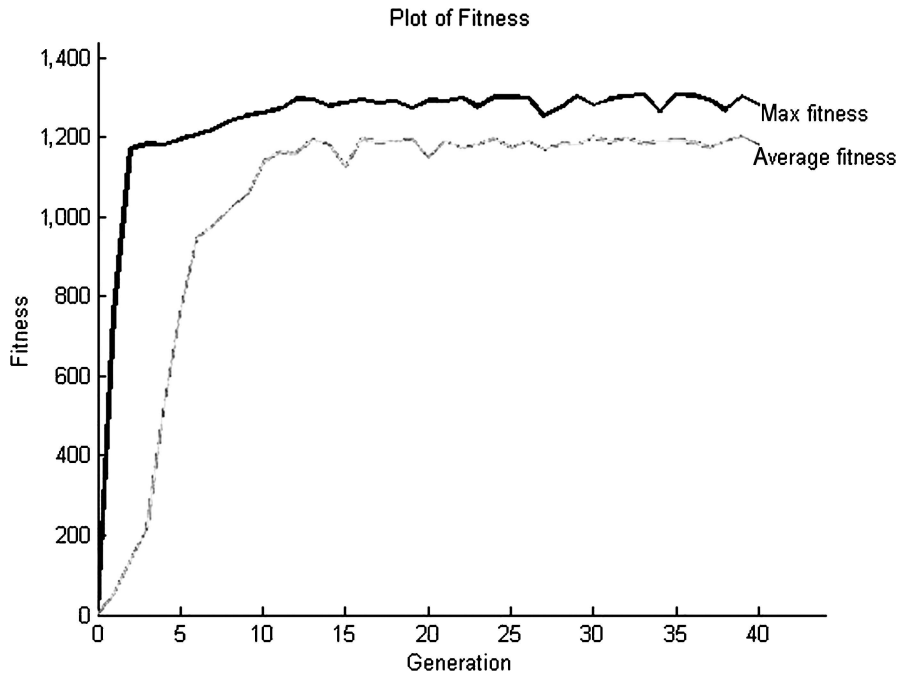


*Figure 5* The fitness for the wall-seek behaviour is plotted over 40 generations.

fitness formula (equation (3)) for the obstacle behaviour in wall-seek was

$$f_{c2} = \sum_{i=0}^{3000} abs(ls_i)(1 - \sqrt{\mathrm{d}s_i})(1 - \max\_ir_i) \ , \qquad (3)$$

where for iteration $i$: $ls$ is the linear speed in both wheels (the absolute value of the sum of the left and right speeds), d$s$ is the differential speed on both wheels (a measurement of the angular speed), and max_$ir$ is the highest infrared sensor value. The use of a fitness formula like this rewards those fastest individuals who travel on a straight line while avoiding obstacles.

On the other hand, the development of the *cylinder-seek* behaviour can be seen as a form of obstacle avoidance plus a cylinder-locator feature. The robot gripper of the Khepera is designed to reach an object exactly in front of the robot body. Thus, the robot has to be positioned right in front of the cylinder if the collection of the can is to occur. The cylinder-seek behaviour shares the same neural topology as the previously described behaviour, and its fitness formula (equation (4)) was as follows:

$$f_{c3} = f_{c2} + K_1 \cdot cnear + K_2 \cdot cfront. \qquad (4)$$

A formula such as this selects individuals that avoid obstacles and reach cylinders at different orientations (*cnear*), and are able to orientate the robot body in a position where the gripper can be lowered and the cylinder collected (*cfront*). The constants $K_1$ and $K_2$, $K_1 < K_2$, are used to reward the right positioning of the robot in front of a cylinder. The fitness plot for this behaviour is next presented (Figure 6).

For the sequential behaviours a different approach was followed due to its sequential nature. The *cylinder-pickup* behaviour was programmed as an algorithmic routine with a fixed number of iterations for clearing the space for lowering the arm, opening the claw, moving the arm downwards, and closing and lifting the gripper. Alternatively, the *cylinder-deposit* behaviour was programmed to execute a similar sequential movement, though in reverse order, which consists in lowering the arm, opening the claw, and lifting the gripper. Finally, the behaviour *look-around* commands the robot to make a full spin when the shaft encoders surpass a given threshold.

