Path planning self-learning Algorithm for a dynamic changing environment

Rafiq Ahmad [1, 2, a], Peter Plapper [1]

[1] FSTC-RUES, 6 rue Coudenhove-Kalergi, University of Luxembourg, Luxembourg
[2] CECOS University of IT and Emerging sciences, Peshawar, Pakistan

Abstract. Safe and optimal path planning in a cluttered changing environment for agents' movement is an area of research, which needs further investigations. The existing methods are able to generated secure trajectories, but they are not efficient enough to learn from their mistakes, especially when dynamics of the environment are concerned. This paper presents an advanced version of the Ant-Air algorithm, which can detect the changed scenario and while keeping the lessons learnt from the previously planned safe trajectory, it then generates a safe and optimal path by avoiding collisions with the obstacles. The method presented can learn from the experience and hence improve the already generated trajectories further by using the lessons learned from the experience. The concept developed is applicable in various domains such as path planning for mobile robot, industrial robots, and simulation of part movement in narrow passages.

1 Introduction

Path planning in cluttered environment is a problem which is explored in many areas of research such as object movement in a complex virtual environment [1], [2], robot assisted safe operations [3, 4], collision detection and avoidance in machine tools [5, 6] and assembly/disassembly processes [7]. A C-space method [8, 9] is one of the important approaches used for collision detection and avoidance, especially in robotics, which is based on a 2D information grid from a C-space map. The method proposed can detect collisions, but may stuck into a local minima when non-uniform obstacles are treated for collision avoidance. A D-plan [7] is an algorithm that can generate safe path planning during assembling/disassembling processes but narrow passages are one of the problems, which is not treated by many of the existing approaches [7, 8, 9, 10].

Path planning in a cluttered area is one of the important issues in robotics. A robot manipulator may be exposed to a scattered area, where it needs to find a safe path to reach a particular pick and place solution. A mobile robot in the industrial setup for transporting products is an easy example for imagining the need of the path planning problem. Several methods exist for finding path planning solutions to avoid collisions in robotics, for example, A* approach [3], which is the extension of Edsger Dijkstra. A* method is used to determine the cost optimal path for a robot in congested area, but this method is computationally expensive; may stuck in the local minima and also is not providing any learning opportunities.

Learning capabilities of an algorithm is important in order to minimize continuously the computational efforts of the method where a quick decision is required. There exists some methods, which can learn from the collision queries [11] and also from the previously generated trajectories [12] and use this learning processes to decide quickly during the next possible scenario. The current advancement in mechatronic devices demands highly the artificial intelligence capabilities to learn from the past mistakes [13] and should be able to adjust to the dynamic-changing environment.

This paper presents the advanced version of Ant-Air algorithm [14], with new updates for considering dynamic scenarios and improved learning capabilities. The proposed method can adapt itself to the unknown-changing environment and updates its learning capabilities with a window based approach for not compromising the precious knowledge and updating itself to the dynamics of the environment.

The presented algorithm is inspired from the Ant-Air phenomenon [14]: "The phenomenon of observed change in the direction of an Ant in response to an external small air blown onto it, while it is moving straight on a flat surface." The Ant in motion, in this case, is the object agents for which actually the path planning is considered, such as: a robot, a machine part or any other mechatronic device, etc. The proposed algorithm minimizes the trajectory generation time, enhances safety of the objects in motion during motion in a cluttered environment, with advanced learning capabilities, applied to static and dynamic scenarios.

