ROMERO'S ODYSSEY TO SANTA FE: FROM SIMULATION TO REAL LIFE

CHRISTOF TEUSCHER, EDUARDO SANCHEZ and MOSHE SIPPER

Logic Systems Laboratory, Swiss Federal Institute of Technology,
INN–Ecublens, CH-1015 Lausanne, Switzerland.
E-mail: {name.surname}@epfl.ch, URL: http://lslwww.epfl.ch.

ABSTRACT

Many a researcher in evolutionary robotics would love to have at his or her disposal a real population of bopping critters. Toward this end, the ultimate goal of our work is twofold: (1) to build a robot that is cheap, modular, and adaptable; and (2) to use a population of such robots in evolutionary-robotics experiments. As a first step in the study of adaptive robotics using populations of cheap robots, we have designed the Lausanne trail, a real-world version of the simulated Santa Fe trail. We use Lego-based robots equipped with low-cost sensors and actuators to evolve solutions in the real world to the trail-following problem. We describe two different sets of experiments and conclude that the real-world problem exhibits fundamental differences from the virtual one; in particular, the robot's lack of positional information hinders greatly its performance. We propose to study in the future collective map building, an important and interesting problem, which will, among others, provide a solution to the Lausanne-trail problem.

KEYWORDS: robotics, evolutionary algorithms, Santa Fe trail, real-world setup, LEGO

INTRODUCTION

Researchers working in the field of robotics and especially in the field of evolutionary robotics are confronted with several problems: (1) How many sensors does a robot need in order to perform a given task in optimal time? (2) What is the best fitness-function definition? (3) What is the best problem representation? (4) Will my robot be "intelligent" enough to perform the task? (5) In what ways do the robot's goals and its environment influence the evolutionary approach? These questions are hard and probably do not have general answers. Every roboticist knows that solving a problem, even using an evolutionary approach, requires a considerable amount of human knowledge. Unconstrained and open-ended evolution is still too computation intensive and is not yet suitable for solving real-world problems.

In this paper we try to explore the physical limits of a robot and of the evolutionary algorithms applied to a simulated and to a real-world setup of the Santa Fe trail. We describe the problems encountered when our robot escapes from the simulated Santa Fe trail into the real-world Lausanne trail. We performed various experiments both with the real-world setup and with a simulator in order to investigate the lower bounds of complexity necessary to eat the "food pellets" lying along the trail.

This paper is organized as follows: the next section describes the Santa Fe trail, followed by a motivation for our approach and a discussion of the importance of working with real robots. Then, our simple robot is presented that we used in all our experiments. A description of the real-world setup is followed by a presentation of the experiments and the results obtained. We conclude the paper with thoughts on future work in the last section.

THE SANTA FE TRAIL

In search for a simple task in an environment that would be easy to create, we have taken interest in ant navigation and foraging tasks. This class of problems—inspired by nature [8]—is still a topic for many researchers in biology.

Jefferson *et al.* described in [9] what they called the *Tracker* task. This evolutionary experiment was inspired by the behavior of certain species of ants which lay down pheromone trails in a process of collective foraging. In the initial Tracker setup, an ant was required to follow a crooked, broken trail of black cells in a white toroidal grid (John Muir trail [9, p. 557]). The trail gets more and more difficult towards its end. The ant's task was to move from cell to cell, traversing as much of the trail as possible in a given time. A somewhat more difficult trail was used by Koza [11]: the Santa Fe trail, designed by Christopher Langton [11, p. 55].

Ant navigation and foraging tasks are advantageous for evolutionary robotics in real-world setups because they normally take place in simple environments (e.g., a black trail on a white surface) and require simple robots with few sensors. They are thus easy to set up. Their solution, however, is far from easy.

In the Santa Fe trail (e.g., [11]) an artificial ant is placed on a 32×32 grid of cells, its objective being to collect as many food pellets strewn about as possible in a given number of times steps. The ant starts out facing south, and positioned in the upper left-hand cell of the grid, identified by coordinates (0,0). At each time step it can turn in any direction and walk one cell in that direction. The main difficulty faced by the ant lies in the positions of the pellets, which are not arranged in a simple, consecutive manner: the trail comprises an obstacle course of ascending difficulty, containing different kinds of gaps. All in all, the meandering trail contains 89 food pellets.