## Experiments and results

In these experiments the small (55 mm diameter) Khepera robot (Mondana *et al.* 1993) and the robot simulator software Webots (2007) have been employed (Figure 7). The Khepera has eight infrared sensors distributed around its body for sensing the environment; furthermore, two DC motors control the movement of the wheels. For the foraging task the robot was set in a four-walled square arena. A standard RS232 interface was used to connect the Khepera to the host computer. Additionally, the Khepera was also equipped with a standard gripper turret and a colour camera.

The camera is employed for the colour segmentation of blue cylinders, simulated 'food', distributed around the arena. The image of a detected cylinder is presented in Figure 8. In order to obtain colour segmentation, chromaticity invariance has to be achieved first; hence, illumination changes in an image can be attenuated by passing from *RGB* space to a normalized colour space (*rgb*). The

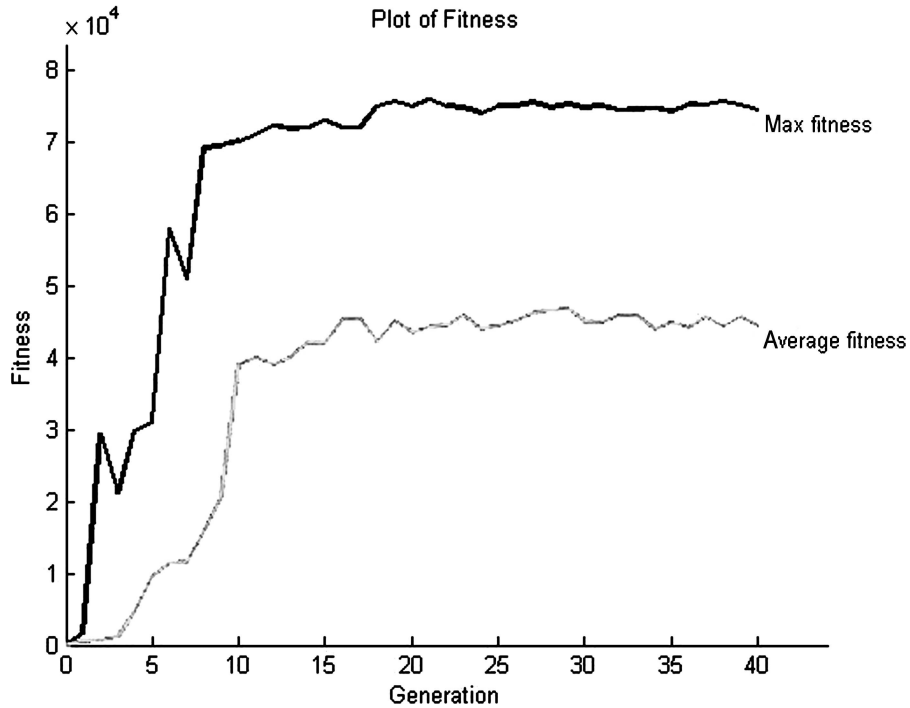


*Figure 6* Fitness is plotted for the cylinder-seek behaviour over 40 generations.