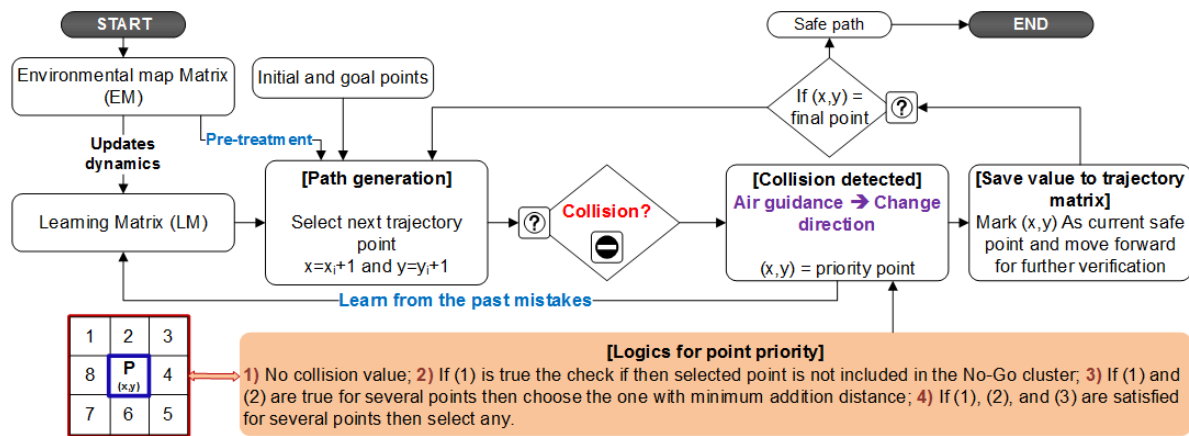[a] Corresponding author: Rafiq.Ahmad@uni.lu, engrrafiq@gmail.com.

Figure 1. Flowchart Ant-Air algorithm

2 Path Planning

The Ant-Air algorithm takes information regarding the environmental map, initial position (x_i, y_i), provisional goals (x_{pf}, y_{pf}) and a final goal (x_f, y_f) as detailed in the flowchart in Fig. 01. It then starts finding a shortest path to the next goal location by verifying adjacent points. Each coming trajectory point $P(x=x_i+1, y=y_i+1)$ is verified according to a straight-line strategy. In the case, if the next trajectory point falls on an object, a collision is detected and a diversion is programmed by selecting a secure diversion point amongst the eight adjacent points. A safe next point is selected from the eight neighbouring points with the following conditions:

1. No collision value, i.e. free space = 0; object value = 1,
2. If condition (1) is true then check if the selected point is not included in the No-Go learning matrix,
3. If conditions (1) and (2) are true for some points, then choose the one with minimum addition distance = distance from point **P** to this point + distance from this point to the goal point,
4. If conditions (1), (2), and (3) are satisfied for few points, then select any.

Once a next position is selected the onward trajectory points are verified according to the same criteria conditions discussed above unless the provisional and final goal points are reached.

An application example is shown in Fig. 2, where 4 different images are shown. Image **1**, gives the original scenario specifying obstacles, initial position A, provisional goal B, the second provisional goal C and the final goal point D. Image **2**, shows the resulted safe generated path using the Ant-Air algorithm. Image **3** gives the generated learning points that need to be stored in the Learning matrix, which are finally omitted to optimize the secure trajectory as shown in image **4**.

3 Self-learning process

The self-learning process generates a learning matrix that stores various No-Go points and its associated clusters, which need to be avoided in current and future path planning unless a dynamic scenario is concerned. Once a

safe trajectory is generated, the already developed trajectory is re-evaluated for learning to see No-Go points, which give no advantage to the safe trajectory as shown in Fig. 2, image **3**. Each trajectory point generated in image **2**, in the same figure is removed from the trajectory matrix, and trajectory is verified for collision between the previous and next position. In the case, if secure trajectory is possible, the considered point is removed and added into the "No-Go" cluster. This means that for the existing state of the trajectory, these points can be removed and can be considered no-go unless the dynamics of the environment are considered. The learning process corresponds to the specific area on the scenario map. Each learning cluster is stored in a different matrix of information data. This data is retrieved when a new trajectory needs to be generated on the same area for the moving object.

