Procedia Computer Science
Volume 80, 2016, Pages 2261–2265
ICCS 2016. The International Conference on Computational Science

An Evolutionary Algorithm for Autonomous Robot Navigation

Lucas da Silva Assis[1], Anderson da Silva Soares[1], Clarimar José Coelho[2], and Jeffrey Van Baalen[3]

[1] Instituto de Informática, Universidade Federal de Goiás, Brasil
anderson@inf.ufg.br
[2] Departamento de Ciência da Computação, Pontifícia Universidade Católica de Goiás, Brasil
[3] Computer Science Departament, University of Wyoming, U. S. A

Abstract

This paper presents an implementation of an evolutionary algorithm to control a robot with autonomous navigation in avoiding obstacles. The paper describes how the evolutionary system controls the sensors and motors in order to complete this task. A simulator was developed to test the algorithm and its configurations. The tests were performed in a simulated environment containing a set of barriers that were observed by means of a set of sensors. The solution obtained in the simulator was embedded in a real robot, which was tested in an arena containing obstacles. The robot was able to navigate and avoid the obstacles in this environment.

Keywords: Evolutionary Algorithm, Autonomous Navigation, Artificial Intelligence, Simulation

1 Introduction

The navigation problem in mobile robotics is the problem of making navigation decisions for one or more autonomous mobile robots, placed in an arbitrary environment, to accomplish certain predefined tasks[5]. There are many aspects of this problem: environment settings, application peculiarities, robot characteristics, and task priorities. Many techniques such as fuzzy systems and evolutionary algorithms have been used in an attempt to solve this problem.

Evolutionary Algorithms (EAs) are a computing strategy that solves difficult optimization problems. EAs are inspired by biology, more specifically in the Darwinian Theory of Evolution. They abstract and mimic some of the traits of natural evolution to produce functional adaptive processes [1]. Genetic algorithms are intrinsically parallel because the search enables discovery of multiple solutions, and can control a population of robots that work simultaneously.

This paper describes a genetic algorithm, where the population exists as actual robots that exchange genetic information in order to adapt to solve a particular problem. The goal is to train a robot to interact with an unknown environment. Our main contribution is a novel chromosome encoding for mobile robotics and also a simulator for this task. As a case study,

Selection and peer-review under responsibility of the Scientific Programme Committee of ICCS 2016

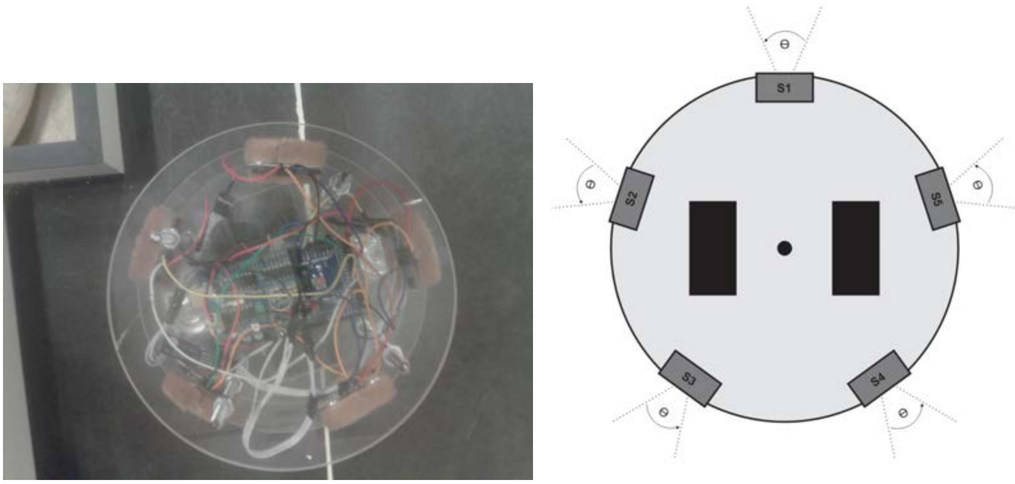
© The Authors. Published by Elsevier B.V.

doi:10.1016/j.procs.2016.05.404

we randomly initialize the robot to test whether it can actually be trained by evolution to do something practical such as exploration with obstacle avoidance.

