DOI: http://dx.doi.org/10.7551/978-0-262-33027-5-ch052

Investigating a cellular automata model that performs three distance diffusion on a robot path planning

Gina M. B. Oliveira[1], Patrícia A. Vargas[2] and Giordano B. S. Ferreira[1,3]

[1]Universidade Federal de Uberlândia, Uberlândia, Brazil
[2]Heriot-Watt University, Edinburgh, United Kingdom
[3]Tufts University, Medford, MA, United States
gina@facom.ufu.br

Abstract

An improved cellular automata (CA) model is proposed and evaluated on a path planning problem for an autonomous robot. The objective is to construct a collision-free path from the robot's initial position to the goal by applying the improved CA model and environment pre-processed images captured during its navigation. CA rules are used to enlarge obstacles and to perform three distance diffusion. Goal distance is spread by a CA rule using the free cells. Distance diffusion spread two new metrics used for cells inside the enlarged obstacle regions. The new distances are then used to plan routes when the robot needs to pass inside enlarged obstacle regions. The algorithm performs a new path planning at each n steps of robot navigation using its current position. Our inspiration came from the possibility to mimic the cognitive behaviour of desert ants, which re-plan their path to the goal constantly in time intervals, using only local cues. An e-puck robot was used in simulated scenarios to evaluate the new CA based adaptive path planning model. The simulations show promising results on the single robot's performance confirming that the model could also be adapted for robot swarms.

Introduction

Path planning is one of the most studied tasks for autonomous mobile robots (Arkin, 1998) (Siegwart et al., 2011). The objective is to define a list of movements from an initial position to the goal. There is a lot of research on using for example "A^* like" techniques for path-planning (Hernández et al., 2011) (Phillips et al., 2014). Cellular Automata (CA) are totally discrete models and have been recently considered for path planning Behring et al. (2000), (Parker et al., 2003), (Ioannidis et al., 2011), (Ferreira et al., 2014). CA consist of a large number of simple components with local connectivity. Despite of the simplicity of their basic components, CA are able to solve very complex problems such as scheduling (Swiecicka et al., 2006) and cryptography (Oliveira et al., 2010).

This work proposes an improved CA model derived from (Behring et al., 2000) which was further explored in (Parker et al., 2003), (Tavakoli et al., 2008) and (Kostavelis et al., 2012). The results show investigations of the new model for a single robot. It is important to indicate that our main objective is not to propose a better method for path planning by comparing, for example, with A^* like algorithms (Hernández et al., 2011) (Phillips et al., 2014). Here the CA model is used to further investigate its intrinsic distributed feature performance in this particular robot application. Our inspiration came from the possibility to mimic the cognitive behaviour of desert ants using CA. These ants re-plan their path to the goal constantly in time intervals, using only local cues as there is no pheromone to guide them (Tinbergen, 1932) (Wystrach et al., 2014) (Collett et al., 2014). This work is a first step towards fully exploiting CA intrinsic parallelism.

The planning algorithm proposed here uses a CA to enlarge the obstacle cells initially identified by an environment pre-processed image that generates a grid map. It is beyond the scope of this paper to give a full account of the pre-processing image system. The reader should refer to (Behring et al., 2000) for further details.

The distance to goal (GD) is calculated by CA iterations for all the free cells starting from those closest to the goal up to the initial position. The major enhancements in the present work are: (i) the path planning is continuously applied while the robot is navigating at each *n* time steps; (ii) a new distance diffusion was incorporated to the model in which two new metrics are calculated for the enlarged obstacle cells, i.e. the *Next Free Cell Distance* (NFCD) and the *Next True Obstacle Distance* (NTOD). The first modification makes it possible to approximate the actual path performed by the robot to the ideal route calculated in the initial position, turning its final position closest to the goal. The second modification is performed by two new CA rules to diffuse the new distances, which are used to help the robot to escape from regions near to obstacles. The V-REP simulator (Robotics, 2015) was used to evaluate the new model through experimenting with a single e-puck robot (EPFL, 2015).

This paper is structured as follows: Section 2 reviews CA to path planning. The model proposed by Behring and collaborators (2000) is described in Section 3. Section 4

Gina M. B. Oliveira, Patrícia A. Vargas, Giordano B. S. Ferreira (2015) Investigating a cellular automata model that performs three distance diffusion on a robot path planning. Proceedings of the European Conference on Artificial Life 2015, pp. 271-278

describes the new model and shows the simulation results. Section 5 concludes the paper and suggests future work.

Cellular automata and path planning

Cellular automata are totally discrete dynamical systems composed by simple components with local interactions. Basically, a cellular automaton consists of two parts: the cellular space and the transition rule. Cellular space is a regular *d*-dimensional lattice of *N* cells, each one with an identical pattern of local connections to other cells. CA are characterized by a transition rule that determines which will be the next configuration of the lattice, from its current one. Cells interact locally in a discrete time *t*: the state of the cell *i* at time *t + 1* depends only on the states of its neighborhood at time *t* including cell *i*. Cells updating is usually performed in a synchronous way. Considering 2D lattices, spatial neighborhoods must be used as the Moore neighborhood, which is composed by the central cell and its eight immediate neighbors (Sarkar, 2000).

