Sādhanā Vol. 28, Part 6, December 2003, pp. 999–1009. © Printed in India

Evolutionary robotics – A review

DILIP KUMAR PRATIHAR

Department of Mechanical Engineering, Indian Institute of Technology,
Kharagpur 721 302, India
e-mail: dkpra@mech.iitkgp.ernet.in

MS received 26 June 2001; revised 9 May 2003

Abstract. In evolutionary robotics, a suitable robot control system is developed automatically through evolution due to the interactions between the robot and its environment. It is a complicated task, as the robot and the environment constitute a highly dynamical system. Several methods have been tried by various investigators to solve this problem. This paper provides a survey on some of these important studies carried out in the recent past.

Keywords. Evolutionary robotics; genetic algorithm; neural network.

1. Introduction

Evolutionary robotics aims to develop a suitable control system of the robot through artificial evolution. Evolution and learning are two forms of biological adaptation that operate on different time scales. Evolution is capable of capturing slow environmental changes that might occur through several generations, whereas learning may produce adaptive changes in an individual during its lifetime. Recently, researchers have started using artificial evolution techniques, such as genetic algorithm (GA) (Goldberg 1989) and learning technique, namely neural network (NN) (Kosko 1994), to study the interaction between evolution and learning. Evolutionary robotics deals with this interaction.

In behaviour-based robotics, a task is divided into a number of basic behaviours by the designer and each basic behaviour is implemented in a separate layer of the robot control system (Brooks 1986). The control system is built up incrementally layer by layer and each layer is responsible for a single basic behaviour. The coordination mechanism of basic behaviours is usually designed through a trial and error process and the behaviours are coordinated by a central mechanism. It is important to note that the number of layers increases with the complexity of the problem and for a very complex task, it may go beyond the capability of the designer to define all the layers, their interrelationships and dependencies. Hence, there is a need for a technique by which the robot is able to acquire new behaviours automatically depending on the situations of changing environment. Evolutionary robotics may provide a feasible solution to the said problem.

In evolutionary robotics, the designer plays a passive role and the basic behaviours emerge automatically through evolution due to the interactions between the robot and its environment. Thus, decomposition of a task into a number of basic behaviours and their coordination are obtained through a self-organizing process (rather than by an explicit design). The designer defines the fitness function, which measures the ability of a robot to perform a desired task. The robot and the environment form a highly dynamical system, in which the robot's decision at a particular time depends on the sensory information and its previous actions. The principle of evolutionary robotics has been explained with the help of figures 1, 2, and 3. Developing a suitable adaptive NN-controller for a robot is the prime aim of evolutionary robotics. A binary-coded GA is used to provide training to the NN-controller. The GA (refer to figure 1) starts with a population of binary strings, created at random and each string indicates the weights of the neural network. The fitness of each string (see figure 2) is determined as follows.

- A number of training scenarios/cases are created at random and each scenario differs from the other in terms of the initial position, size and velocity of the moving obstacles.
- For each training scenario, a robot collects information about its dynamic environment using the sensors mounted on it. The sensory information is then fed as input to the robot controller (i.e. NN-controller) and it determines the output (refer to figure 3), which is realized through motor action. It is important to mention that the motor action does not modify the environment but the NN-controller tries to improve its output gradually so that it can produce better solutions in terms of travelling time. The robot's motion is executed and the travelling time to reach its destination (T) is determined, whereas t (figure 2) indicates the time required to travel a distance-step.
- After determining the travelling time for each scenario (i.e. T), the total travelling time (tot_T) is calculated considering all the training scenarios. The average travelling time

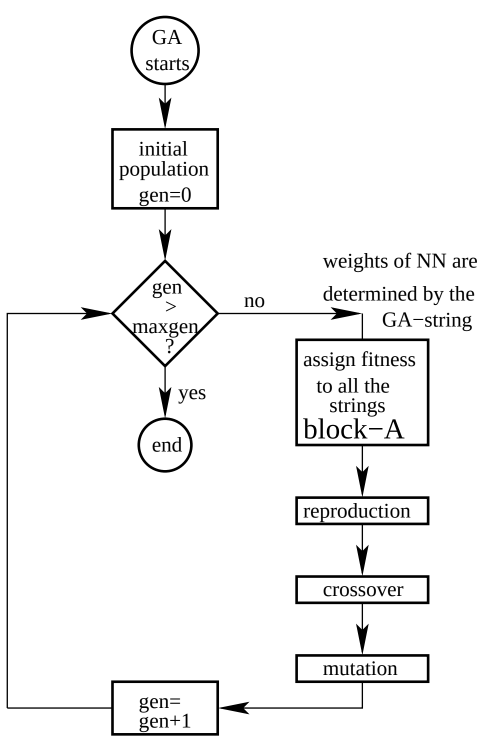

Figure 1. A schematic diagram showing the principle of a genetic algorithm.