To the best of our knowledge, no one has studied the Santa Fe trail problem in a real-world setup.

MINIMAL ROBOTICS AND WHY WE NEED REAL ROBOTS

The interest in *minimal robotics* has largely influenced our work. Simple approaches to solve complex problems—if they exist—are often fascinating, need few sensors, simple hardware, and simple basic behaviors. The phenomenon commonly called *emergent behavior*, where the sum is greater than its parts [1, 14], should not be considered as mystical. Emergent behavior is the consequence of the complexity of the world, where a system—in our case a robot—moves, perceives, and interacts. In the field of robotics, Braitenberg [4] demonstrated striking examples of complex robot behavior emerging from simple rules. Emergent behavior becomes even more interesting if the robot not only interacts with the environment but also with other individuals.

For the present work, we chose a simple example which was relatively easy to analyze and easy to decompose into sub-problems, i.e., basic behaviors. We tried to use very simple approaches to solve the problem in order to find the lower bounds of robot and algorithm complexity needed to eat all the food pellets lying along the trail. This task turned out to be a real challenge, particularly for the real-world setup.

Embodied evolution (EE) [7,18] avoids the pitfalls of the simulate-and-transfer approach. Ficici *et al.* describe the method as follows: "We define EE as evolution taking place within a population of real robots where evaluation, selection, and reproduction are carried out by and between the robots in a distributed, asynchronous, and autonomous manner" [7]. This technique allows to speed-up evaluation time through parallelism. Note that in EE, there is no central controller or supervisor. EE also highlights the need for populations of real robots. The problems occurring when transferring an algorithm from simulation to reality are mostly due to lack of fidelity in the simulator. It is very difficult to take into account all the physical characteristics of the robots and the environment. "The world is its own best model" [5] summarizes the problem of the inability to model faithfully the world.

When working in the real world, total predictability is no longer possible—nature is noisy. Even when working with very simple environments, "...it is very hard to simulate the actual dynamics of the real world" [6, p. 4]. Generally, the number of trials (generations) needed to evolve a system discourages the use of physical robots, at least during the training period. However, a real and large population of robots can markedly speed up the process of evolution.

We acknowledge that simulation is a valuable tool to validate and understand different algorithms and setups. For our experiments, we built a very simple simulator used for comparison with the real-world experiments. The simulator was not intended to provide high-fidelity simulation of the Lausanne trail. We use the simulator to identify problems and critical factors as early as possible in the evolutionary process of a robot controller, but we did not transfer solutions from simulation to reality.

ROMERO: A MINIMAL ROBOT

We decided to build a robot population that is cheap, modular, and adaptable. $LEGO^{TM}$ MindStorms [10, 12] allows one to build any of several types of robots in a matter of minutes, adapting it to a specific experiment, or to a given environment. Our very simple robot dubbed *Romero* (Spanish for pilgrim) [16, 17]: it is cheap, uses modular LEGO components, and can be easily adapted and re-adapted to new tasks and novel environments. Romero is equipped lightly: it has two wheels, each controlled by a separate motor, and can be equipped with up to seven light sensors or touch-buttons. For our experiments, the sensors were attached to the bottom panel and directly faced the ground. Romero is built of Lego components, except for the controller, which we designed and build ourselves. Each robot is equipped with a sonar receiver/transmitter attached onto the mainboard. Compared to the well-known Khepera robot [13], Romero is more lightly equipped with sensors and electronic yet cheaper, thus opening up the possibility of building a true robot population.

THE LAUSANNE TRAIL: A REAL-WORLD SETUP

The Lausanne trail (Figure 1) is a real-world version of the original, simulated trail, intended for use with actual robots. We added a dark-colored track enclosing the environment, which the robot uses to get back to the starting point; furthermore, the 32×32 grid is not toroidal, as with previous simulated setups. The proportion between gaps and trail has been respected. For our experiments, we printed the trails on A0 paper, i.e., of size 90cm x 120cm. The Lausanne trail is well adapted to the size of the robots, contains no active and complicated elements, is cheap, and needs no maintenance.