CA are able to represent high complex phenomena at the same time that they can be exactly simulated by digital processors due to its intrinsic discrete nature. Moreover, CA-based models can be efficiently executed by multiprocessors architectures due to its inherent parallelism. Finally, the decentralized CA architecture permits the development of high-distributed solutions (Swiecicka et al., 2006), (Oliveira et al., 2010). Those characteristics lead CA to be considered for robotics, more specifically for path planning (Shu and Buxton, 1995), (Marchese, 2002), (Behring et al., 2000), (Ioannidis et al., 2011), (Akbarimajd and Hassanzadeh, 2012) and (Ferreira et al., 2014). A classification scheme was presented in (Ferreira et al., 2014) which divides the previous works on CA applied to path planning into six approaches. In this work we will concentrate on investigating and improving a previous model from the distance diffusion planning approach.

Distance diffusion planning

(Behring et al., 2000) presented an approach that is one of the most cited path planning models using CA. The robot is considered as a single non-oriented point not subjected to kinematics and dynamic laws. The space is a discrete 2D space divided into square cells in such way that the robot occupies a unique cell. In a pre-processing phase, the algorithm receives an environment pre-processed image as input with obstacle positions, the robot initial position and the goal cell. The discrete environment is transformed into a CA lattice cell and each cell state has four possible values - Free, Obstacle, Initial and Goal.

The planning algorithm is divided into three phases. The first phase provokes the enlargement of the obstacles initially identified aiming not only at compensating the effects of the dynamics not considered in the model but to avoid obstacle collisions during the robot navigation. The CA transition rule that enlarges the obstacles is defined by:

Rule Enlargement:

If the cell is *Free* and one of its neighbors is *Obstacle*, **then** the next value of the central cell is *Obstacle*;

This rule is applied for a fixed number of time steps (x), which depend on the cellular space resolution and robot size. Behring and colleagues used $x = 4$, which results in an enlargement of 4 cells in obstacles' thickness.

A CA rule on the second phase spreads the Goal Distance (GD) for each free cell. Free cells states change to integer numbers, which are the distance from the cell to the goal in terms of robot's steps. The goal cell changes its state to 0. The CA rule only updates free cells states and is defined by

Rule Goal Distance:

If the central cell is *Free* and one of its neighbors is v, **then** the next value of the central cell is $v + 1$;

Applying this CA rule for some steps, GD is calculated for all free cells spreading from the goal to the robot's neighbors, if there is a collision-free route from the initial cell to the goal. If this route does not exist, the spreading will be stopped before the initial cell is reached. Figure 1 shows an example of goal distance spreading. The last phase uses GD values to construct a collision free route starting from the initial position of the robot until the goal cell, decrementing 1 in the value of GD at each step. Figure 1.b shows an example of a planned path.

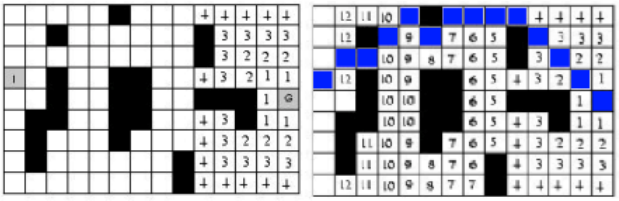

Figure 1: GD diffusion and planned route adapted from (Behring et al., 2000)

Adaptive path planning model

Simulations showed a significant difference between the planned route and the actual one performed by the e-puck robot using the original model described in (Behring et al., 2000). An example of such situation is presented in Figure 2.a, which indicates the route planned in the start of simulation (in green) and the actual path traveled over by the e-puck (in red).

The original model was modified to overcome this first critical point, in a way that the algorithm would now per-

Gina M. B. Oliveira, Patrícia A. Vargas, Giordano B. S. Ferreira (2015) Investigating a cellular automata model that performs three distance diffusion on a robot path planning. Proceedings of the European Conference on Artificial Life 2015, pp. 271-278

form new distance diffusion at each n time steps. Using this strategy, even if the robot starts to diverge from the expected route, a new path planning will be executed, correcting the deviation. For such, the method needs to receive a pre-processed new image at each n time steps to restart the path planning using the current robot position. An adequate value for n can be determined according to the latency of the image processing system. All simulations were carried out using $n = 5$. Figure 2.b shows the planned path and actual one performed using the distance diffusion at each 5 steps for the same initial scenario of Figure 2.a. One can see that although the robot's route performed with the adaptive planning is not similar to the original planned path, after correcting its route, the robot's final position is very close to the planned goal. This modification turns the new model into an adaptive path planning, instead of a static planner as (Behring et al., 2000).

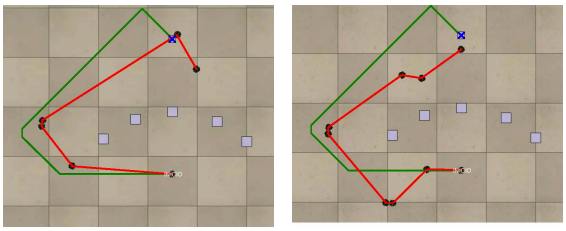

Figure 2: Planned route (in green) and the observed navigation (in red): (a) Behring et al.s model (b) New adaptive planning model.