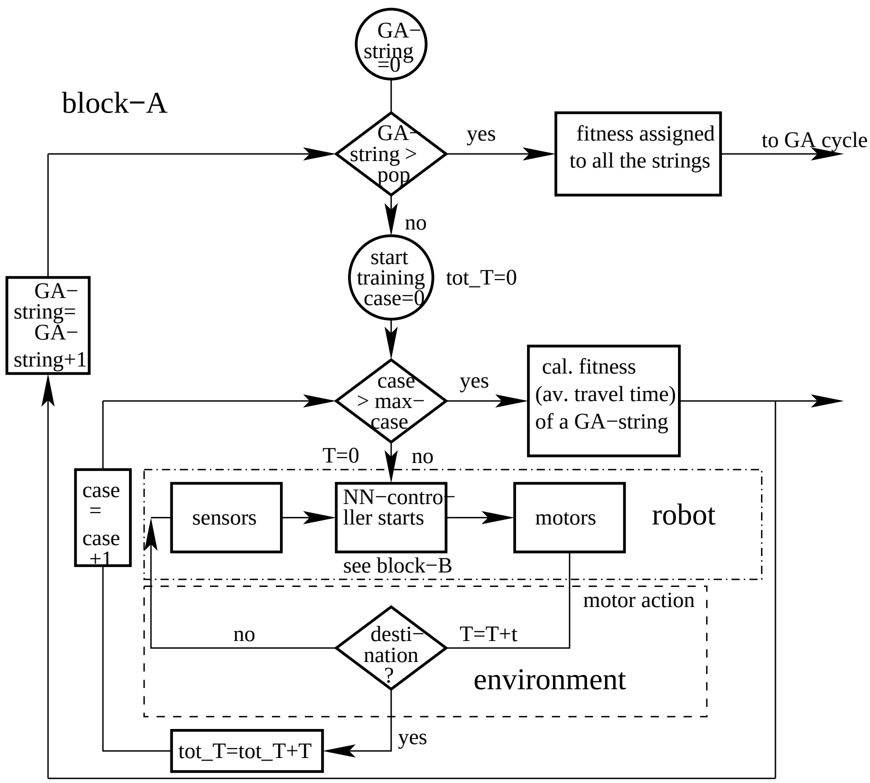

Figure 2. A schematic diagram to show the method of calculating the fitness of the GA-strings.

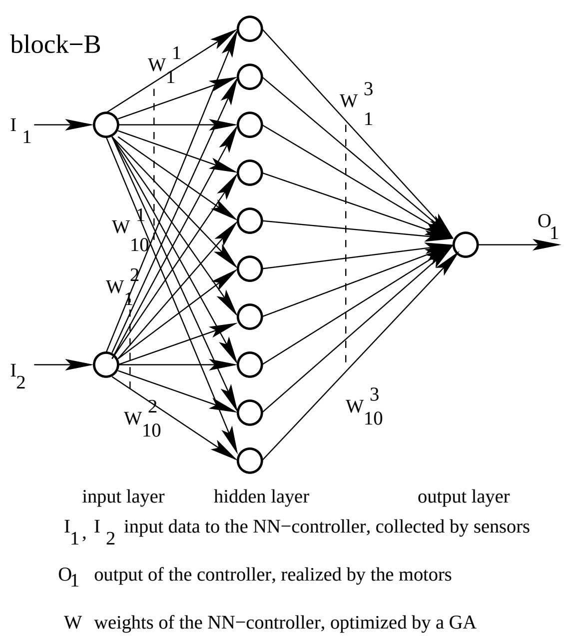

Figure 3. A schematic diagram showing the principle of a neural network.

