Scientific Journals
Maritime University of Szczecin

Zeszyty Naukowe
Akademia Morska w Szczecinie

2013, 36(108) z. 2 pp. 118–122
ISSN 1733-8670

2013, 36(108) z. 2 s. 118–122

Artificial intelligence in solving collision problem in restricted area

Marcin Mąka, Mariusz Dramski

Maritime University of Szczecin
70-500 Szczecin, ul. Wały Chrobrego 1–2, e-mail: {m.maka; m.dramski}@am.szczecin.pl

Key words: shortest path, safe route, restricted area, trapezoidal grid, area discretization, simplified ant algorithm, A* algorithm

Abstract
This paper presents one of the approaches to solve the collision problem in restricted area for two moving objects using artificial intelligence (SACO algorithm). Although AI should be used only when the classic methods fail, a simple comparison between them is very interesting. As we know the main task of navigation is to conduct safely an object from the point of departure to destination. This problem does not seem easy, especially if we consider the movement in restricted areas such narrow passages, ports etc.

Introduction

In [1] the use of simplified ant colony algorithm (SACO) was described. The results showed that this approach can be efficient in some situations in navigation in restricted area. Anyway, it is necessary to add that there was only the static situation considered. Of course artificial intelligence is not only ant colonies, but also other methods described in the literature, but the authors decided to choose this approach because of the specific kind of problems which can be solved by methods proposed initially by M. Dorigo [2].

In [3] the proposal of solving collision problem of two objects in restricted area was described. The authors proposed a combination of the dynamic generation of trapezoid grid and the A-star algorithm (for the optimal path search). It has been proven that using this approach it's possible to find an optimal trajectory for the object in restricted area in reasonable time.

For the purpose of comparison SACO algorithm and a method described above, the other approach is proposed. What will be the result if we connect trapezoid grid and ant colonies?

Trapezoidal grid – area discretization algorithm

Trapezoidal grid is one of the forms of data representation which allows to determine the location of each point in the plane and consider other point in given area in relation to our point. In this case the problem is reduced to determine the number of element containing considered point or it's edge or corner, which are the borders of this element. This method involves creating a grid of trapezoids and next the graph of possible connections between them. Figure 1 illustrates the way of making the connections between the points belonging to the trapezoids.

More detailed description of trapezoidal grid can be found in [3, 4, 5, 6]. As it was mentioned in [3] the dynamic situation there is a need to analyze all the objects with the ability to change their position, course, speed etc. These tasks are:

- the analysis of the location of dynamic objects in relation to the static ones;
- local modification of base grid, according to all moving objects in given moment;
- determination of the dynamic restricted areas.

All steps mentioned above are repeated periodically until our object reaches the point of destination (or there is no need to support navigator anymore).

First of all, the analysis of all static objects has to be done. The trapezoidal mesh for them is created by conducting the straight vertical lines by the points which are the beginnings / ends of the vectors, representing the edges of each object. The end of each section is the point of crossing the vertical lines with the vectors representing the edges. All the elements of grid are trapezoids or triangles (a triangle can be treated as an exceptional case of trapezoid with one base equal to zero).

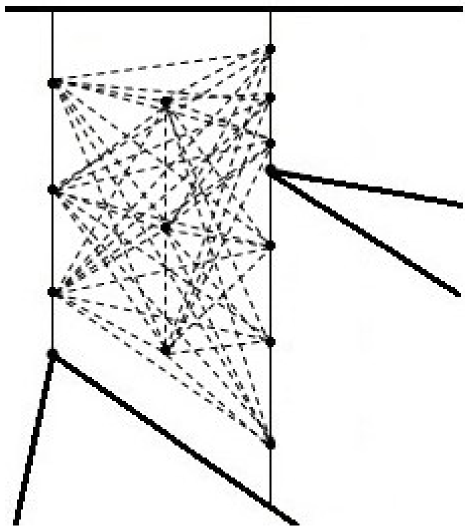

Fig. 1. Possible vertex connections

Such method of mesh generation allows to exclude a situation where in one mesh element one can find chart fragments of different properties, e.g. with various depths or fragments of an allowed area and a prohibited area.