The second critical point identified in the original model refers to situations in which the algorithm cannot plan a path because the robot is initially positioned inside enlarged obstacle cells. The main reason why the algorithm in (Behring et al., 2000) cannot plan a valid path in this case is because all enlarged obstacle cells are treated as true obstacles and the CA diffusion rule is not applied to them in the second phase. Therefore, GD spreading does not achieve the initial position and the path cannot be constructed. This situation can happen quite often from the initial scenario, i.e. when the plan is applied for the first time. However, this kind of case is more likely to occur when using the new adaptive path planning at each n steps. This is due to the fact that several times the original route passes in the limit of enlarged regions and during navigation the robot invades such areas.

In order to overcome this second critical point we first considered applying GD spreading also in the enlarged obstacle cells, using higher values to these cells to avoid using them by the route planner. However, if the distance to the goal is also used in these regions, when the routes needed to be calculated inside them, the planned route must approximate to the true obstacles, increasing the risk of collisions if they make the robot closer to the goal. Consequently GD metric was discarded as it cannot be used in such regions.

Two new measurements were then calculated using CA rules and both were applied only to obstacle cells. The first indicates the shortest distance of an obstacle cell to a true obstacle, i.e. Next True-Obstacle Distance (NTOD). The second measurement indicates the shortest distance of an obstacle cell to a free one and it is calculated only for enlarged obstacle cells. It starts from a relatively high limit (V) and it increments in one unit as the distance also increments. We call it the Next Free-Cell Distance (NFCD). Both metrics are used to construct a route that takes the robot to free cells as fast as possible, but also takes into account the need to pass as distant as possible from the real obstacles.

Each cell state is composed by a triple of sub-states $(s1, s2, s3)$: **s1** is a categorical sub-state with four possible values - Free, Obstacle, Initial and Goal; **s2** is a numeric sub-state that store the distance between a cell and a true obstacle; **s3** is a numeric sub-state that can store the distance between a cell and the goal (if the cell is free) or the distance between a cell and the free one (if the cell is an enlarged obstacle). Note that sub-states $s2$ and $s3$ are finite values which maximum values depend on environment grid dimensions (NM cells) being that the maximum possible values are N+M (the maximum Manhattan distance). Moreover, this CA model uses a fixed boundary condition, since the extreme/border cells of the 2-dimensional lattices is considered to have obstacle cells (the wall) on all their non-defined neighbors.

The planning algorithm is applied at each n steps of the robot locomotion and it is divided in a pre-processing and four major phases. It uses CA rules with Moore neighborhood (Sarkar, 2000) in the first three phases. In the pre-processing phase, the algorithm receives an environment pre-processed image as input, the robot initial position and the goal cell. It then establishes the initial value of s1 for each cell of the lattice. The new proposed algorithm is described as follows.

Phase 1 - causes the enlargement of the obstacles initially identified and spreads the value of next true-obstacle distance (NTOD) at same time. NTOD is stored at sub-state $s2$. It starts establishing that $s2 = 0$ for all obstacle cells by applying the following rule one:

Rule NTOD Initialization:

If the central cell has $\{s1 = Obstacle\}$, **then** the next value of the central cell is $s2 = 0$.

The CA rule transition enlarges the obstacles and updates the values of sub-states $s1$ and $s2$ and it is defined by:

Rule Enlargement and NTOD Diffusion:

If the central cell has $\{s1 = Free$ and one of its neighbors has $s1 = Obstacle$ with $s2 = v\}$, **then** the next values of $s1$

Gina M. B. Oliveira, Patrícia A. Vargas, Giordano B. S. Ferreira (2015) Investigating a cellular automata model that performs three distance diffusion on a robot path planning. Proceedings of the European Conference on Artificial Life 2015, pp. 271-278

and $s2$ of the central cell are *Obstacle* and $v+1$, respectively.

This rule is applied for a fixed number of time steps (x), which results in an enlargement of x cells in obstacles' thickness and in the calculus of NTOD for these cells. The behavior of this CA rule can be observed in Figure 3.

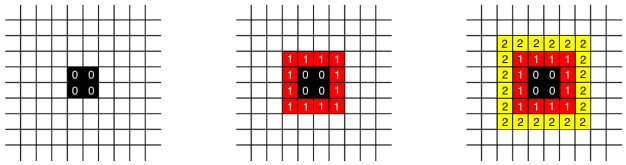

Figure 3: Obstacle enlargement and NTOD diffusion (x = 2).

Phase 2 - uses a CA rule to spread the goal distance (GD) into free cells, as in (Behring et al., 2000), except that now they are stored in sub-state $s3$. At the start of second phase we have that all free cells, obstacle cells and the cell with the robot ($s1 = Free$ or $s1 = Initial$ or $s1 = Obstacle$) have $s3$ undefined. The initialization rule is applied once and it establishes that the goal cell will assume $s3 = 0$:

Rule GD Initialization:

If the central cell has $\{s1 = Goal\}$, **then** the next value of the central cell is $s3 = 0$.