(i.e. total travelling time divided by the number of training scenarios) is considered the fitness of a GA-string.

The fitness values of all the GA-strings contained in the population are determined similarly. The initial population of solutions is then modified using the GA-operators, namely reproduction, crossover and mutation. Thus, optimal/near-optimal parameters/weights of the NN-controller are determined by the evolution technique, i.e., GA through iterations.

It is important to mention that the robot controller may not behave in an optimal sense initially, but the controller improves its performance and produces better results gradually (after a few generations of GA runs). Thus, a suitable robot controller evolves through a self-organizing process. In evolutionary robotics, the environment plays a central role by determining which basic behaviour is active at any given time. Thus, the behaviour of a robot is an emergent property of its interaction with the environment. It is important to mention that a simple robot can show complex behaviour. Moreover, due to the dynamical interaction between the robot's control system and the environment, it is difficult to predict what kind of behaviour will be produced by a given control system. It is also difficult to predict which control system will produce a desired behaviour. Thus, design of the control system is a complex task. Adaptive behaviour is obtained through evolution (self-organization) and is difficult to achieve through explicit design. Evolutionary robotics may solve this problem in a more effective way as compared to behaviour-based robotics, as the evolution of behaviours and their coordination mechanism can be obtained through a self-organizing process rather than by an explicit design.

Evolutionary robotics uses techniques, such as genetic algorithm (Goldberg 1989), genetic programming (Koza 1992) and evolutionary strategy (Schwefel 1995) to evolve the controllers of robots. An initial population of different genotypes (i.e. information that evolves through successive generations) – each coding the robot's architecture, behaviour (i.e. phenotype), is created at random. Each robot is allowed to interact with the environment and a fitness is assigned to each of them. Robots with higher fitness are allowed to reproduce to hopefully create better solutions. The population of solutions is modified using some operators (crossover, mutation etc.) and ultimately good solutions are obtained through iterations.

Evolutionary robotics is a comparatively new field of robotics research, which seems to have enough potential to develop the control system of an intelligent and autonomous robot. It is a challenging area with many open research issues, such as robot–environment interactions, description of a behaviour, evolution and coordination of behaviours, effect of learning on evolution and vice-versa, and others. Some of these research issues have been discussed below.

(a) *Robot-environment interactions:* During navigation, a robot collects information about its environment with the help of its sensors, which is then fed as input to the robot controller and the output of the controller is realized through motor action. Thus, there are two kinds of robot–environment interactions – the first one is between the sensors mounted on the robot and the environment, and the second one is between the motors of the robot and the environment. Both the interactions are equally important to ensure the successful navigation of the robot. The sensory data are generally associated with imprecision and uncertainty and may also contain a lot of noise. Thus, carrying out data mining of the imprecise sensory data to extract the necessary and useful information, is a challenging task. Moreover, the sensors have to be mounted on the body of the robot, in an optimal sense, so that maximum information about the surroundings can be collected. The second kind of robot–environment interaction is the interaction between the motors mounted on the robot and its environment. Depending on the output of the controller, the motor runs to move the robot starting from its initial position to the final position. There must be proper coordination between the controller and the motor to ensure efficient movement of the robot and to maintain its stability (both kinematic as well as dynamic) during movement.

(b) *Description of behaviour, evolution and coordination of behaviours:* During navigation, the robot's task is divided into a number of basic behaviours. The problems related to evolution of basic behaviours and their proper coordination is solved by continuous interaction between the robot and its dynamic environment through a self-organizing process. Realizing the fact that adaptive behaviour is difficult to obtain through explicit design, researchers working in the field of evolutionary robotics have started considering whether a single agent can produce complex behaviour (besides the simple behaviour for which training is given to the agent) by exploiting the complexity of the environment.

(c) *Effect of learning on evolution & vice-versa:* Learning and evolution are two distinct processes, which occur on two different time-frames. Learning takes place during the life-time of an individual, whereas evolution is a slow process which generally occurs in a population through several generations. Learning may have a beneficial effect on the rate and adaptive power of evolution. The performance of a learning process may also be improved, if it is combined with an evolutionary process. Thus, the two processes – learning and evolution are related to each other and the effect of one on the other has to be studied in detail.