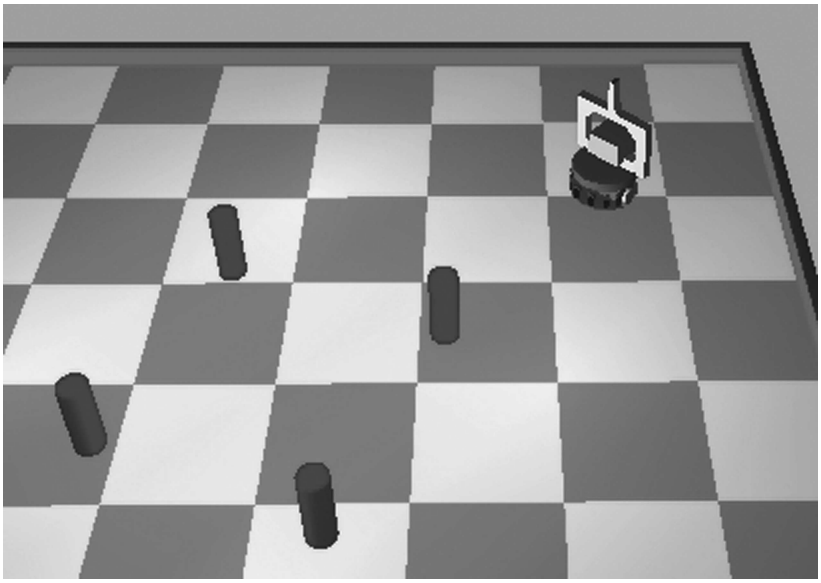

*Figure 7* The Khepera robot takes cylinders from the centre to the walls of the arena.

corresponding transformations are defined by:

$$r = \frac{R}{R+G+B}, \quad g = \frac{G}{R+G+B},$$
$$b = \frac{B}{R+G+B} \tag{5}$$

Following this approach we obtain a simple form of colour constancy, under various illuminating conditions, which preserves the directional information of the colour vector (Figure 8b). Therefore a blue cylinder can be characterized using its chromaticity, in the normalized colour space, with a statistical colour distribution. Then, a statistical band-pass filter produces the colour segmentation of this cylinder (shown in Figure 8c).

In our experimental setup four wooden cylinders, wrapped with colour paper, were initially placed in the middle of the trial arena (Figure 9). Then, the foraging behaviour consists of regular patterns of selection of the five basic behaviours. Therefore, one foraging bout can be resumed as the selection of the next behaviours: *cylinder-seek* travels the arena searching for cylinders while avoiding obstacles; *cylinder-pickup* then grasps a detected cylinder;

 

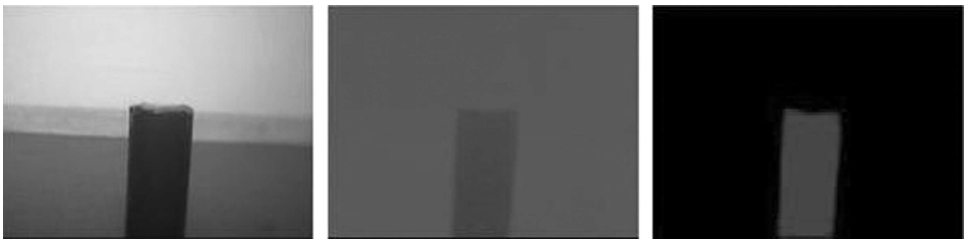

*Figure 8* The colour-segmentation of a cylinder is shown in a normalized colour space. A blue cylinder is shown in (a); then the same cylinder is presented in a normalized space (b). Finally, a colour-segmented image is shown in (c), where the cylinder appears as blue pixels.