Let's consider an example area illustrated in figure 2.

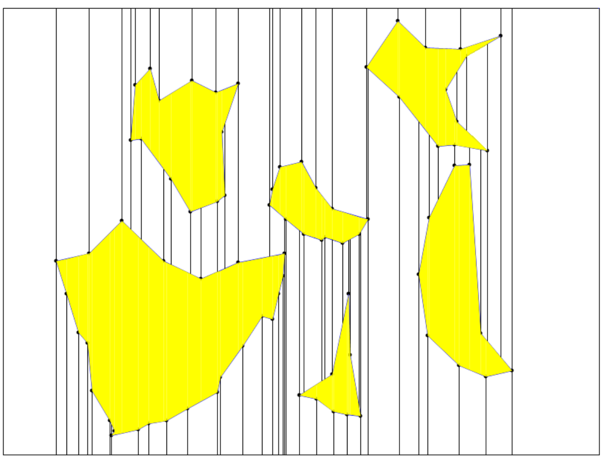

Fig. 2. The area to be analyzed

Then the trapezoidal grid was created and static restricted areas were determined (Fig. 3).

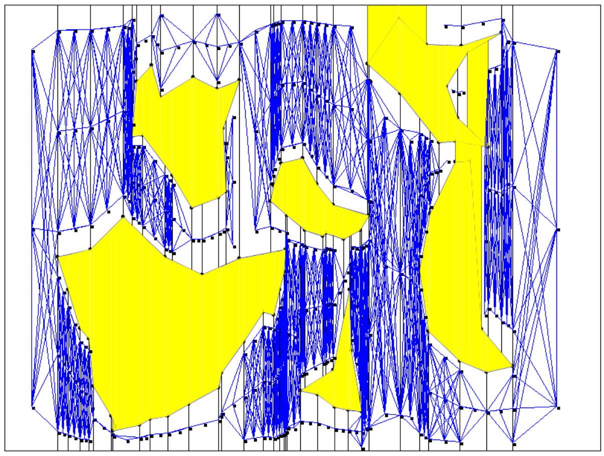

Fig. 3. The area with all possible connections

As it can be seen, this step of proposed algorithm is very simple. Now we have to take into consideration all the dynamic obstacles. It's necessary to change the grid structure in every successive time steps, depending on the current position of all moving objects in our area. After the position of moving objects is known, all the elements contained them are eliminated from the mesh.

The mesh modification process, where further objects are added, requires that at an initial stage it should be specified which of the existing mesh elements must be modified. These elements are removed and replaced by new ones, incorporating an object being added. The algorithm contains more than 20 defined basic variants of mutual position of vectors determining a newly added object and currently modified / removed mesh elements. Depending on the qualification of a situation, a mesh element is divided and replaced by new ones.

Next, because of the change of possible paths, there is a need to find an optimal path again, considering the current position of our object. There is one more thing which has to be said also. We should eliminate from the grid not only the elements containing the dynamic objects, but also these fragments of the area where there is no possibility to find a safe path for our moving object. These elements can change depending on the current situation. Examples: narrow passages, restricted areas around the objects etc.

Optimal path search

Finding an optimal path between two points it is very interesting problem in optimization. In 1959 E. Dijkstra proposed an algorithm which is now one of the most popular solutions. The main advantage of his method was the least computational complexity in comparison to previous methods. In 1968 P. Hart, N. Nilsson and B. Raphael proposed an extension of Dijkstra algorithm and called it A*. The computational complexity of this approach is only O(n) which fact makes it one of the fastest method in optimal path search problem. In [7] the comparison of some methods was done (including Dijkstra and A*).

Sometimes, the very well-known methods fail or there are new ones proposed by researchers. There is always a need to find some alternative solutions and one of them is artificial intelligence. One of the ways to solve optimization problems is ant colony optimization proposed originally by M. Dorigo.

The behavior of many ant species is based on indirect communication using pheromones. While walking from food source to the nest and back, ants leave some amount of pheromones on the ground. This way the pheromone trail is formed. Because ants can smell this trail they tend to choose paths marked by strong pheromones concentration. This fact in nature was an inspiration for researchers to build a stochastic, artificial model.