The approach of evolutionary robotics appears to be promising for the development of intelligent and autonomous robots but the main drawback of this field lies in the fact that evolution is a slow process. Evolutionary algorithms are slow and considerable amounts of time are needed to conduct the evolutionary process on a real robot. Thus, attention should be paid to make these algorithms reasonably faster. In this connection, the recent work of Pratihar *et al* (2001) is worth mentioning, in which they have proposed a faster GA (around five times faster than the normal GA), called visualized interactive GA (VIGA), though its performance depends on the task. Moreover, it is felt that learning may have a beneficial effect on evolution and has to be explored properly. The next section describes some of the recent work on evolutionary robotics done by several researchers.

2. Previous work

Floreano & Mondada (1994) developed neural controllers for autonomous robots using a genetic algorithm (GA) (Holland 1975; Goldberg 1989). The evolution of controllers was carried out on a physical robot – *Khepera*, a miniature test mobile robot (Mondada *et al* 1993). The goal was to develop the controller of a robot which can find a collision-free, preferably straight path, with high velocity during navigation. There were eight inputs and two outputs of the neural network (NN). The inputs were directly taken from the sensors and the outputs were connected to the motors. After GA-based learning of the neural controller, the navigation of the robot was successful. Later on, Floreano & Mondada (1996) used a GA to evolve a neural network controller for a *Khepera* whose task was to locate a battery charger. They considered twelve inputs (sensory information) and two outputs (one for each motor) of the NN controller.

Miglino *et al* (1994) proposed a technique in which a GA is used to evolve the weights for a neural network controller. The training of the controller was done on computer simulations and the best controller was downloaded onto a real robot. Although the real environment differed from the simulated environment, a reasonably good result was obtained. Moreover, feed-forward neuro-controllers were successfully evolved using a GA for the robot whose task was to clean an arena surrounded by walls (Nolfi & Parisi 1995; Nolfi 1996, 1997). A *Khepera* equipped with a gripper module had to clean an arena full of small cylindrical trash objects lying at random. The fitness was evaluated by counting the number of objects placed outside the arena by the robot during a given evaluation time. The controller was first evolved on simulations and it was then downloaded onto a real *Khepera* robot. Moreover, a GA was used by Miglino *et al* (1995), and Lund & Miglino (1996) to evolve a two-layer feed-forward NN with no hidden units. The controller was developed for a *Khepera* whose aim was to find a collision-free, straight path moving at maximum velocity. The networks were transferred to a *Khepera*, which was found to perform well. A similar two-staged approach was also developed by Salomon (1996) using a (3,6)–evolution strategy (ES) with self adaptation of the step size.

Smith (1997) was successful in evolving a football playing *Khepera*. A GA was utilized to set the weights in a fixed architecture NN. There were sixteen inputs (sensory information), sixteen hidden units and two outputs (motor control information) of the NN. Each of the sixteen hidden units was connected to both left and right motors. The behaviours were evolved first on simulation and then downloaded to the real *Khepera*. The robot was successful in locating the ball and scoring the goals both on simulation and real experiment.

Nolfi *et al* (1994) suggested a method in which evolution of a neural network controller was carried out on simulation and later continued on a real robot, *Khepera*. There were eight inputs and two outputs of the feed-forward NN and a simple GA (Holland 1975; Goldberg 1989) were used to evolve the weights for the NN. Due to mismatch between the simulated and the real sensory-motor apparatus, there was some difference in performance of the evolved networks when tested in the real environment.

Koza (1992) used genetic programming (GP) to develop subsumption architectures (Brooks 1986) for simulated robots engaged in wall-following and box-moving tasks. Later, Reynolds (1993) developed the control programs using GP, which helped a simple simulated moving vehicle to avoid collisions. GP was also utilized by Nordin & Banzhaf (1996) to evolve obstacle avoidance and object following behaviours in *Khepera*. They proposed an on-line GP approach to evolve behaviours, based on a probabilistic sampling of the environment. The sensory information was fed as input to the GP system and the outputs were the motor values controlling its behaviour. The goal of the learning robot was to find a control function (program) which reacts to the sensor data and produces the control action. Moreover, Lee *et al* (1997) developed behaviour primitives and behaviour arbitrators for a *Khepera* to push a box toward a goal position indicated by a light source. GP was used to evolve the controller programs of two behaviour primitives – box pushing and box side-following. An arbitrator program was also evolved to arrange the executing sequence of the behaviour primitives. The controllers evolved through simulations were downloaded to the real robot and the performance was found to be satisfactory.