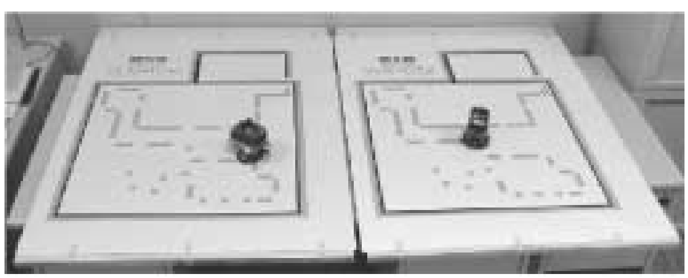

Figure 1: The Lausanne trail experimental setup. We added a special track enclosing the environment, which is used to guide the robot back to the starting point.

There are some important differences between the real-world setup and the original simulated environment: Romero has—apart from the ground-facing light sensors—no other navigation or localization system. Moreover, as opposed to the simulated worlds, the grid markings (i.e., the cells) have been removed; thus, Romero cannot move from cell to cell—as these latter do not exist. We felt that this was closer to the predicament of real ants, which do *not* live on a checkerboard.

In Koza's simulations [11], the food pellets were removed when the robot passed over them, whereas in our case this is not possible (the pellets form part of the printed environment). One problem raised by the pellets' fixity is that the robot can collect much food by simply revolving about its axis like a spinning top. As we will see later on, this is one of the major problems arising in the Lausanne trail.

EXPERIMENTS AND RESULTS

We currently have but two robots. Nonetheless, we proceeded under the "wishful" illusion that we were dealing with a population of fifty real robots. Fifty Lausanne trails were thus arranged in a line, with one robot per trail (an extension of the situation depicted in Figure 1). This was done virtually, i.e., each of the (fifty) individual robots was tested on one of the two real robots; nonetheless, they were (virtually) arranged along a line. The linear arrangement allowed us to localize the selection operation by using tournament selection [3,15]. Basically, a robot "sees" only those genomes of its neighbors which are within sensor range; in our current virtual tournament, each robot saw its neighbor to the immediate left and its neighbor to the immediate right. (Other neighborhood topologies—e.g., two-dimensional—are also possible.)

The local tournaments took place asynchronously (in a manner reminiscent of steady-state genetic algorithms): each robot, upon terminating an evaluation run (either due to a timeout or when the genome program has been executed), sent independently its calculated fitness along with its genome to its two neighbors—which transmitted their own fitness values and genomes in turn. Crossover then took place between the genome of the higher-fitness neighbor and the robot's own genome. This operation was performed in a standard manner by selecting a crossover point at random, and exchanging the genome parts after this point. The genome thus formed replaced the robot's old genome.

In the simulated-environment works [9,11], an individual was run for a given number of time steps, at the end of which its fitness was the number of food pellets amassed (thus, the maximal fitness value was 89—the number of food pellets along the trail). Such a fitness function was not possible in our case, since Romero could not distinguish between two pellets in a contiguous block. To overcome this problem, we used a different measure of fitness: the time (in seconds) spent over food "cells" (note that this time is proportional to the number of pellets amassed, provided the robot is on the move): $fitness = \sum_{i=1}^{blocks}(t_{out} - t_{in})$, where *blocks* is the number of contiguous food blocks the robot traverses, and $t_{out} - t_{in}$ expresses the traversal time for such a block.

The First Experiment and the Lessons Learned. Similarly to how Koza [11] defined the basic actions of the robot, we used four tokens—`LEFT (L)`, `RIGHT (R)`, `FORWARD (F)`, `NOP (N)`—representing elemental movements Romero is able to make. Furthermore, Romero had two innate (non-evolved) behaviors: (1) it traversed innately a contiguous block of food pellets; and (2) it traversed innately a single corner gap. The evolved behavior thus involved the decisions to be taken at the remaining possible gaps. Analysis of the Santa Fe trail revealed that only six possible behaviors existed upon encountering such a gap: `F, FF, FL, FR, FFL, FFR`. Based on these six basic behaviors we defined the following genetic "codons": `NNN, NNF, NFF, NFL, NFR, FFL, FFR`. A `NOP` command was added so as to render all codons of equal length. Romero's genome was defined straightforwardly as a sequence of codons, e.g., `–NNF–FFL–NNN–FFR–`. The seven codons thus formed the set of basic evolutionary building blocks.