2 The Proposed Simulated Hardware

The mechanics of the robot architecture were inspired by the Khepera III [4], a research platform for mobile robots widely used around the world. The developed platform is shown in Figure 1. It comprises the engine, motor controller, micro-controller, and the sensors in pentagonal configuration.



(a) (b)

Figure 1: Real (a) and simulated (b) robot configuration.

The simulation was carried out in a MATLAB environment that included a differential drive that simulates the physical behavior of the robot using the equations of motion of a wheeled robot differential. The differential steering model is a simple and reliable wheel-based drive system that is commonly used in smaller robots. It is essentially the same system used in a wheelchair: two wheels, mounted on a single axis, are independently controlled, providing both drive and steering functions. Known equations are used to predict how a robot equipped with such a system will respond to changes in its wheel speed and what path it will follow under various conditions.

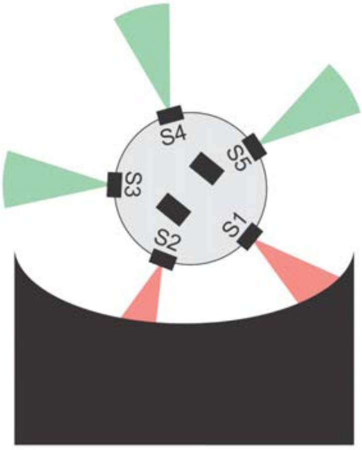
3 The Proposed Algorithm

In robotics, the use of evolutionary algorithms has been proposed for both optimizing morphology and for developing navigation control strategies [2]. These algorithms have been used mainly because autonomous robots and their controllers are unstructured, i.e., environments are flexible and/or partially unknown. This makes the design task very difficult for human designers. It is extremely difficult to know a priory every situation that a robot will encounter.

In this work, we propose an encoding for evolutionary algorithms in which the robot learns and builds its navigation rules unsupervised. We propose a chromosome encoding directly linked to behavior. An individual robot is represented by the set of actions that express all possible combinations of information provided by the sensors.

To simplify the representation we limit the measurements to a binary encoding, that is, values less than a threshold indicate obstacle detection and larger than threshold indicate a clear path. In the proposed architecture we have five sensors available, making 32 actions for each possible set of observations taken by the sensors.

As can be seen in the Figure 2, a combination of sensor values produces an integer output. And as illustrated in Figure 3, each possible integer output is associated with an action to take. Hence, a chromosome is a vector that encodes 32 states of each engine. These actions are indicated by the intensity of Pulse-Width Modulation (PWM), and are integers in the range -255 – 255.



(a)

Sensors	Sensors
1st Sensor - S1	Blocked
2nd Sensor - S2	Blocked
3rd Sensor - S3	Free
4th Sensor - S4	Free
5th Sensor - S5	Free
Binary Response	1 1 0 0 0

(b)

Figure 2: Proposed encoding.

S1	S2	S3	S4	S5
0	0	0	0	0
0	0	0	0	1
0	0	0	1	0

ind	M1	M2
0	51	0
1	32	-90
2	46	-30

Figure 3: PWM chromosome encoding.

Selection operators are applied in the population to determine which individuals will reproduce, generating new individuals for the next generation. Individuals with greater relative fitness have higher probability of reproducing, transmitting their features to new individuals. In this work, the fewer points an individual gets, the better qualified it is.

To achieve the objective, an individual must be able to efficiently explore the largest possible portion of the arena, avoiding collisions with static obstacles.