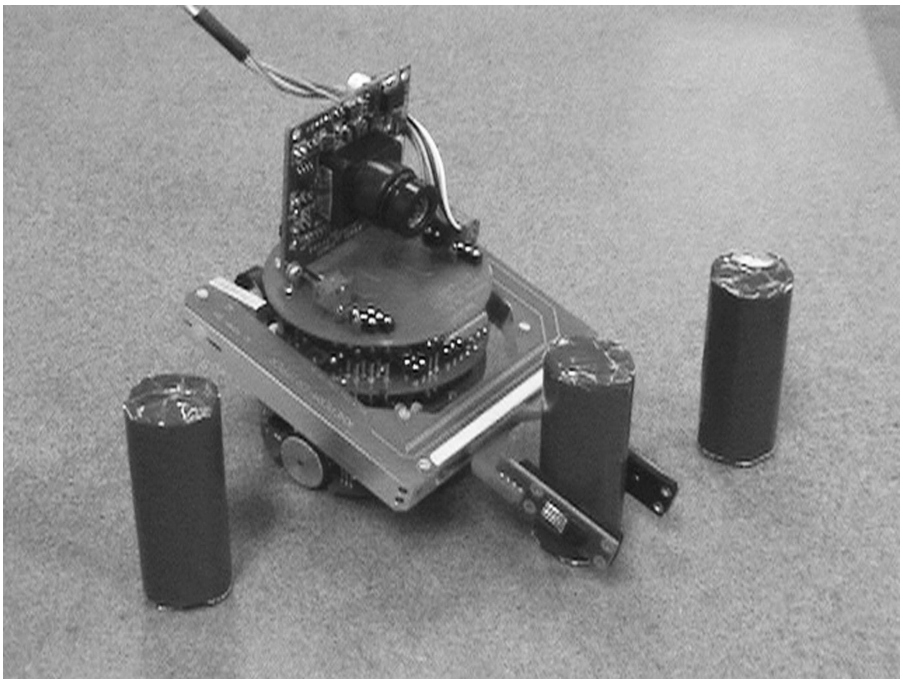

*Figure 9* The Khepera robot was set in the arena where the collection of blue cylinders occurs.

*wall-seek* is then used to find the nearest wall while avoiding obstacles; *cylinder-deposit* lowers and opens an occupied gripper for the release of the cylinder; finally, *look-around* makes the robot to spin until a cylinder is located.

It is important to notice that the genetic algorithm optimizes the selection of the behaviours, thus look-around is left out despite the fitness function rewarding individuals that make regular spins. One of the reasons for the banning of look-around in the foraging bout is due to colour segmentation being seldom ineffective for the localization of a cylinder when the robot is spinning. Furthermore, evolution runs on a main loop that is updated every 64 milliseconds; thus, if the localization of a cylinder occurs then the robot has to run straight to the cylinder in the next iteration. The latter explains why the localization of a cylinder is difficult to obtain when the behavioural modules have not been specifically set to exploit 'occasional' visual information.

The cylinder-seek module employs visual information to run straight to a detected can, and infrared information to position the robot body exactly in front of the can. In the absence of visual information this module randomly explores the arena until the can is detected relying only on the infrared information. Therefore, the genetic algorithm specializes cylinder-seek, avoiding the selection of look-around, to locate a can in a 'blindness' situation. Alternatively, the weights of a hand-coded decision-network have been set to execute cylinder-seek after the selection of look-around. Consequently, the hand-coded network performs slightly better because it is employing a combination of visual and infrared information.

The results of the experiments are presented as an ethogram summarizing four foraging bouts (shown in Figure 10). Additionally, some statistical data related to the foraging behaviour that lasted around 15 seconds is shown in Table 1. We observe on both the ethogram and the table fourteen instances of cylinder-seek, which is correct because the robot keeps looking for a cylinder. Also, in the table we observe fourteen occurrences in the selection of cylinder-pickup; however, in the ethogram fewer instances can be noticed. The latter, can be explained because the fetching of a can is disturbed by noise in the infrared

*Table 1* **Elementary statistics for a typical run of the foraging behaviour**

| Behavioural elements | | Freqency | Latency | | TotDurTotDur% | | | Mean | Std Dev | StdErr MinDur | MaxDu |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1. cylinder-seek | 14.00 | 0.02 | 10.71 | 71.97 | 0.76 | 1.33 | 0.36 | 0.00 | 4.75 | | |
| 2. cylinder-pickup | 14.00 | 1.17 | 1.81 | 12.19 | 0.13 | 0.20 | 0.05 | 0.00 | 0.55 | | |
| 3. wall-seek | 4.00 | 1.59 | 1.39 | 9.34 | 0.35 | 0.15 | 0.07 | 0.23 | 0.56 | | |
| 4. cylinder-deposit | 5.00 | 1.53 | 0.95 | 6.40 | 0.19 | 0.13 | 0.06 | 0.00 | 0.28 | | |
| 5. look-around | 0.00 | 14.88 | 0.00 | 0.00 | NaN | NaN | NaN | 0.00 | 0.00 | | |
| 6. none | 1.00 | 0.00 | 0.02 | 0.02 | 0.02 | 0.00 | 0.00 | 0.02 | 0.02 | | |
| Total | 38.00 | 0.00 | 14.88 | 14.88 | 0.39 | 0.85 | 0.14 | 0.00 | 4.75 | | |