The transition rule updates $s3$ value and it is applied only to free cells by some time steps until the state of $s3$ is not modified for any cell in the lattice. The CA rule transition is defined by:

Rule GD Spread:

If the central cell has $\{s1 = Free$ and $s3$ is undefined and one of its neighbors has $s3 = v\}$, **then** the next state of the central cell is $s3 = v + 1$.

Applying this CA rule for some steps, GD (stored in $s3$) is calculated for all free cells. The resultant calculus is the same illustrated in Figure 1, except that is the new model it represents the $s3$ spreading.

Phase 3 - uses a CA rule to spread the next free-cell distance (NFCD) through obstacle cells, which will also be stored in sub-state $s3$. This phase will only be applied if at the end of the second phase the value of $s3$ is still undefined for neighbor cells of the robot's initial position. In other words, it is applied when the GD spreading finished but the initial position was not reached. In this case, it is probable that the robot is inside an enlarged obstacle area or there is a barrier with enlarged obstacle cells between the goal and the initial position. As a result, the value of next free-cell distance will be calculated for the enlarged obstacle cells and it will be stored in sub-sate $s3$ of such cells. At the end of the second phase, all obstacle cells have undefined $s3$. An initialization rule is applied to start the value of the obstacle cells closest to free cells. This rule is applied to all lattice cells:

Rule NFCD Initialization:

If the central cell has $\{s1 = Obstacle$ and one of its neighbors has $s1 = Free\}$, **then** the next value of the central cell is $s3 = V$.

V is an arbitrary discrete value chosen to be bigger than any goal distance calculated to free cells. For instance, we chose $V = 100$ because it is impossible to have $GD > 99$ due to the lattice dimensions used in our experiments. After using this rule to initialize $s3$ values for the obstacle cells in the border of the enlarged regions, the algorithm applies a CA rule to spread next free-cells distance (NFCD). This rule transition is defined by:

Rule NFCD Spread:

If the central cell has $\{s1 = Obstacle$ and $s2 > 0$ and $s3$ is undefined and at least one neighbor has $s3$ defined$\}$, **then** the next value of the central cell is $s3 = vmin + 1$, where $vmin$ is the minimum value of $s3$, considering all the neighbors of the central cell.

Applying this CA rule for some steps, the NFCD stored in $s3$ is calculated for all obstacle cells with $s2 > 0$ (which excludes the true obstacles). Figure 4 illustrates the application of the CA rule for 7 time steps, until all the cells with $s1 = Obstacle$ and $s2 > 0$ have the value of $s3$ calculated. Black cells indicate the true obstacles, i.e. the cells identified in the image processing as obstacles. The red, yellow and green ones indicate the enlarged obstacle cells (x equal to 1, 2 and 3, respectively). The robot's initial position is a yellow cell ($x = 2$). In Figure 4.a one can observe $s3$ values after the initialization rule is applied to identify cells at the borders (i.e. obstacle cells with at least one free cell as neighbor). These cells assume $s3 = V$ ($V = 100$ in this example). In Figure 4.b one can observe the value of $s3$ after applying the CA rule once. Figure 4.c shows the final values after NFCD diffusion.

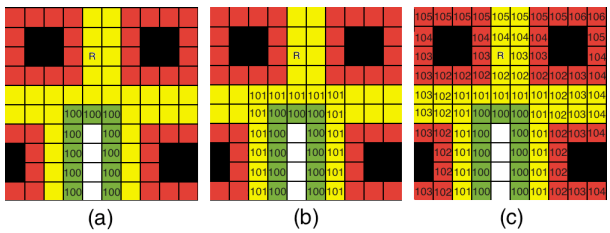

Figure 4: NFCD diffusion: a) initialization $t = 0$, b) $t = 1$ c) Final

Gina M. B. Oliveira, Patrícia A. Vargas, Giordano B. S. Ferreira (2015) Investigating a cellular automata model that performs three distance diffusion on a robot path planning. Proceedings of the European Conference on Artificial Life 2015, pp. 271-278

Phase 4 - constructs the final path between the robot's current position and the goal. The last phase use all the spread distances to choose a collision free route. The planned route is calculated by either the *DR Procedure* or *EDR Procedure* explained below:

DR Procedure: Constructing just Descending Routes

If the third phase was not necessary, the planned path will be calculated as the original model: any route starting from initial position of the robot until the goal cell, using only free cells, decrementing 1 in the value of $s3$ (GD) at each step of the path. In the case of draw when choosing the next free cell (that is, there are at least two neighbors with a decreasing GD), we implemented some options of conflict decisions: random choice, or chosen the next in a predefined order (clockwise or anticlockwise) or even chosen the next cell that not provokes a robot rotation (that is, it keeps the current robot's orientation). The conflict decision is a parameter of the algorithm.