Deneubourg et al. (1990) [8] proposed a simple model that describes the dynamics of the ant colony as observed in the experiment illustrated at figure 4. In this model ψ ants per second cross the bridge in each direction at a constant speed of v cm/s, depositing one unit of pheromone on the branch. Given the lengths l_s and l_l (in cm) of the short and long branch, an ant crossing the short branch will traverse it in $t_s = l_s/v$ seconds, while the ant choosing the longer branch will use $r \cdot t_s$ where $r = l_l / l_s$. The probability $p_{ia}(t)$ that an ant arriving at decision point $i \in \{1,2\}$ selects branch $a \in \{s, l\}$ where s and l denote the short and long branch respectively can be given as the function:

$$p_{is}(t) = \frac{(t_s + \varphi_{is}(t))^\alpha}{(t_s + \varphi_{is}(t))^\alpha + (t_s + \varphi_{il}(t))^\alpha} \tag{1}$$

where $\alpha = 2$ is given experimentally [8].

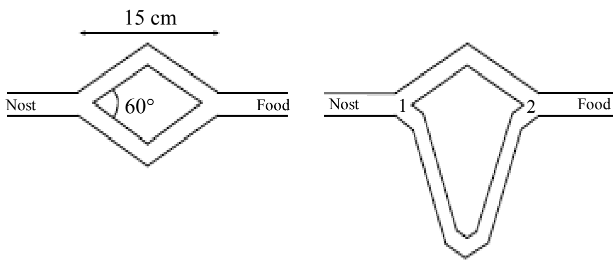

Fig. 4. The bridge experiment for illustrating ants behavior

SACO algorithm

The simplified ant colony algorithm can be described as follows.

The ants are in two modes:

– forward,
– backward.

In the "forward" mode ants moves from the nest to the source of food. While the ant reaches its target switches the mode to "backward" and comes back to the nest. In the "forward" mode ants choose randomly the path in graph $G = (N, A)$, if there exists a connection. This choice is dictated by the amount of pheromone left by other ants. Only few rules are to be fulfilled:

– pheromone update only in "backward" mode;
– elimination of the possible loops;
– the quality of the solution is dictated by the amount of pheromone trace.

Ever connection (i, j) in the graph $G = (N, A)$ can be represented as:

$$\tau_{ij} = 1,\ \forall (i, j) \in A \tag{2}$$

When the ant k is in the node i, the probability of choosing the way to node j can be expressed as:

$$p_{ij}^k = \begin{cases} \dfrac{\tau_{ij}^\alpha}{\sum_{i \in N_i^k} \tau_{ij}^\alpha} & \text{for} \quad j \in N_i^k \\ 0 & \text{for} \quad j \notin N_i^k \end{cases} \tag{3}$$

where N_i^k is the neighbourhood of ant k in the node i, $\alpha = 2$ and τ_{ij} is the artificial phermone trail which amount is updated according to the equation:

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \Delta\tau_{ij} \quad \text{where} \quad \Delta\tau_{ij} = 1 \tag{4}$$

Experiments

The area from figure 2 is given. The departure point is situated on the left side of each figure and the destination one on the right side. There are two moving objects which have the significance influence on optimal path search for our object. One object is added in the first time step. The second is added at a later step.

The static area consisted of 6 objects with a total of 94 lines. As a result of discretization, a mesh consisting of 159 elements was obtained. Once the prohibited areas were taken into account and the trapezoids included in those areas were eliminated, 103 elements remained. The total time needed for area analysis, mesh generation and identification of prohibited areas was 47 ms.

The amount of allowed mesh elements in subsequent time steps, in dynamic situation was variable and included within the limits of from 117 to 145.