The two robots were then subjected to a few days of experimenting on the real-world Lausanne trail. The most successful robots evolved amassed about 75% of the food "pellets." At first we thought that evolution did not have sufficient time to complete the task. We then verified that the simulation produced the same results.

So what was the problem? Nature is noisy! One can see that the linear sequence of instructions is extremely sensitive to noise. Executing the same genome in reality does not always result in the same path being taken. We concluded that this approach, though efficient in its genomic representation, was too simple for the real-world setup. One has to note that the codons contained an important amount of human knowledge about the environment. The evolutionary algorithm would certainly fail on another trail because of the codons being tailored for the Lausanne trail.

The Second Experiment and the Lessons Learned. We next looked for solutions less sensitive to noise and for alternative sensor inputs. One genomic representation we implemented was that of a rule-based system with a memory of two or three previous steps . The notion of a "step" had to be defined, since our trail is continuous, as opposed to the discrete, checkerboard-like simulated environments. Again, this is part of the real-world problem's difficulty.

The simulated robots were equipped with two additional sensors, one to the left and one to the right of the original centered infrared sensor. All three sensors faced the ground. At time t, the sensor input of these three sensors was stored in memory. The action (`N, L, R, M`) executed at time t depended on the current sensor inputs and on the past (i.e., memorized) two or three sensor inputs. The system can easily be implemented as a look-up table, each entry representing a rule (i.e., a basic action).

Results. This genomic representation worked very well with the simulated Santa Fe trail where the food pellets were removed when the robot passed over them. But it did not work with the simulated Lausanne trail! The reason is quite simple *a posteriori*: the food pellets could not be removed and thus the robot learned quickly to turn and increase its fitness *ad infinitum* on a small segment at the beginning of the trail.

The simulated Lausanne trail was used only to identify problems and critical factors as early as possible in the evolutionary process. For example, we did not transfer the above described experiment to the real-world setup because it did not even work in simulation. Enlarging the memory, adding more sensors, removing sensors, or changing the definition of the fitness did not help. The problem remained the same: the robot did not have sufficient knowledge about the environment, in particular about its position, in order to detect loops (which was not necessary in the original setup because the pellets were removed when the robot passed over them).

CONCLUSION

Is the simulated setup easier to solve than the real-world setup? No. The problems are different! The original Santa Fe trail and the Lausanne trail, *prima facie* equivalent, are in fact very different. The main difference is that the food pellets cannot be removed in the real-world setup. This makes the problem very different from the original Santa Fe food-collecting task.

Our experiments showed that the robot had to be well adapted to a given problem. Alas, to adapt a robot to a given environment, human knowledge (about the environment, e.g., its difficulties, its specialties, etc.) is required. How far should we go in this process, if we wish to study as open-ended and unconstrained an evolutionary scenario as possible? Should we start evolving the morphology of the robot? That is certainly an option.

The lack of information about the current position of the robot is the main reason why the algorithms did not work in our real-world setup: there was simply no means to detect loops.

From this standpoint, our robot is too simple. Adding an odometry device—even if inaccurate—would allow Romero to determine the approximate position in the environment and would thus allow to detect loops. One could say that the human knowledge (i.e., robot morphology, genomic representation) we put into the project at the beginning was not sufficient. In other words, the most critical factor in an evolutionary setup is still the system designer. What we currently call "evolution," "evolutionary algorithm," etc, are in fact a very small part in the complete design of a system.

A solution to our problem may be the following: one does not necessarily need an odometry system to determine an approximate position of the robot. A robot with a given sensorimotor system can learn a set of features and build an internal, cognitive map [2]. We propose to use a population of robots to explore in parallel the Lausanne trail and to collate the information gleaned about the environment so as to form a collective map, available to the entire population. The cognitive map will allow the robot to detect loops and thus to follow the Lausanne trail without passing several time over the same location.