4 Treatment of Dynamic scenarios

To consider dynamic scenarios, where objects in the scene are changing, the learning process will be sacrificed in the corresponding area. In order to tackle such problem, a method is proposed to divide the environmental map into a number of map windows, i.e. 16 windows as shown in Fig 3, image **1**. The sixteen windows are considered for simplicity and the number might change according to the scenario on-hand. The map updates the learning matrix about the expected changes in any already originally defined window, e.g. shown in image **3** with the changed window shown in image **2** and **4**. Each window is checked for revisited separately, in the case of any updates; the previously learnt No-Go clustering points are forgotten for that specific window and new changes as given by the subtraction window, as shown in image **5**, are then added to the lessons learnt. These changes then corresponds to the new cluster learning values for the specific area. The new learned clusters are verified each time when a new trajectory is generated.

The innovation of the method presented lies in the *advanced learning capabilities* and *dynamic scenario consideration*. In the case of change into the environmental map, a specified window can be updated and the rest of the learning process is kept untouched.

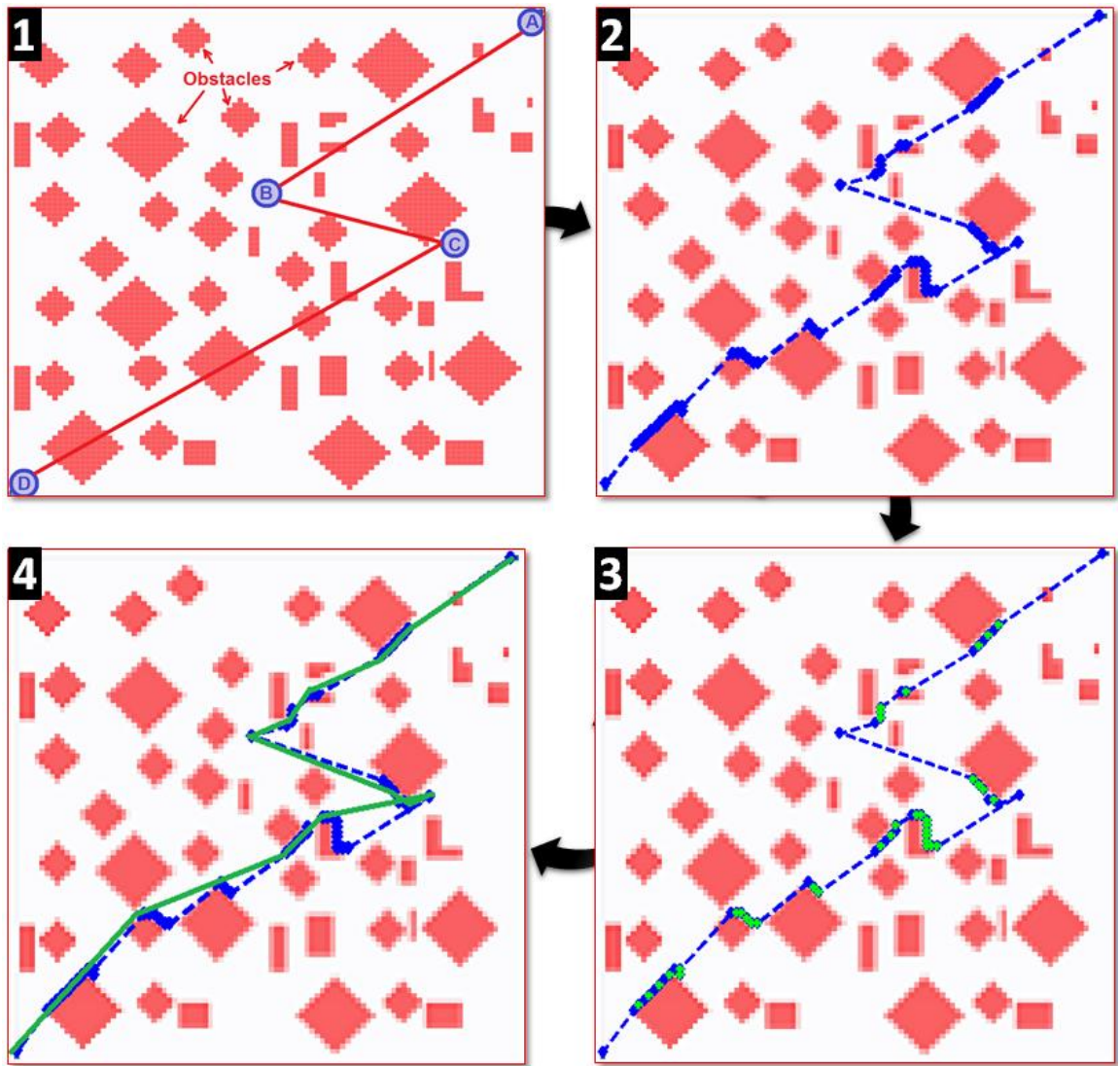a Corresponding author: Rafiq.Ahmad@uni.lu, engrrafiq@gmail.com.

