

Introducing Chemotaxis to a Mobile Robot

Christina Semertzidou¹, Nikolaos I. Dourvas¹,
Michail-Antisthenis Tsompanas^{1,2}, Andrew Adamatzky²,
and Georgios Ch. Sirakoulis^{1,2}✉

¹ School of Engineering, Democritus University of Thrace, Xanthi, Greece
{chriseme,ndourvas,mtsompan,gsirak}@ee.duth.gr

² University of the West of England, Bristol, UK
andrew.adamatzky@uwe.ac.uk

Abstract. This paper deals with the path planning problem of a robot in a maze based on a parallel chemotaxis bio-inspired model. The goal is the effective search of a route, which can connect the starting position of an autonomous robot with a final requested destination. To find this route the robot has to take under consideration its geometry, elements of its environment such as movements' restrictions by obstacles and other characteristics of the topology. Chemotaxis is a term found in biology and refers to the movement of an organism in response to a chemical stimulus. Among numerous examples of such biological form here we get inspiration by *Physarum polycephalum*, since this slime mold has shown the ability to find the shortest path in a maze between two spots, where chemicals exist, by following the gradient of the chemo-attractants. Inspired by this behavior, chemotaxis will be used here to lead a robot to its desired destination inside a labyrinth. A device transmitting signals can be considered as an equivalent chemical source and the robot's receiver will follow the increased gradient of signal's amplitude. Moreover, an effective model, that has the ability to simulate such a problem reducing the calculations' complexity and in the same time mimicking the specific behavior, namely Cellular Automata (CA) is coupled with chemotaxis. As a result, the design and implementation of a CA based bio-inspired algorithm is proposed and an E-Puck robot uses the exact algorithm to find the shortest path in different topologies as a proof of concept.

Keywords: Artificial intelligence · Cellular Automata · *Physarum polycephalum* · Bio-inspired algorithm · Shortest path problem

1 Introduction

One of the most well known and complex problems in the field robotics is to find the optimum path in a topology from one point to another. Many algorithms have been successfully proposed during the previous years to control the robots movement effectively and guide them to find the shortest path between two

specific spots in a topology [1–3]. In many cases the nature itself provides for centuries, solutions in similar problems. In this paper, the proposed algorithm is inspired by the chemotaxis operation, meaning the guidance of an organism to a specific place using chemical stimulus. Moreover we envisage modeling of chemotaxis through *Physarum*, a life form which has an extremely interesting behavior during its life cycle. It has been found that during its reproductive stage called plasmodium, takes a form that has the ability to find the minimum path between two food spots in a labyrinth. Moreover, it can perform all these acts without using any central nervous control system, a common characteristic as well to all single-celled creatures. Many bio-inspired models are based on this behavior [4, 5].

The algorithm that is used in this paper is inspired by the model that was proposed in [6] and modified accordingly in order to fit in the needs of the under study problem, which is the robot's guidance to an unknown environment in order to find the minimum path from one point to another. The mathematical model of chemotaxis and more specifically the diffusion equation of chemo-attractants is based on the parallel modeling tool, Cellular Automata (CA). CA can be assumed as an alternative form of microscopic reality, which supports the desired macroscopic behavior. The implementation of the proposed control algorithm has been applied on a E-Puck robot using programming environments that are created by the simulation program *Webots*. Different scenarios have been tested and presented, in order to preserve universality and effectiveness. Furthermore, a comparison is made with the results that are produced by MATLAB programming environment implementing the same algorithm in order to ensure their validity.

2 Cellular Automata Principles

A CA is characterized by the following ones:

- a regular lattice of cells covering a portion of a d -dimensional space;
- a set $C(\mathbf{r}, t) = (C_1(\mathbf{r}, t), C_2(\mathbf{r}, t), \dots, C_m(\mathbf{r}, t))$ of variables attached to each site \mathbf{r} of the lattice giving the local state of each cell at the time $t = 0, 1, 2, \dots$;
- a rule $R = (R_1, R_2, \dots, R_m)$ which specifies the time evolution of the states $C(\mathbf{r}, t)$ in the following way: $C_j(\mathbf{r}, t + 1) = R_j(C(\mathbf{r}, t), C(\mathbf{r} + \boldsymbol{\delta}_1, t), C(\mathbf{r} + \boldsymbol{\delta}_2, t), \dots, C(\mathbf{r} + \boldsymbol{\delta}_q, t))$, where $\mathbf{r} + \boldsymbol{\delta}_k$ designate the cells belonging to a given neighbourhood of cell \mathbf{r} .