References

[1] R. C. Arkin. *Behavior-Based Robotics*. The MIT Press, Cambridge, Massachusetts, London, England, 1998.

[2] A. Arleo, J. del R. Millán, and D. Floreano. Efficient Learning of Variable-Resolution Cognitive Maps for Autonomous Indoor Navigation. *IEEE Transactions on Robotics and Automation*, to appear.

[3] W. Banzhaf, P. Nordin, R. E. Keller, and F. D. Francone. *Genetic Programming–An Introduction: On the Automatic Evolution of Computer Programs and its Applications.* Morgan Kaufmann Publishers, San Francisco, 1997.

[4] V. Braitenberg. *Vehicles: Experiments in Synthetic Psychology.* MIT Press, Cambridge, MA, 1984.

[5] R. A. Brooks. The Whole Iguana. In M. Brady, editor, *Robotics Science*, pages 432–456. MIT Press, Cambridge, MA, 1989.

[6] R. A. Brooks. Artificial Life and Real Robots. In F. J. Varela and P. Bourgine, editors, *Toward a Practice of Autonomous Systems: Proceedings of the Firts European Conference on Artificial Life*, pages 3–10, Cambridge, MA, 1992. MIT Press/Bradford Books.

[7] S. G. Ficici, R. A. Watson, and J. B. Pollack. Embodied Evolution: A Response to Challenges in Evolutionary Robotics. In J. L. Wyatt and J. Demiris, editors, *Proceedings of the Eighth European Workshop on Learning Robots*, pages 14–22, 1999.

[8] B. Holldobler and E. O. Wilson. *The Ants.* Harvard University Press, Cambridge, Massachusetts, 1990.

[9] D. Jefferson, R. Collins, C. Cooper, M. Dyer, M. Flowers, R. Korf, C. Taylor, and A. Wang. Evolution as a Theme in Artificial Life: The Genesys/Tracker System. In C. G. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen, editors, *Artificial Life II*, volume X of *SFI Studies in the Sciences of Complexity*, pages 549–578, Redwood City, CA, 1992. Addison-Wesley.

[10] J. B. Knudsen. *Lego Mindstorms Robots.* O'Reilly & Associates Inc., 1999.

[11] J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* The MIT Press, Cambridge, Massachusetts, 1992.

[12] LEGO MindStorms. http://www.legomindstorms.com.

[13] F. Mondada, E. Franzi, and P. Ienne. Mobile Robot Miniaturization: A Tool for Investigation in Control Algorithms. In T. Yoshikawa and F. Miyazaki, editors, *Proceedings of the Third International Symposium on Experimental Robotics*, pages 501–513. Springer Verlag, 1993.

[14] E. M. A. Ronald, M. Sipper, and M. S. Capcarrère. Design, observation, surprise! A test of emergence. *Artificial Life*, 5(3), 1999. (to appear).

[15] A. Tettamanzi and M. Tomassini. Evolutionary algorithms and their applications. In D. Mange and M. Tomassini, editors, *Bio-Inspired Computing Machines: Toward Novel Computational Architectures*, pages 59–98. Presses Polytechniques et Universitaires Romandes, Lausanne, Switzerland, 1998.

[16] C. Teuscher, E. Sanchez, and M. Sipper. Romero: Un pèlerinage robotique à Santa Fe. In *11èmes Journées Jeunes Chercheurs en Robotique JJCR11*, pages 145–150. Swiss Federal Institute of Technology Lausanne, 8–9 avril 1999.

[17] C. Teuscher, E. Sanchez, and M. Sipper. Romero's Pilgrimage to Santa Fe: A Tale of Robot Evolution. In A. S. Wu, editor, *Workshop Proceedings of the Genetic and Evolutionary Computation Conference, GECCO'99*, pages 409–410, Orlando, Florida, USA, July 13–17 1999.

[18] R. A Watson, S. G. Ficici, and J. B. Pollack. Embodied Evolution: Embodying an Evolutionary Algorithm in a Population of Robots. In *Congress on Evolutionary Computation CEC'99*, volume 1, pages 353–342, Washington D.C., USA, July 6-9 1999. IEEE.