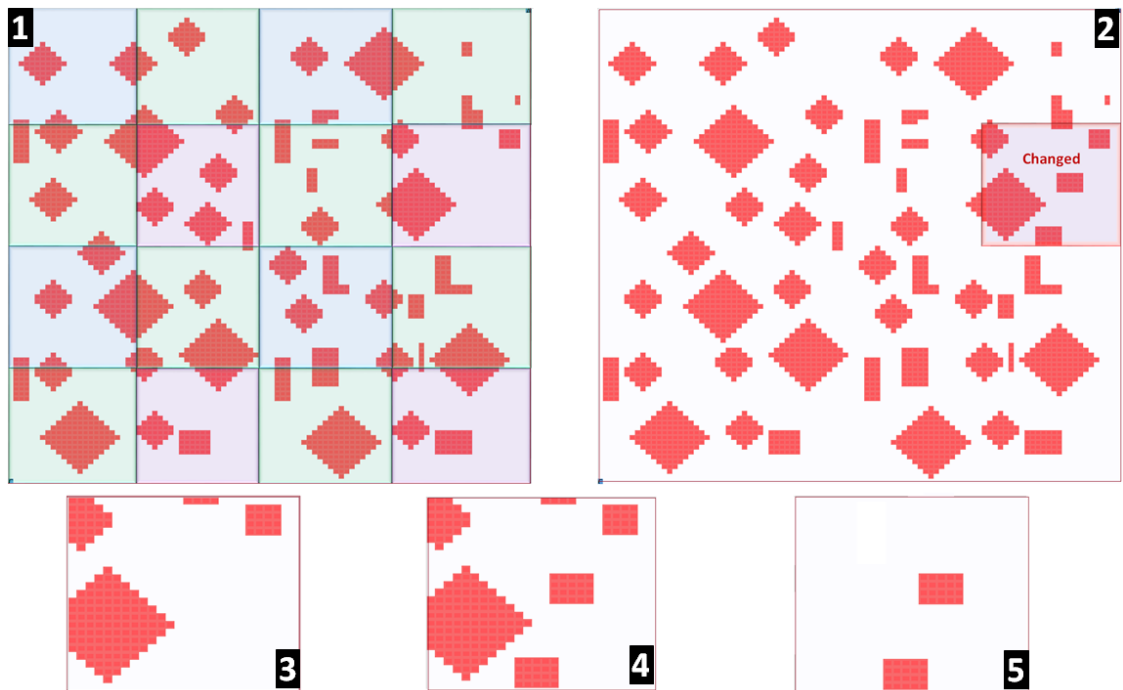**Figure 2.** Example: Application scenario with static obstacles

Figure 3. Dynamic scenario and updates

a Corresponding author: Rafiq.Ahmad@uni.lu, engrrafiq@gmail.com.

The algorithm presented in this paper is a good tool for path planning in cluttered unknown static and dynamic environment. The various applications foreseen may include the path planning for mobile robots in changing production setup, virtual part trajectory generation for simulation in a narrow spaced assembling process and industrial robot manipulators' path planning during assembly/disassembly processes. The proposed algorithm is capable of generating safe path in a millisecond for object moving in a static/dynamic environment. The perspective work will focus on the practical application of the algorithm on a swarm robot moving in a cluttered area.