Researchers at the University of Sussex, UK, felt that evolutionary robotics needs an adaptive improvement technique, rather than an optimizer and they switched over from the traditional GA (Holland 1975; Goldberg 1989) to a species adaptation genetic algorithm (SAGA). The concept of SAGA was introduced by Harvey (1992). In traditional GA, crossover plays the vital role in searching and mutation probability is kept to a low value. On the other hand, in SAGA, mutation is the main operator and generally a high value of mutation rate is considered. Moreover, the standard GA stops working whenever a converged population is obtained, whereas in SAGA, most of the evolutionary progress is made after the convergence of the algorithm. They realized that evolution is not an optimization process but an ongoing adaptation to the changing environments. Several attempts were made at the University of Sussex, UK, in which SAGA was used along with a dynamical recurrent neural network (DRNN) (in which the architecture is not fixed) to evolve behaviours for a gantry robot (Harvey *et al* 1994,1997; Husbands *et al* 1997; Jakobi 1997). There were seven input nodes (connected to sensory information), four output nodes (two for the virtual wheels and the other two for mirror angular velocities) and variable hidden nodes in the NN.

Colombetti & Dorigo (1993) used classifier systems (CSs) for robot control. ALECSYS software (Dorigo 1993) was utilized to evolve control architecture of the AutonoMouse, a mouse-shaped autonomous robot. The robot's control architecture was a set of some classifier systems – one for each desired behaviour and a coordinating CS. The behaviour to be evolved was light-chasing and the robot's fitness was determined by measuring light intensity using the light sensor. The performance was first tested on simulations and then it was transferred to a real robot and found to be efficient. Moreover, SAMUEL classifier system (Grefenstette & Cobb 1991) was used by Grefenstette & Schultz (1994) to evolve collision-free navigation of a *Nomad 200* mobile robot in the presence of some moving obstacles. The rule sets obtained through simulations were downloaded to the real robot and found to be efficient.

Meeden (1996) developed recurrent NN-based controllers for a 4-wheeled robot using a GA. The robot had to avoid contact with the walls during its movement and either to seek or avoid light depending on its goal. These conditions were considered in determining the fitness of the GA solution. A fixed architecture NN with seven inputs (depends on the sensory information), five hidden units and four outputs (directly connected to the motors) was considered in his work. The effectiveness of the controllers was also proved through the experiments.

Ram *et al* (1994) proposed one method in which navigation problem of a robot was divided into some basic behaviours, namely move-to-goal, avoid-static-obstacle, and others. These behaviours were implemented with the help of some parameters, such as goal gain (strength with which a robot moves towards its goal), obstacle gain (strength with which a robot moves away from obstacles) and obstacle sphere of influence (distance from obstacle at which a robot is repelled). They used a GA to find the suitable combination of these parameters so that the robot could find a collision-free path during its navigation.

Baluja (1996) developed an evolutionary algorithm named population-based incremental learning (PBIL) for designing a neural controller. The performance of the controller was tested on Carnegie Mellon's *NAVLAB autonomous land vehicle* for its steering control and found to be satisfactory.

Pratihar *et al* (1999a) developed optimal fuzzy logic controller by using a GA-based tuning off-line. The GA-tuned fuzzy logic controller was efficient in planning optimal collision-free path of the robot, while navigating in the presence of some moving obstacles. The algorithm was tested on simulations but is yet to be tried on experiments with real robots.

Jeong & Lee (1997) developed a two-stage controller for two-wheeled soccer-playing robots using a GA. In the first stage, some rules were evolved whose condition parts involve the positions of the ball, opponents, partners and goal, whereas the action parts indicate the actions to be taken, such as a move, a dribble or a kick. In the second stage, optimal on-off type control signals were produced which allowed a robot to reach a position with desired coordinates and orientation.

Beer & Gallagher (1992) used a standard GA to determine time constants, thresholds and connection weights of a continuous-time fully-connected recurrent neural network. The locomotion control of a six-legged insect-like robot was studied in their work. The effectiveness of their method was tested on both simulation and experiment with real robots.