The new state of the transition function during the time step $t + 1$ is depended only to the previous time step t . The update of a cell is calculated using the information of its neighbors at the previous time step. However, in some cases it is needed to consider bigger memory resulting to dependence of more previous time steps $t - 1, t - 2, t - 3, \dots, t - k$. The rule R is the same for all the cells and is applied simultaneously in each one of them creating a synchronous system. As a possible relaxization of CA original definition, the asynchronous updating of cell states can be also considered. Moreover, in regards

to the CA rules' homogeneity, spatial or even temporal inhomogeneities can be introduced by the appliance of different rule in some specific cells in the lattice. The boundary cells are considered such an example of spatial inhomogeneity. CA are used in many fields and are capable of forming efficient models [7–9].

3 Chemotaxis Example

For readability reasons, some simple principles of chemotaxis are going to be presented through the paradigm of a living organism, namely *Physarum*. *Physarum* is a slime mould and Nakagaki *et al.* [10] were the first to observe that the plasmodium of this slime mould changes its shape as it crawls over a plain agar gel. If food is placed in two certain spots, it puts out pseudopodia that connect those food spots using the minimum path. Adamatzky in [11] placed the plasmodium in one place of the maze and, simultaneously, placed one food source (FS) in another place of the maze, before the plasmodium covers all the maze. The laboratory experiments show that the plasmodium spreads its pseudopodia trying to reach the food. Simultaneously, the food, releases the chemo-attractants to any direction in the maze. When the plasmodium finds those chemo-attractants, it follows them to the source food forming the minimum distance path between its initial site and the food site. So the plasmodium solves the maze in one pass because it is assisted by a gradient of chemo-attractants propagating from the target food. In conclusion, compared to Nakagaki *et al.* approach where thickness of the tube is proportional to flow, i.e. feedback, in Adamatzky approach *Physarum* acts simply as a concurrent navigator and “half” of the computation is done by diffusing chemo-attractants.

4 The Proposed CA Algorithm that the Robot Utilizes

In order to describe the chemotaxis behavior of *P. polycephalum* and enclose it to the robot's “brain”, a bio-inspired CA algorithm has been designed when inspired by the original [6] model and modified according to the robot related short path. The study case is as follows: a robot is introduced into a specific place in the topology and a signal's transmitter is placed in another position of the maze representing the food spot. The signal is released inside the topology following the diffusion equation. When the sensors of the robot catch these “attractants”, the algorithm chooses the direction with the greater attraction. After that the robot acts, turns to the specified direction and moves to the CA cell with the maximum attractants value. To simulate this experiment, the area where it takes place is divided into a matrix of squares with identical areas and each square of the surface is represented by a CA cell. The state of the (i, j) cell at time t , defined as $C_{i,j}^t$, is equal to Eq. 1:

$$C_{i,j}^t = \{ObstacleExistence_{i,j}, F_{i,j}^t, Robot_{i,j}^t\} \quad (1)$$

$ObstacleExistence_{i,j}$ is a variable that indicates the type of the area represented by the corresponding (i, j) cell. The possible values of

$ObstacleExistence_{i,j}$ are 1 for a free site of the lattice or 0 for an inaccessible site of the topology. Furthermore, $F_{i,j}^t$ represents the value of attractants (0–100) at time t in the area corresponding to the (i, j) cell. Finally, $Robot_{i,j}^t$ is a one-bit variable, which illustrates if the (i, j) cell is included in the final path and if it takes the value 1 then the robot takes the decision to move in this CA cell. The type of neighborhood that was used in this CA model is the Moore neighborhood consisting of all the eight side and diagonal neighbors of the central cell.