5 Summary

Path planning in a dynamic changing environment while learning from the previous learnt trajectories is one of the research objectives, which needs sophisticated methods. The method proposed based on Ant-Air concept is adapted to static and dynamic scenarios for collision detection and avoidance and generating the safe path. The generated secure path is then exposed to a learning process which gives a learning from the past and finally an optimized safe path is achieved. The changing environment is considered and only the specified area of the environment will influence the learning about the region, and the learning parts for the un-changed positions will remain unchanged. The algorithm enjoys applications in many path planning scenarios such as mobile robot path planning in changing environment, industrial manipulator path generation during assembly and disassembly and part simulation during virtual assembly processes.

References

1. J.J. Dugas, M. Lee, Terrier, and J.Y. Hascoet, Virtual manufacturing for high speed milling. In 35th CIRP Intern. Seminar on Manufacturing Systems, South Korea, p. 199 – 206, (2002).
2. E. Ferre, J. P. Laumond, G. Arechavaleta, C. Esteves, Progresses in assembly path planning, in: International Conference on Product Lifecycle Management, p. 373-382, (2005).
3. M. Bennewitz, W. Burgard, A probabilistic method for planning collision-free trajectories of multiple mobile robots. Proc. of the workshop Service Robotics – Applications and Safety Issues in an Emerging Market at the 14th ECAI, 9, (2000).
4. C.W. Warren, Multiple robot path coordination using artificial potential fields. In Robotics and Automation Proceedings. 1990 IEEE International Conference, 01, p. 500 –505, (1990).
5. R. Ahmad, S. Tichadou, J.Y. Hascoet, 3D Safe and intelligent trajectory generation for multi-axis machine tools using machine vision, International Journal of Computer-Integrated manufacturing, Vol. 26 (4), p. 365-385, (2012).
6. R. Ahmad, S. Tichadou, J.Y. Hascoet, New computer vision based Snakes and Ladders algorithm for the safe trajectory of two axis CNC machines. Journal of Computer Aided Design, Doi:10.1016/j.cad.2011.12.008, Vol 44 (5), p. 355-366, (2011).
7. L. Zhang, X. Huang, Y. J. Kim, and D. Manocha, D-plan: Efficient collision-free path computation for part removal and disassembly. In Journal of Computer-Aided Design and Applications, (2008).
8. S. Jun, K. Cha, Y. S. Lee, Optimizing tool orientations for 5-axis machining by configuration-space search method. Computer-Aided Design 35 (6), p. 549 – 566, (2003).
9. Z. Liangjun, D. Manocha, An efficient retraction-based RRT planner, in: International Conference on Robotics and Automation, ICRA, p. 3743-3750, (2008).
10. J.J. Kuffner, S.M. LaValle, "RRT-connect: An efficient approach to single-query path planning, Proceedings of International Conference on Robotics and Automation, p. 995-1001, (2000).
11. J. Pan, S. Chitta, D. Manocha, Faster sample-based motion planning using instance-based learning, in: E. Frazzoli, T. Lozano-Perez, N. Roy, D. Rus (Eds.) Algorithmic Foundations of Robotics X, Springer Berlin Heidelberg, p. 381-396, (2013).
12. N. Jetchev, M. Toussaint, Fast motion planning from experience: trajectory prediction for speeding up movement generation, Auton Robot, 34, p. 111-127, (2013).
13. P. Berenson, K. Abeel, Goldberg, A robot path planning framework that learns from experience, IEEE International Conference on Robotics and Automation, p. 3671-3678, (2012).
14. R. Ahmad, P. Plapper, Ant-Air self-learning algorithm for path planning in a cluttered environment, 2nd International conference on Control Mechatronics and Automation, Dubai, UAE, (2014).

---
[a] Corresponding author: Rafiq.Ahmad@uni.lu, engrrafiq@gmail.com.