Galt *et al* (1997) successfully derived the optimal gait parameters of an 8-legged walking and climbing robot using a GA. The gait parameters, namely phase (time relationships between the legs) and duty factor (support period of a leg) were encoded in the GA. The evolved controllers were found to perform well in generating suitable gaits of the 8-legged robot.

Gruau & Quatramaran (1997) evolved neural controllers using cellular encoding. They were successful in generating a quadruped-locomotion gait of an *OCT-1* robot. Gomi & Ide (1997) used a set of 50 control algorithms (software) to evolve a suitable gait of an 8-legged *OCT-1* robot. Each of these control algorithms was tested for generating gaits during a fixed amount of time. The robot is required to stand up and generate its gait to move forward. The program was run for a few generations and a mixture of tetrapod and wave gaits was obtained.

More recently, Pratihar *et al* (1999b, 2000) developed optimal/near-optimal fuzzy logic controllers (FLCs) for generating suitable gaits of a six-legged robot using a GA. The hexapod is supposed to cross a ditch or take a circular turn, keeping the minimum number of legs on the ground with maximum average kinematic margin. Each leg of the six-legged robot is controlled by a separate FLC. The GA-based tuning of the FLCs is done off-line. Thus, this algorithm is suitable for on-line implementations. The effectiveness of the algorithm is tested through computer simulations and found to be satisfactory.

3. Discussion and concluding remarks

Designing a suitable controller for an autonomous and intelligent robot, which can plan its motion on-line, in an unknown and changing environment, is a great challenge to the investigators working in this field of robotic research. Several methods have been tried by various researchers to solve this problem. Behaviour-based robotics is one of the notable outcomes of all such efforts. Although it can solve the motion planning problems of a robot, it has its inherent limitations, which can be eliminated using the principle of evolutionary robotics.

In evolutionary robotics, a suitable robot controller is evolved depending on the situations of a changing environment. In a changing and unpredictable environment, a robot controller cannot always be designed beforehand but a suitable controller may be evolved depending on the situation of the environment using some evolutionary techniques, namely genetic algorithms, genetic programming, and others. Thus, in evolutionary robotics, a suitable NN-controller or fuzzy logic controller is evolved using evolutionary techniques, which can provide an acceptable solution on-line, to real-world problems. The main drawback of evolutionary techniques is their slow convergence rate and the considerable amount of time that has to be spent to conduct the evolutionary process on a real robot. Thus, evolutionary algorithms have to be fast enough to get the real advantage of evolutionary robotics. Moreover, the issues related to interaction of learning with evolution have to be dealt with more carefully. Although evolution and learning are two distinct types of change occurring on two distinct entities (i.e., population and individual organism), they strongly influence each other. Learning may increase the adaptive power of evolution. On the other hand, the performance of a learning process is changed when it is combined with an evolutionary process. Thus, interaction between learning and evolution deeply alters both the learning and the evolutionary process. The effect of learning on evolution and vice versa should be studied in detail.

Emotion plays an important role along with intelligence in the decision-making process of a human being. Thus, the issues relating to modelling of artificial emotion will be researched in future along with the intelligence of a robot. Evolutionary robotics will also deal with modelling of artificial emotion, in future. Thus, tomorrow's evolutionary robot will be intelligent, autonomous and emotional as well. A lot of research is yet to be carried out to reach this milestone.

The author is profoundly grateful to the anonymous reviewers, whose valuable comments helped to improve the quality of the paper significantly. He thanks Prof H Takagi of the Kyushu Institute of Design, Fukuoka, Japan, for inviting him to his laboratory. During that visit, he collected some of the material in this paper. The manuscript was written and modified by the author during his visit to Darmstadt University of Technology, Germany, under the AvH Fellowship Programme. Financial help from the AvH Foundation, Bonn, Germany and the cooperation of Prof. Bibel and his group (at Darmstadt University of Technology) are gratefully acknowledged.