EDR Procedure: Mixing Escaping and Descending Routes

If the third phase was applied, the planned route is divided in two parts: the first one is obtained using the $s2$ and $s3$ values of obstacle cells (NTOD and NFCD) and the second part is obtained using the $s3$ values of free cells (GD). The initial part of the path must be a route to move the robot from the enlarged obstacles region as fast as possible (or the "escape route"). As such, the algorithm starts from the initial position cell and choose as the next step the neighbor cell with the smallest $s3$ value (i.e. the neighbor with the smallest NFCD). If there is a draw involving neighbors, the algorithm chooses the neighbor with the highest $s2$ (NTOD) value (i.e. between the tied cells it chooses the one more distant from true obstacles). If the draw persists, conflict decisions criteria can be applied, as explained in DR Procedure. The next steps of the route are defined in this way until the algorithm find a cell with $NFCD = V$, which indicates that it is at a border cell in respect to the obstacle regions. The next step of the path will be the neighbor free cell with the smallest GD. Starting from this point, GD stored in $s3$ for free cells will be used to construct the second part of the path, that is, the "descending route".

Figure 5 presents an example of path generated by simulating the new planning algorithm. It shows the complete route formed by the junction of the escape route (in purple) from the cul-de-sac scenario and the descending route (in blue) toward to the goal. Figure 6 shows a summary of the new path planning model algorithm described in this section. It is applied at each n time steps.

One may argue that the escaping route calculus is not necessary, and a simple solution to this problem could be associating high costs (GD) to cells close to obstacles excluding Phase 3 of the algorithm and simplifying Phase 4. Indeed if this approach is used the descending route will be generated as in the new algorithm. The difference in such situation is that the descending route is calculated here in a direct way, just disregarding the closest cells in the calculus. On the other hand, considering critical situations where the robot is positioned within a close distance to the obstacles, the use of the distance to the next free cell for calculating escaping routes makes it a crucial difference. The calculus focuses on keeping the robot away from the enlarged region in the smallest number of steps possible. In that way GD is ignored and an optimal route based on NFCD is found. Besides, several optimal rules (NFCD) are expected since there are several free cells. Thus, the new algorithm uses a second measure (NTOD) to choose between the NFCD optimal routes one with the maximum distance to obstacles. As a result, it finds short escaping routes passing on a safe distance from obstacles.

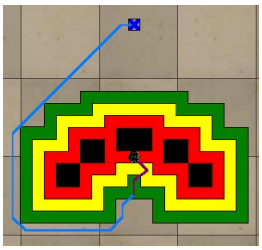

Figure 5: A complete path in a V-REP simulation generated using the new algorithm: escape route (in purple) and descending route (in blue). A blue x represent the goal.

Simulation and Results

A series of simulations were performed aiming at analyzing the behavior of the robot with the proposed adaptive path planning algorithm. Figures 7 shows two simulations using a scenario with 8 obstacles. Figure 7.a illustrates the robot path from (Behring et al., 2000) and Figure 7.b the new path using the adaptive path planning algorithm. A yellow "x" represents the starting point and a blue "x" represents the goal.

In the first simulation (Figure 7.a), the robot final position is very distant from the goal. On the other hand, the e-puck could reach the goal when the new model was used (Figure 7.b). One can see that in the first navigation, the robot has passed closest to obstacles, while in the second one, the robot kept a safe distance from them. In fact, in both simulations the e-puck invaded the enlarged obstacle region ($x = 4$). However, while using the original model the robot continued its route getting even closer to the obstacles.

By using the adaptive planning model, the e-puck robot was able to identify this invasion and to generate an escaping route, correcting its navigation to avoid the obstacles. The green line in Figure 7.b indicates a point in which the replanning was needed.

Gina M. B. Oliveira, Patrícia A. Vargas, Giordano B. S. Ferreira (2015) Investigating a cellular automata model that performs three distance diffusion on a robot path planning. Proceedings of the European Conference on Artificial Life 2015, pp. 271-278

- Inputs:
  M: a grid map with dimension *W*× *L*, *where each position* M_{wl} is 0 or 1;
  G=(X_g,Y_g): *coordinates of the goal in M*
  I=(X_i,Y_i): *coordinates of the robot's initial position in M*
  x: *number of steps of obstacle's enlargement*
- Output:
  P: a sequence of cells M_{wl} from *I* to *G*.
- Description:

- Pre-processing phase:
1. Construct *S* a 2D matrix W×*L* where each position/cell S_{wl} in S is a triple (s1,s2,s3)
2. Initialize each cell S_{wl} = (s1,s2,s3) in a such way that:
   - s1= Free, if M_{wl} = 0 and G≠(w,l) and I≠(w,l);
   - s1= Goal, if M_{wl} = 0 and G=(w,l);
   - s1= Initial, if M_{wl} = 0 and I=(w,l);
   - s1= Obstacle, if M_{wl} = 1;
   - s2 and s3 starts undetermined;

- Phase 1:
1. Apply once the ***NTOD Initialization*** rule for each cell S_{wl}.
2. Apply transition rule ***Enlargement and NTOD Diffusion*** by *x* steps for each cell S_{wl}.

- Phase 2:
1. Apply once the ***GD Initialization*** rule for each cell S_{wl}.
2. Apply ***GD Diffusion*** transition rule by several steps for each cell S_{wl} until the value of s3 is not modified for any cell in *S*.