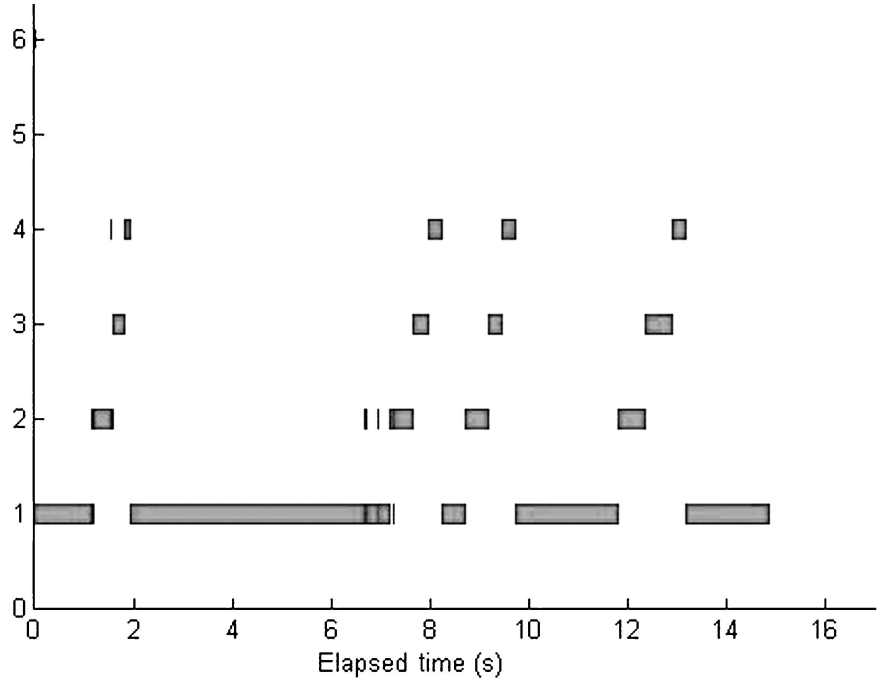


*Figure 10* The ethogram of the foraging behaviour presents the selection of behavioural patterns as bars with associated numbers. Therefore, behavioural patterns were coded as follows 1. cylinder-seek; 2. cylinder-pickup; 3. wall-seek; 4. cylinder-deposit; 5. look-around and 6. none behaviour. The ethogram in this image shows the best of the individuals after forty generations, which scored the highest fitness during evolution.

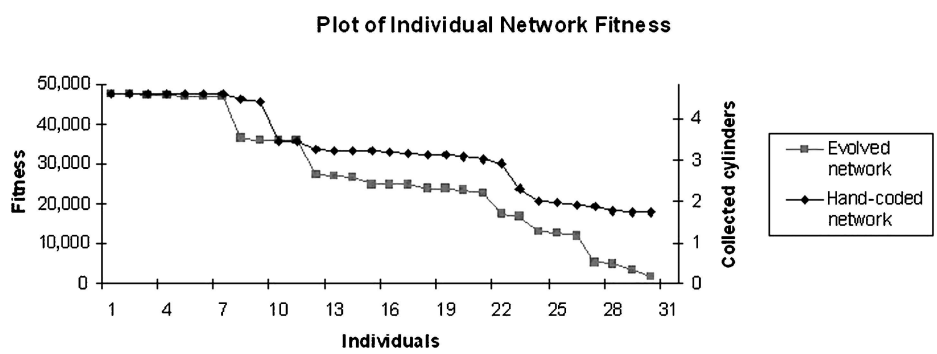


*Figure 11* The fitness of the hand-coded and the evolved decision networks are plotted for 30 individuals. The secondary *y*-axis shows the number of collected cylinders. Although the evolved decision network performs better for some individuals, this evolved network also produces the worst individuals. In contrast, the hand-coded network collects at least one cylinder in the worst case.

sensors; thus, cylinder-pickup is immediately initiated after being interrupted. In the ethogram the grasping of the four cans is identified by four bars having a similar size. The number of grasped objects is similar to the selection of wall-seek as presented in the same table. Nevertheless, in this table cylinder-seek is reported as selected five times, the additional execution of this behavioural module occurs after one of the occurrences of cylinder-pickup during the first bout. The activation of cylinder-deposit is the result of the blockage of the infrared sensors by the gripper arm when cylinder-pickup is selected; thus, an object similar to a wall is detected while holding a can, which are the sufficient conditions for cylinder-deposit to be executed.