References

Baluja S 1996 Evolution of an artificial neural network based autonomous land vehicle controller. *IEEE Trans. Syst., Man Cybern.* B26: 450–463

Beer R D, Gallagher J C 1992 Evolving dynamic neural networks for adaptive behaviour. *Adaptive Behav.* 1: 91–122

Brooks R A 1986 A robust layered control system for a mobile robot. *IEEE J. Robot. Autom.* 2: 14–23

Colombetti M, Dorigo M 1993 Learning to control an autonomous robot by distributed genetic algorithms. In *Proc. of the Second International Conference on Simulation of Adaptive behaviour: From Animals to Animats 2*, (eds) J Meyer, Roitblat, S W Wilson (Cambridge, MA: MIT Press/Bradford)

Dorigo M 1993 Genetic and non-genetic operators in ALECSYS. *Evolut. Comput.* 1: 151–164

Floreano D, Mondada F 1994 Automatic creation of an autonomous agent: Genetic evolution of a neural network driven robot. In *From Animals to Animats 3: Proceedings of Third Conference on Simulation of Adaptive behaviour*, (eds) D Cliff, P Husbands, J Meyer, S W Wilson (Cambridge, MA: MIT Press/Bradford)

Floreano D, Mondada F 1996 Evolution of homing navigation in a real mobile robot. *IEEE Trans. Syst. Man, Cybern.*, 26: 396–407

Galt S, Luk B L, Collie A A 1997 Evolution of smooth and efficient walking motions for an 8-legged robot. *Proc. of the Sixth European Workshop on Learning Robots*, Brighton, UK

Goldberg D E 1989 *Genetic algorithms in search, optimization and machine learning*. (Reading, MA: Addison-Wesley)

Gomi T, Ide K 1997 Emergence of gaits of a legged robot by collaboration through evolution. *Proc. of the International Symposium on Artificial Life and Robotics* (Springer Verlag)

Grefenstette J, Cobb H C 1991 User's guide for SAMUEL (NRL Memorandum Report 6820). Washington, DC: Naval Research Laboratory, USA

Grefenstette J, Schultz A 1994 An evolutionary approach to learning in robots. *Proc. of the Machine Learning Workshop on Robot Learning*, New Brunswick, NJ

Gruau F, Quatramaran K 1997 Cellular encoding for interactive evolutionary robotics. In *Proc. of Fourth European Conference on Artificial Life*, (eds) P Husbands, I Harvey (Cambridge, MA: MIT Press/Bradford)

Harvey I 1992 Species adaptation genetic algorithms: The basis for a continuing SAGA. In *Proc. of the First European Conference on Artificial Life*, (eds) F Varela, P Bourgine (Cambridge, MA: MIT Press/Bradford) pp 346–354

Harvey I, Husbands P, Cliff D 1994 Seeing the light: Artificial evolution, real vision. In *Proc. of the Third International Conference on Simulation of Adaptive behaviour: From Animals to Animats 3*, (eds) D Cliff, P Husbands, J Meyer, S W Wilson (Cambridge, MA: MIT Press/Bradford)

Harvey I, Husbands P, Cliff D, Thompson A, Jakobi N 1997 Evolutionary robotics: The Sussex approach. *Robotics Autonomous Syst.* 20: 205–224

Holland J H 1975 *Adaptation in natural and artificial systems* (Ann Arbor, MI: University of Michigan Press)

Husbands P, Harvey I, Cliff D, Miller G 1997 Artificial evolution: A new path for AI? *Brain Cogn.* 34: 130–159

Jakobi N 1997a Half-baked, ad-hoc and noisy: Minimal simulations for evolutionary robotics. In *Proc. 4th Eur. Conf. on Artificial Life* (eds) P Husbands, I Harvey (Cambridge, MA: MIT Press/Bradford Books)

Jakobi N 1997b Evolutionary robotics and the radical envelope of noise hypothesis. *Adaptive Behav.* 6: 131–174

Jeong I K, Lee J J 1997 Evolving cooperative mobile robots using a modified genetic algorithm. *Robotics Autonomous Syst.* 21: 197–205

Kosko B 1994 *Neural networks and fuzzy systems* (New Delhi: Prentice-Hall)

Koza J 1992 *Genetic programming* (Cambridge MA: MIT Press)

Lee W P, Hallam J, Lund H H 1997a Applying genetic programming to evolve behaviour primitives and arbitrators for mobile robots. *Proc. of IEEE 4th Int. Conf. on Evolutionary Computation* (Piscataway, NJ)

Lee W P, Hallam J, Lund H H 1997b Learning complex robot behaviours by evolutionary approaches. *Proc. of the Sixth European Workshop on Learning Robots,* (Heidelberg: Springer Verlag)