- Phase 3:
This phase is applied only if s3 in the initial position I is still undefined after Phase 2.
1. Apply once the rule ***NFCD Initialization*** for each cell S_{wl}.
2. Apply transition rule ***NFCD Spread*** by several steps for each cell S_{wl} until the value of s3 is not modified for any cell in *S*.

Phase 4:
1. If the Phase 3 was not applied, construct the final path *P* using **DR Procedure.**
2. If the Phase 3 was applied, construct the final path *P* using **EDR Procedure.**

Figure 6: Summary of the adaptive path planning model algorithm.

It should be noted that several obstacle collisions were observed during simulations using the original model. Moreover, the robot made several rotations and steps to achieve the goal. On the other hand, using the adaptive planning model the robot performed a smooth navigation, going around the obstacles while keeping a safe distance.

Aiming at better exploring the differences between both models we performed a series of experiments using the scenario presented in Figure 7. Since simulations are non-deterministic, each algorithm was run for 30 simulations in the same scenario and some statistical results were obtained. In comparison, the average of the total distance navigated (in meters) was $\{8.33;7.82\}$. The first value represents the original model, whereas the second value represents the new model respectively. The average of the final distance to the goal (in meters) was $\{0.97;0.03\}$. The average of the total navigation time (in seconds) was $\{382;411\}$ and the number of collisions observed during simulations was $\{7;0\}$. The final distance indicates that the robot stopped very distant from the goal when the original method is employed. We could observe 7 collisions in 30 runs when using the original model. One can also observe that the adaptive planning

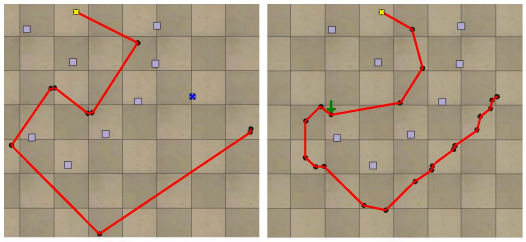

Figure 7: (a) Behring's model and (b) Proposed model.

did not cause a severe impact on the total time: on average 8% of increase in both scenarios.

We created another 9 simulation scenarios with a different number of obstacles (Figure 8) and also made a comparative analysis using 30 simulation runs in each scenario. The new adaptive model performance presented: (i) an average reduction of 82% relative to the final distance to the goal (at least 70% of reduction was observed in any scenario); (ii) an average reduction of 5.8% relative to the total navigation distance (in all simulations a decrease was observed); (iii) an average increase of 7% in the total navigation time (in all scenarios the time was increased, but none above 10%) and (iv) no collisions. Therefore, we can concluded based on the results of 300 simulation runs (30 for each scenario) that by applying the new model it was possible to substantially reduce the final distance between the robot and the goal with an additional reduction on the total navigated distance and avoiding obstacles and a very small increase in the total navigation time.

It is important to highlight that the re-planning strategy is used here to correct a control problem since it was possible to observe in our simulations that the robot is not able to navigate as planned. Indeed, one could expect a difference between the planned and the executed route even if more sophisticated control strategies were employed. Using re-planning is a step towards a solution to some dynamic environments where re-planning might be necessary (Wystrach et al., 2014) (Collett et al., 2014).

Discussion

Path planning is a very relevant task for robot autonomous navigation and different techniques have been studied so far (Siegwart et al., 2011). This work investigates an adaptive path planning model based on cellular automata using three different metrics: Goal Distance (GD), Next Free Cell Distance (NFCD) and Next True Obstacle Distance (NTDO). A new plan is constructed at each n steps, starting from the robot's current position, which is extracted from an environment pre-processed image (Behring et al., 2000). The path can be composed by descending and escaping routes whenever is necessary. Descending routes are generated using only GD. Escaping routes are generated using NFCD and

Gina M. B. Oliveira, Patrícia A. Vargas, Giordano B. S. Ferreira (2015) Investigating a cellular automata model that performs three distance diffusion on a robot path planning. Proceedings of the European Conference on Artificial Life 2015, pp. 271-278