The statistics in Table 1 present NaN (Not a Number) values as the result of the look-around never being selected. Furthermore, the latency for the selection of this behavioural pattern is reported as the total time because the execution of the behaviour was delayed up to the end of the experiment. On the other hand, in order to compare the fitness of evolved with hand-coded weights for CASSF we evaluated thirty individuals sharing the same behavioural patterns for both sets of weights. Therefore, in Figure 11 we observe the plot of the fitness of these individuals for the evolved and the hand-coded networks. Additionally, in the same plot the quantity of delivered cylinders that correspond to the number of foraging bouts is also shown. In the next section these results are discussed.

### General discussion

Colour segmentation was employed by the look-around behavioural pattern for the evolved and the hand-coded network. However, after the optimization of the evolved network the use of look-around was discarded for locating the cylinders in the arena. Therefore, an approach relying only on infrared information was followed, despite rewarding those individuals with a spinning behaviour. In contrast, the hand-coded network employed colour segmentation for locating the cylinders, thus combining visual and infrared information for this task. The use of the camera for locating the cylinders accounts for the better performance of the hand-coded decision network. In these experiments evolution optimized the foraging behaviour in time, thus ruling out the selection of look-around and specializing cylinder-seek to fetch cylinders with long 'blind' search-periods. As a result of looking for a cylinder without the aid of visual information, the worst performance of the evolved decision network occurs when none of them are collected. In contrast, the hand-coded decision network collects at least one cylinder in its worst performance.

Finally, in this paper we are proposing to develop a robot navigation scheme that relies on the use of the camera. Therefore, an intensive use of visual perception during evolution may improve the fitness of individuals over those that make occasional use of the camera. Furthermore, in order to optimize selection in the foraging behaviour we are planning to co-evolve the behavioural modules and the CASSF decision network.

### REFERENCES

Holland J H. 1992. Adaptation in Natural and Artificial Systems. Cambridge, MA: The MIT Press.

Maes P. 1989. How to do the right thing. *Connection Sci. J.*, 1: 291–323.

Mondana F, Franzi E, Ienne P. 1993. Mobile robot miniaturisation: A tool for investigating in control algorithms. In Proceedings of the 3rd International Symposium on Experimental Robotics, Kyoto Japan, Berlin: Springer-Verlag.

Montes Gonzalez F, Marin Hernandez A. 2004. Central action selection using sensor fusion. In The 5th Mexican International Conference on Computer Science (ENC'04), Colima, México, Los Alamitos, CA: IEEE Computer Society.

Montes González FM, Marín Hernández A, Ríos Figueroa H. 2006. An Effective Robotic Model of Action Selection. In Marín R et al., eds. CAEPIA 2005, LNAI 4177, pp. 123–132.

Montes-González F, Santos Reyes J, Ríos Figueroa H. 2006. Integration of Evolution with a Robot Action Selection Model. In Gelbukh A, Reyes-García, CA, eds. MICAI 2006, LNAI 4293, pp. 1160–1170.

Nolfi S, Floreano D. 2000. Evolutionary Robotics. Cambridge, MA: The MIT Press.

Nordin P, Banzhaf W. 1997. Real time control of a Khepera robot using genetic programming. *Cybernet. Control,* 26(3):533–561.

Santos J, Duro R. 2005. Artificial Evolution and Autonomous Robotics (in Spanish). Ra-Ma Editorial.

Cyberbotics. 2007. Webots, Commercial Mobile Robot Simulation Software.

Yamada S. 1998. Learning behaviors for environment modeling by genetic algorithm. In The First European Workshop on Evolutionary Robotics (EVOROBOT'98), Paris, France.

 