Lund H H, Miglino O 1996 From simulated to real robots. *Proc. of the Third IEEE International Conference on Evolutionary Computation* (IEEE Computer Society Press)

Meeden L A 1996 An incremental approach to developing intelligent neural network controllers for robots. *IEEE Trans. Syst., Man Cyber.* B26: 474–485

Miglino O, Nafasi K, Taylor C 1994 Selection for wandering behaviour in a small robot. Technical Report: UCLA-CRSP-94-01, Department of Cognitive Science, University of California, Los Angeles, CA

Miglino O, Lund H H, Nolfi S 1995 Evolving mobile robots in simulated and real environments. *Artif. Life* 2: 417–434

Mondada F, Franzi E, Ienne P 1993 Mobile robot miniaturization: A tool for investigation in control algorithms. *Proc. 3rd Int. Symp. on Experimental Robotics*, Kyoto, Japan

Nolfi S 1996 Evolving non-trivial behaviours on real robots: a garbage collecting robot. Technical Report, Institute of Psychology, CNR, Rome

Nolfi S 1997a Using emergent modularity to develop control systems for mobile robots. *Adaptive Behav.* 5: 343–363

Nolfi S 1997b Evolving non-trivial behaviour on autonomous robots: Adaptation is more powerful than decomposition and integration. In *Evolutionary robotics: From intelligent robots to artificial life (ER'97)* (ed.) T Gomi (AAI Books)

Nolfi S 1997c Evolving non-trivial behaviours on real robots: A garbage collecting robot. *Robotics Autonomous Syst.* 22: 187–198

Nolfi S, Parisi D 1995 Evolving non-trivial behaviours on real robots: an autonomous robot that picks up objects. In *Proc. 4th Congress of the Italian Association for Artificial Intelligence* (eds) Gori, Soda (Springer)

Nolfi S, Miglino O, Parisi D 1994 Phenotypic plasticity in evolving neural networks: Evolving the control system for an autonomous agent. Technical Report: PCIA-94-04, Institute of Psychology, CNR, Rome

Nordin P, Banzhaf W 1996 An on-line method to evolve behaviour and to control a miniature robot in real time with genetic programming. *Adaptive Behav.* 5: 107–140

Pratihar D K, Deb K, Ghosh A 1999a A genetic-fuzzy approach for mobile robot navigation among moving obstacles. *Int. J. Approximate Reasoning* 20: 145–172

Pratihar D K, Deb K, Ghosh A 1999b Design of a genetic-fuzzy system for planning crab gaits of a six-legged robot. *J. Comput. Info. Technol.* (special issue on Evolutionary Computation) 7: 93–101

Pratihar D K, Deb K, Ghosh A 2000 Optimal turning gait of a six-legged robot using a GA-fuzzy approach. *Artificial intelligence for engineering design, analysis and manufacturing - AIEDAM* 14: 207–219

Pratihar D K, Hayashida N, Takagi H 2001 Comparison of mapping methods to visualize the EC landscape. *Proc. of the Fifth Int. Conf. on knowledge-based intelligent information engineering systems & allied technologies, KES'2001*, Nara, Japan, pp 223–227

Ram A, Arkin R, Boone G, Pearce M 1994 Using genetic algorithms to learn reactive control parameters for autonomous robot navigation. *Adaptive Behav.* 2: 277–305

Reynolds C 1993 An evolved, vision-based model of obstacle avoidance behaviour. In *Artificial Life III, Santa Fe Institute Studies in the Sciences of Complexity, Proceedings* (ed.) C Langton (Reading, MA: Addison Wesley) vol. 16

Schwefel H P 1995 *Evolution and optimum seeking* (New York: Wiley)

Salomon R 1996 Increasing adaptivity through evolution strategies. Maes, In *Proc. 4th Int. Conf. on Simulation of Adaptive behaviour: From Animals to Animats 4* (eds) P Maes, M J Mataric, J A Meyer, J Pollack, S W Wilson (Cambridge, MA: MIT Press/Bradford Books)

Smith T 1997 *Adding vision to Khepera: An autonomous robot footballer*. Master's thesis, School of Cognitive and Computing Sciences, University of Sussex