The diffusion equation (Eq. 2) allows us to talk about the route of randomly moving particles in n dimensions.

$$\frac{\partial}{\partial t}c(\bar{x}, t) = D\nabla^2c(\bar{x}, t) \quad (2)$$

But CA is a modeling tool that is based on discrete space and time. For this purpose, the discrete diffusion equation is used to describe the spread of the signal's values produced by the transmitter (Eq. 3).

$$\begin{aligned} F_{i,j}^{t+1} = \{ & F_{i,j}^t + f1 [(F_{i-1,j}^t - f3 \times F_{i,j}^t) \\ & + (F_{i+1,j}^t - f3 \times F_{i,j}^t) \\ & + (F_{i,j-1}^t - f3 \times F_{i,j}^t) \\ & + (F_{i,j+1}^t - f3 \times F_{i,j}^t)] \} \end{aligned} \quad (3)$$

where $f1, f2, f3$ are the parameters that determine the attractants' speed of expansion. After a few time steps the attractants are released inside the topology. When the robot meets them, it follows them towards the transmitter creating a route, which is the path that the robot will follow. In particular, it searches which of its neighbors has the greater value of $F_{i,j}^t$. When this value is justified, the $Robot_{i,j}^{t+1}$ value of this neighbor is changed from 0 to 1, while the robot moves to the specific cell. This procedure is repeated until the robot reaches the transmitter's position, which releases the signal inside the topology.

5 Implementation and Results

For testing reasons the *E-puck* desktop mobile robot (Fig. 1) was selected according to its properties and characteristics as introduced by EPFL [12] and previous citing works [13].

In this aspect, a variable, namely *translation* used for the implementation of the bio-inspired algorithm, is responsible for the definition of the robot's position in the CA space. It comprises three subvariables, x, y, z corresponding to the directions of the Cartesian system. The parameter *rotation* indicates the turning angle of the object to a specific direction, according to the environmental stimuli. Variable *FloorTileSize* defines the size of the CA cell, which for testing

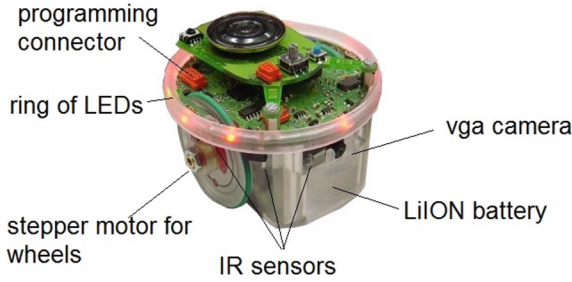


Fig. 1. The E-puck robot.

reasons is also provided in centimeters. More specifically, all simulations take into account a CA cell with side equal to 10 cm. The proposed size is chosen to fit to the exact E-Puck's dimensions which are not greater than 7.4 cm in length. Furthermore, *FloorSize* defines the size of the CA lattice and it is measured in meters. In order to reduce the calculations' complexity handled by E-puck, a 12×12 CA lattice is used. Another variable is *wallHeight*, which defines the height of the obstacles and is also measured in centimeters. For sake of simplicity and without loss of generality, in every simulation, the aforementioned variable takes the value 0.1. Finally, parameter *wallThickness* indicates how wide an obstacle, while in every simulation of this paper, this variable takes the value 0.074.

Three different testing topologies were used as test beds for the aforementioned CA algorithm in this paper. The first one is a topology free of obstacles (Fig. 2). The second experiment is evolved in a space with an obstacle at a specific place, depicted with the white color, (Figs. 3 and 4) and the final one is a more complex one, i.e. a maze (Fig. 5).

The robot and the food source are placed in different sites. Then the bio-inspired algorithm leads the E-puck to the transmitter using the shortest path every time. The transmitter is indicated by a cube with a *FOOD* caption on it. In order to verify the results, the same model, with the same topologies are designed in MATLAB environment.

In Fig. 2b the robot starts at coordinates (2,2) while the food source exists at (11,11). After 30 times steps the robot follows the minimum straight distance from the starting point to the final one. The robot uses the von Neumann neighborhood, consisting of solely side neighbors, and it can move only horizontally or vertically. Apparently, it will not be able to move to diagonal cells. So the 17 cells from the beginning to the ending position is the minimum distance. The same algorithm executed in MATLAB is presented in Fig. 2a. The red cells indicate the boundaries of the lattice. The black cell represents the starting position of the robot and the blue cell indicates the destination. The yellow line is the path that the robot follows. It is clear that the results are identical.

In Fig. 3b the robot starts at coordinates (2,2) and the food source exists at (11,11). However, this time, an obstacle exists inside the lattice creating a corner.