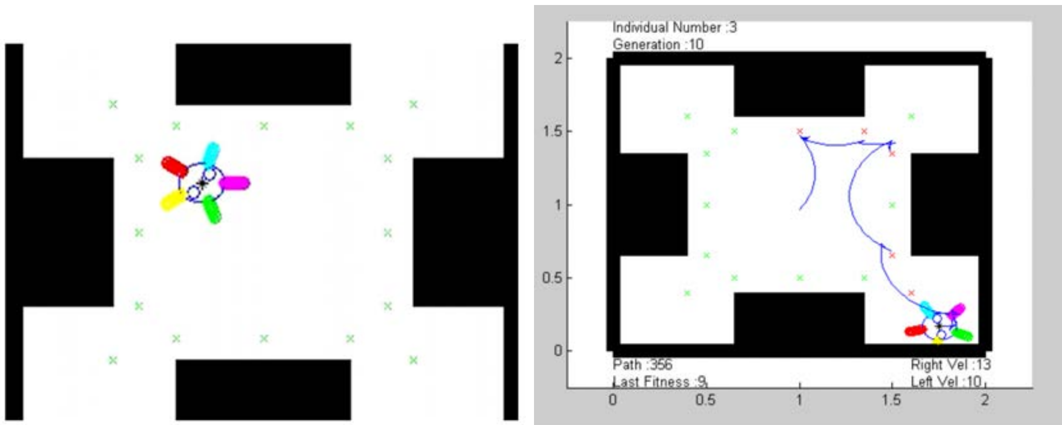
To reward individuals who explored more map area, points of interest (check points) were scattered and collecting these goals assigns a single score, that is, after collecting the point of interest is disabled. The contribution of this reward can be seen in Equation (1).

$$\text{fitness} = \frac{\text{Distance} + (\text{number of collisions}) * (\text{penalty for collision})}{(\text{number of check points}) * (\text{bonus for check points})} \tag{1}$$

In general in evolutionary algorithms, the best 20% of solutions are selected to participate in reproduction as one of the parents and the other parent is randomly chosen from the remaining individuals [2]. The reproduction process is performed using uniform crossover and flip mutation.

4 Results and Discussion

In Figure 4(a) the robot was initialized with a random initial solution. After 10 generations (Figure 4(b)) the solution obtained enables the robot to travel across the arena visiting some check points. The evolution of the robot shows that both the sensor configuration and the speed of the motors are under evolutionary control. This simulated experiment facilitates the correct combination of mutation rate, selection, and reproduction strategies. The results show that the algorithm is able to manipulate the sensor configuration in the chromosome, plus the bits that control the robot speed levels. The chromosomes are able to navigate to avoid collisions with the walls and to visit the check points.



(a) (b)

Figure 4: Simulated environment in initial configuration (a) and after executions (b).

After the simulation, we took the best solution and embedded it in the real robot in a real arena with obstacles similar to the simulation. We measured the number of collisions and the covered distance and compared with the solution obtained by Simões [3]. Each solution, ours and Simões, was executed ten times during fifteen (15) minutes each in the robot. As can be seem the solution provide by our algorithm had a better performance than solution proposed by Simões [3]. The solution proposed has about 56% less collisions on average than the other solution covering a bigger distance. In addition, in one execution the robot does not have any collisions.

	Number of collisions		Covered distance (m)	
	Average	Minimum	Average	Maximum
Our Solution	2.1	0	34.4	41.2
Simoes [3]	3.7	2	32.5	41

Table 1: Comparison between the best solution with the proposed algorithm and the best solution obtained by algorithm proposed by [3].

5 Conclusions and Future Works

This paper proposes an evolutionary algorithm for autonomous robot navigation. The results obtained in simulation enabled the development of new strategies. The simulator also made possible rapid evaluation of different parameters such as different mutation rates, reproduction, and selection strategies. The simulator produced a satisfactory collision-free behaviour on average after 15 generations. The solution developed by our system was embedded in a real robot and experiments demonstrated that this embedded system was able to successfully complete the autonomous navigation task in a real arena.

5.1 Acknowledgments

The authors would like to thank CAPES, CNPq and Foundation for Research Support of the State of Goiás (FAPEG) for the financial support to this project.

References

[1] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley Publishing Company, Inc., Reading, MA, 1989.

[2] Hong Qu, Ke Xing, and Takacs Alexander. An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots. *Neurocomputing*, 120(0):509 – 517, 2013. Image Feature Detection and Description.

[3] E. V. Simões and D. A. C. Barone. Predation: an approach to improving the evolution of real robots with a distributed evolutionary controller. *IEEE ICRA - International Conference on Robot Automation*, pages 764–772, 2002.

[4] Spyros G. Tzafestas. *Introduction to Mobile Robot Control.* Elsevier, 2014.

[5] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.