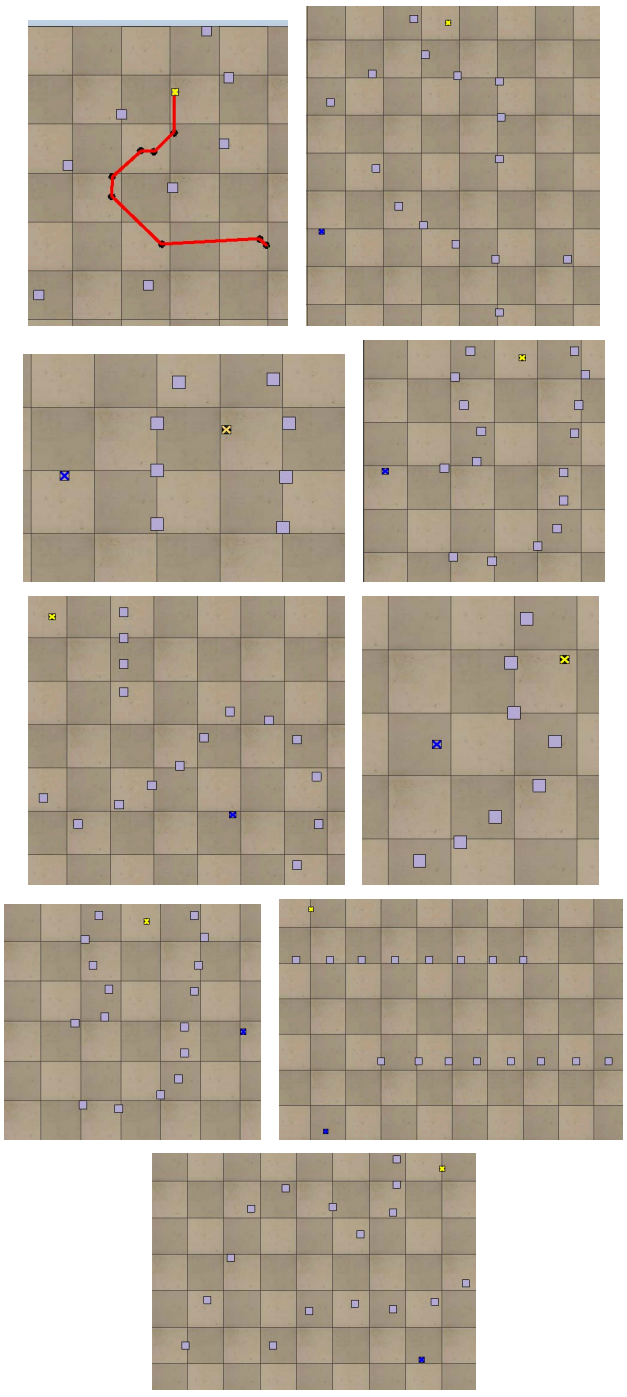

Figure 8: Additional 9 scenarios used in V-REP simulations.

NTOD by avoiding passing too close to the obstacles.

Distance diffusion performed by CA rules returns similar results to the ones obtained by wavefront algorithms in path planning (Zelinsky et al., 1993). An advantage of using CA to perform diffusions is that they are very decentralized structures which enable us to implement them in a parallel way, making them appropriate, for example, to massive parallel architectures as FPGAs (Halbach and Hoffmann, 2004). Moreover, CA performs information spreading in a bottom-up approach instead of a sequential standard wavefront algorithm. This way of modeling enables us in this work to mix different information (GD, NFCD and NTOD metrics) to construct routes with different requisites.

Therefore, GD calculus finds optimal descending paths. Considering escaping routes, the algorithm always finds optimal routes in respect to NFCD and between all possible optimal NFCD routes one with the maximum NTOD. Thus, the escaping route is a point of the Pareto front considering the bi-objective problem (NFCD and NTOD). The algorithm complexity is dominated by GD propagation both in the previous work (Behring et al., 2000) and in the new proposition. Therefore, the full re-planning has the same asymptotic complexity of the previous model. On the other hand, using static environments as mentioned before, at each n steps just phase 4 - which has a simpler complexity than GD propagation (phase 2) - is needed to be performed. This simple complexity analysis justify why we did not observe a significant difference considering the total navigation time comparing the previous model (Behring et al., 2000) and our new proposition.

It is important to point out that full re-planning is not necessary in static environments. Indeed the robot does not do full re-planning, only a new route is calculated given the robot's current position. GD, NFCD and NTOD metrics are not necessary to be recalculated at each n steps, if the environment does not change. One can compute all distances in advance because they are independent of the robot current position.

Conclusion

This current work improves previous research on CA models and path planning problems for a single robot (Behring et al., 2000). In this work we investigated the behaviour of a single robot in a static scenario. Using the V-REP simulator, the current environment configuration (including robot's position), can be easily obtained using the software functionalities. For an effective analysis of the new adaptive model in real robots, it is also necessary to improve the preprocessing image system and investigate the performance on a real robot.

This work is the first step towards merging the proposed model with previous successful models for cooperative robot swarms (Ioannidis et al., 2011) (Ferreira et al., 2014) creating a hybrid CA model that could use local and distributed

Gina M. B. Oliveira, Patrícia A. Vargas, Giordano B. S. Ferreira (2015) Investigating a cellular automata model that performs three distance diffusion on a robot path planning. Proceedings of the European Conference on Artificial Life 2015, pp. 271-278

adaptive path planning tacking similar but more challenging dynamic problems. In static scenarios, few cells change their state at each propagation step. However, the processing performed in each cell is very simple, which turns our algorithm a candidate to be implemented in massive distributed architectures as FPGA that could be incorporated as a dedicated hardware. Our main goal is to apply a CA that would resemble the desert ants behaviour where re-planning is a necessary condition for survival in a dynamic environment (Wystrach et al., 2014) (Collett et al., 2014).

Future work includes: (1) merging this new model with already established models for cooperative robots (Ioannidis et al., 2011) (Ferreira et al., 2014) plus the use of real e-puck robots; (2) using evolutionary robotics techniques (Vargas et al., 2014) to adjust the parameters involved in the model, as the value of n (number of steps to re-planning) and others related to the implementation of the robot's dynamics, and finally (3) adapting the new model to planning paths in environments with mobile obstacles. In fact, our adaptive path-planning model is a step toward dealing with dynamic scenarios and multiple robots.