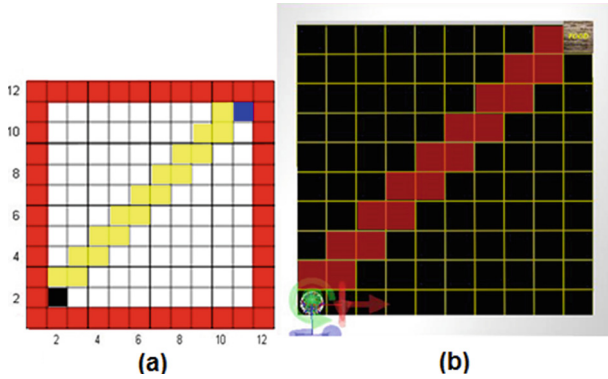


Fig. 2. Experiment in a lattice with no obstacles. (Color figure online)

The robot chooses to pass from the bottom of the obstacle towards its destination. This path consists of 17 cells. But the other minimum way from the left and upper side of the obstacle is also a distance of 17 cells and it is shown using the blue dots. As a result, these two routes are identical. The robot makes a random choice and follows the red path or the one in blue dots. After 48 times steps the robot moves on the minimum path from the starting point to the final one, avoiding the obstacle. The same algorithm executed in MATLAB is presented in Fig. 3a. Once again the results again identical.

In Fig. 4b the architecture of the topology is the same, using the same obstacle in the same position. But the initial conditions are now different. The robot starts at coordinates (2,11) and the food source exists at (11,5). After 28 times steps the robot moves on the minimum path from the starting point to the final one, avoiding the obstacle. This minimum path is the red one with 14 cells. The path passing through the other side of the obstacle is presented in blue dots. This

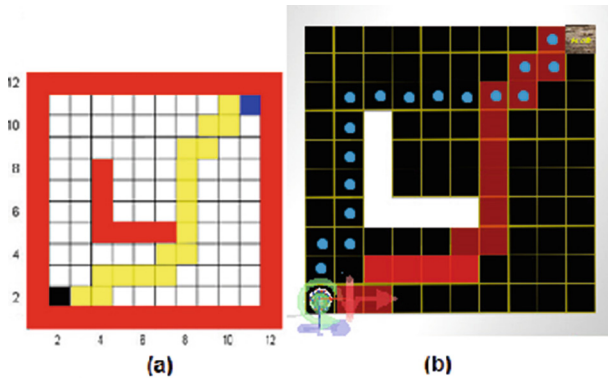


Fig. 3. Experiment in a lattice with obstacle. (Color figure online)

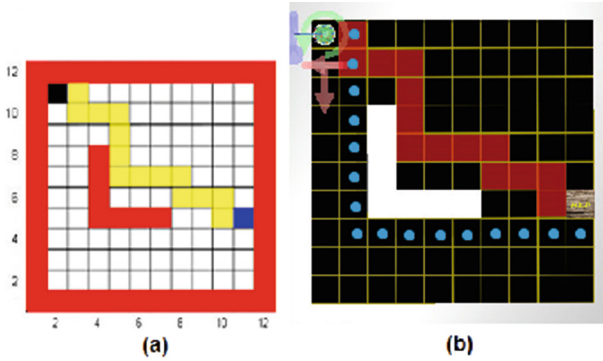


Fig. 4. Experiment in a lattice with obstacle and different starting positions. (Color figure online)

path is longer because it results to 16 cells. The robot never chooses this path. The same algorithm executed in MATLAB is presented in Fig. 4a. The results are again identical.

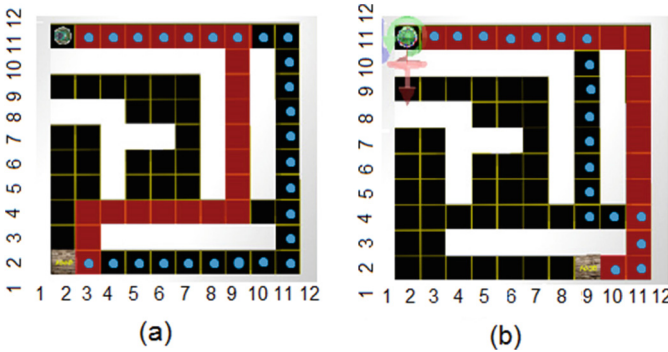


Fig. 5. (a) Experiment in a maze. (b) Experiment in a maze using different initial conditions. (Color figure online)

In Fig. 5a the robot starts at coordinates (2,11) and the food source exists at (2,2). In this situation, a more complicated experiment is evolved with many obstacles to avoid. After 140 times steps and 22 cells the robot reaches the food spot using the minimum path (red line) inside this maze. The path in blue dots present the other path that also leads from the initial position to the destination. But this path has 22 cells and this is the reason why the robot never chooses it. In Fig. 5b the topology of the maze is the same and the initial position of the food changes. In 98 time steps the robot passes through 19 cells and reaches the food using the minimum distance (red line). The line formed by blue dots is also

the minimum one because it also has 19 cells. After multiple experiments, it is proven that the robot chooses randomly the one or the other route.

6 Conclusions

There are many models trying to use chemotaxis and adapt it to their solutions. It is also proven to be a very powerful tool in order to find the minimum distance between two spots in a lattice. In this paper, a proposed chemotaxis bio-inspired CA model is used into a robot, and more specific in an E-Puck. This model tries to reproduce the chemotaxis behavior, which follows the gradient of chemo-attractants towards the food source using the minimum distance. The expansion of chemo-attractants is modeled using a discrete diffusion equation. The *Webots* programming environment is used to simulate the operations of an E-Puck inside the designed topologies. After applying this model, it is shown that the robot is always capable of finding the shortest path, between itself and an attractant source using its sensors and avoiding effectively the obstacles. A MATLAB implementation verifies the *Webots* results. The final product proposed is a robot, which has the ability to find solutions in a complex computational problem.

References

1. Lutvica, K., Velagic, J., Kadic, N., Osmic, N., Dzampo, G., Muminovic, H.: Remote path planning and motion control of mobile robot within indoor maze environment. In: IEEE International Symposium on Intelligent Control (ISIC), pp. 1596–1601 (2014)
2. Rahnama, B., Elci, A., Metani, S.: An image processing approach to solve labyrinth discovery robotics problem. In: IEEE 36th Annual Computer Software and Applications Conference Workshops (COMPSACW), pp. 631–636 (2012)
3. Boukas, E., Kostavelis, I., Gasteratos, A., Sirakoulis, G.C.: Robot guided crowd evacuation. *IEEE Trans. Autom. Sci. Eng.* **12**(2), 739–751 (2015)
4. Dourvas, N., Tsompanas, M.A., Sirakoulis, G.C., Tsalides, P.: Hardware acceleration of cellular automata physarum polycephalum model. *Parallel Process. Lett.* **25**(01), 1540006 (2015)
5. Kalogeiton, V.S., Papadopoulos, D.P., Georgilas, I.P., Sirakoulis, G.C., Adamatzky, A.I.: Cellular automaton model of crowd evacuation inspired by slime mould. *Int. J. General Syst.* **44**(3), 354–391 (2015). ISO 690
6. Tsompanas, M.-A., Sirakoulis, G.C.: Modeling and hardware implementation of an amoeba-like cellular automaton. *Bioinspir Biomimetics* **7**(3), 036013 (19 pp.) (2012). <http://iopscience.iop.org/article/10.1088/1748-3182/7/3/036013>
7. Tsompanas, M.-A., Kachris, C., Sirakoulis, G.C.: Optimization of shared-memory multicore systems using game theory and genetic algorithms on cellular automata lattices. In: IEEE 27th International Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2013, pp. 482–490. IEEE (2013)
8. Tsompanas, M.-A., Kachris, C., Sirakoulis, G.C.: Evaluating conflicts impact over shared last-level cache using public goods game on cellular automata. In: International Conference on High Performance Computing and Simulation (HPCS), 2013, pp. 326–332. IEEE (2013)

9. Tsompanas, M.-A., Sirakoulis, G.C., Karafyllidis, I.: Modeling memory resources distribution on multicore processors using games on cellular automata lattices. In: IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW), 2010, pp. 1–8. IEEE (2010)
10. Nakagaki, T., Yamada, H., Tóth, Á.: Intelligence: maze solving by an amoeboid organism. *Nature* **407**(6803), 470 (2000)
11. Adamatzky, A.: Slime mold solves maze in one pass, assisted by gradient of chemo-attractants. *IEEE Trans. NanoBiosci.* **11**(2), 131–134 (2012)
12. Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klaptoch, A., Martinoli, A.: The e-puck, a robot designed for education in engineering. In: Proceedings of the 9th conference on autonomous robot systems and competitions, vol. 1, pp. 59–65. IPCB: Instituto Politecnico de Castelo Branco. ISO 690 (2009)
13. Ioannidis, K., Sirakoulis, G.C., Andreadis, I.: Cellular ants: a method to create collision free trajectories for a cooperative robot team. *Rob. Auton. Syst.* **59**(2), 113–127 (2011)