Acknowledgements

GMBO is grateful to CNPq, CAPES and Fapemig support.

References

Akbarimajd, A. and Hassanzadeh, A. (2012). Autonomously Implemented Versatile Path Planning for Mobile Robots Based on Cellular Automata and Ant Colony. *International Journal of Computational Intelligence Systems*, 5(1):39–52.

Arkin, R. C. (1998). *Behavior-based robotics*. MIT press.

Behring, C., Bracho, M., Castro, M., and Moreno, J. A. (2000). An Algorithm for Robot Path Planning with Cellular Automata. In *Proceedings of the Fourth International Conference on Cellular Automata for Research and Industry: Theoretical and Practical Issues on Cellular Automata*, pages 11 – 19.

Collett, T. S., Lent, D. D., and Graham, P. (2014). Scene perception and the visual control of travel direction in navigating wood ants. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 369(1636):20130035.

EPFL (2015). E-puck education robot. http://www.e-puck.org.

Ferreira, G. B., Vargas, P. A., and Oliveira, G. M. (2014). An improved cellular automata-based model for robot path-planning. In *Advances in Autonomous Robotics Systems*, pages 25–36. Springer.

Halbach, M. and Hoffmann, R. (2004). Implementing cellular automata in fpga logic. In *Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International*, page 258. IEEE.

Hernández, C., Sun, X., Koenig, S., and Meseguer, P. (2011). Tree adaptive a*. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 123–130. International Foundation for Autonomous Agents and Multiagent Systems.

Ioannidis, K., Sirakoulis, G. C., and Andreadis, I. (2011). A Path Planning Method Based on Cellular Automata for Cooperative Robots. *Applied Artificial Intelligence*, 25(8):721–745.

Kostavelis, I., Boukas, E., Nalpantidis, L., and Gasteratos, A. (2012). Path Tracing on Polar Depth Maps for Robot Navigation. *Cellular Automata*, pages 395–404.

Marchese, F. M. (2002). A directional diffusion algorithm on cellular automata for robot path-planning. *Future Generation Computer Systems*, 18(7):983–994.

Oliveira, G. M., Martins, L. G., Ferreira, G. B., and Alt, L. S. (2010). Secret key specification for a variable-length cryptographic cellular automata model. In *Parallel Problem Solving from Nature, PPSN XI*, pages 381–390. Springer.

Parker, L. E., Birch, B., and Reardon, C. (2003). Indoor target intercept using an acoustic sensor network and dual wavefront path planning. In *In Proceedings of IEEE International Symposium on Intelligent Robots and Systems (IROS 03*, pages 278–283.

Phillips, M., Likhachev, M., and Koenig, S. (2014). Pa* se: Parallel a* for slow expansions. In *Proceedings of the Twenty-Fourth International Conference on Automated Planning and Scheduling*, pages 208–216.

Robotics, C. (2015). V-rep, virtual robot experimentation platform. http://www.coppeliarobotics.com.

Sarkar, P. (2000). A brief history of cellular automata. *ACM Computing Surveys (CSUR)*, 32(1):80–107.

Shu, C. and Buxton, H. (1995). Parallel path planning on the distributed array processor. *Parallel Computing*, 21(11):1749–1767.

Siegwart, R., Nourbakhsh, I. R., and Scaramuzza, D. (2011). *Introduction to autonomous mobile robots*. MIT press.

Swiecicka, A., Seredynski, F., and Zomaya, A. Y. (2006). Multiprocessor scheduling and rescheduling with use of cellular automata and artificial immune system support. *Parallel and Distributed Systems, IEEE Transactions on*, 17(3):253–262.

Tavakoli, Y., Javadi, H. H. S., and Adabi, S. (2008). A cellular automata based algorithm for path planning in multi-agent systems with a common goal. *International Journal of Computer Science and Network Security. v8*, pages 119–123.

Tinbergen, N. (1932). Ueber die orientierung des bienenwolfes (philanthus triangulum fabr.). *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, 16(2):305–334.

Vargas, P. A., Di Paolo, E. A., Harvey, I., and Husbands, P. (2014). *The horizons of evolutionary robotics*. MIT Press.

Wystrach, A., Philippides, A., Aurejac, A., Cheng, K., and Graham, P. (2014). Visual scanning behaviours and their role in the navigation of the australian desert ant melophorus bagoti. *Journal of Comparative Physiology A*, 200(7):615–626.

Zelinsky, A., Jarvis, R., Byrne, J., and Yuta, S. (1993). Planning paths of complete coverage of an un- structured environment by a mobile robot. In *Proceedings of International Conference on Advanced Robotics*, pages 533–538.

Gina M. B. Oliveira, Patrícia A. Vargas, Giordano B. S. Ferreira (2015) Investigating a cellular automata model that performs three distance diffusion on a robot path planning. Proceedings of the European Conference on Artificial Life 2015, pp. 271-278