The obtained mesh is a basis for an analysis of a situation in a restricted area and the obtained

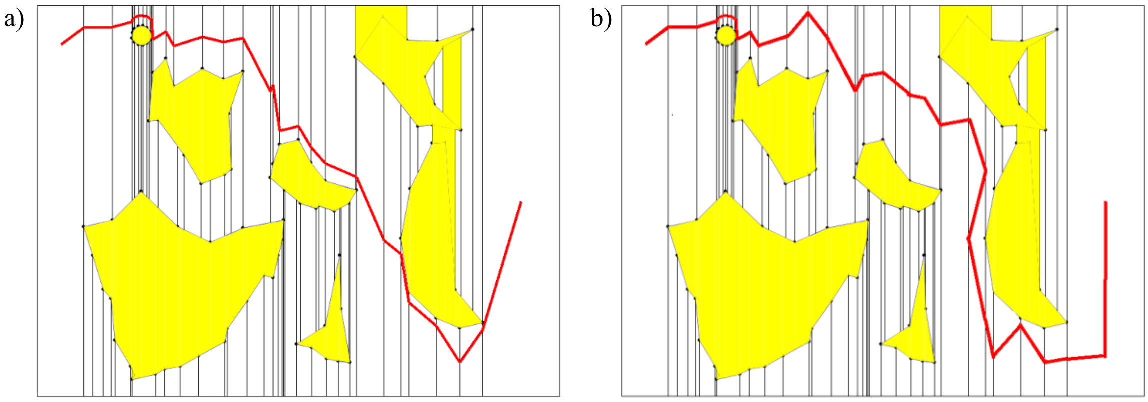

Fig. 5. Experiment results (time step 3): a) A* algorithm, b) SACO algorithm

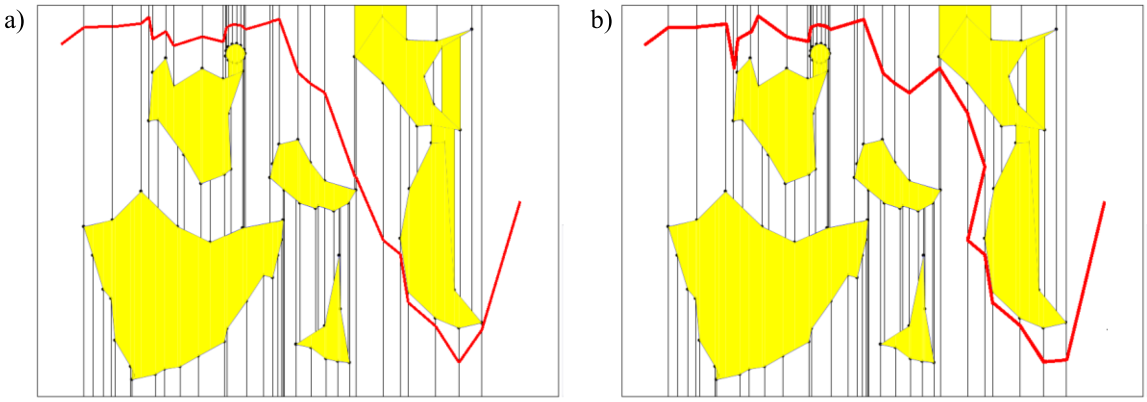

Fig. 6. Experiment results (time step 16): a) A* algorithm, b) SACO algorithm

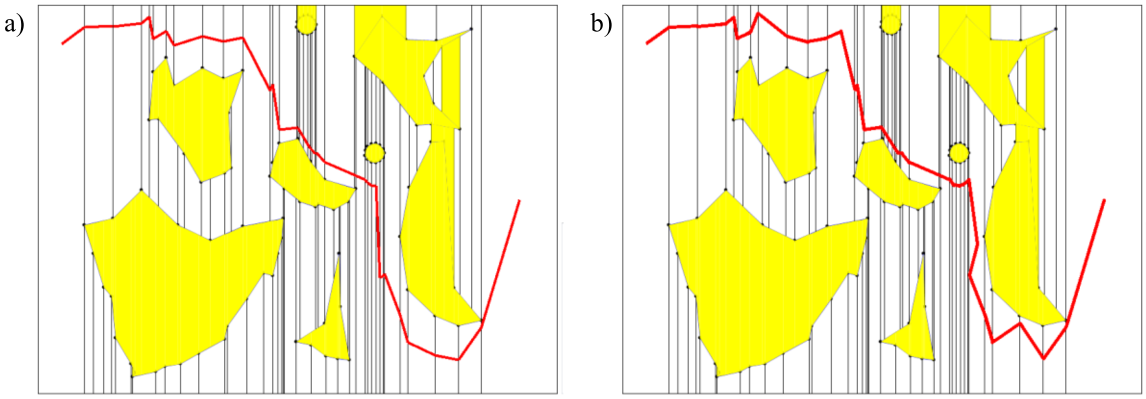

Fig. 7. Experiment results (time step 34): a) A* algorithm, b) SACO algorithm

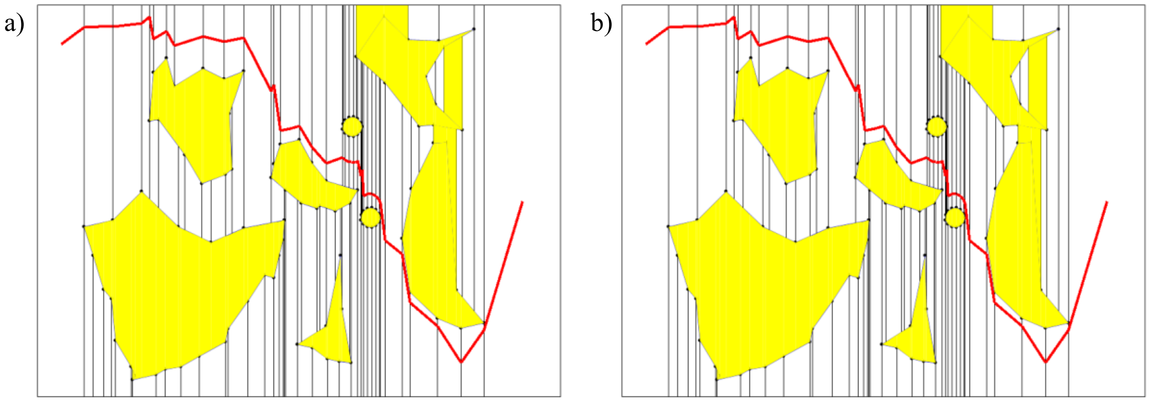

Fig. 8. Experiment results (time step 51): a) A* algorithm, b) SACO algorithm

neighborhood matrix is used for constructing a transition graph, which in turn makes up a basis for route determination and optimization.

As mentioned above, two methods of route determination were used:

– A* algorithm;
– SACO algorithm.

The results of the experiments are presented in the figures 5–8.

The time that was required to designate the trajectory was an average of 71 milliseconds for A* algorithm and 9 seconds for SACO algorithm.

As it can be seen at the figures (5a–8a) the optimal trajectory was always found. This is not an unexpected fact because the A* algorithm has the guarantee of optimality.

At the figures (5b–8b) the use of SACO algorithm is presented. In these experiments there was always possible to find a path but it can't be called the optimal. In addition, the time needed to find the trajectory was very long. This is because the SACO algorithm is not able to find the optimal path due to it's probabilistic character. Anyway, this is an interesting alternative and should be researched.

Conclusions

The aim of this paper was to present the problem of navigation in restricted area. A proposal of the methodology is described. First of all there is a need to generate a mesh. Authors proposed the use of trapezoidal mesh which leads to create the graph of all possible paths. After obtaining the graph, two methods of the shortest path search were used.

The investigations lead to the following conclusions:

– the basic advantage of trapezoidal mesh creation algorithm is short operation time and the construction of elements along to eliminate cases where fragments of two areas are located in one mesh element;
– only the path found by the A* algorithm can be called as a optimal;
– the time it takes to find a path through the SACO algorithm is very long, this algorithm can't be used to find the path in the real-time applications;
– this article treats only about the test situations, so there is a need to further research, having regard to real conditions (e.g. variable speed of objects, weather conditions etc.).

The obtained results provide a starting point for further investigation which goal is to conduct safely a ship from the point of departure to destination in restricted area.

References

1. MĄKA M., DRAMSKI M.: The choice of ship's safe route in a restricted area with the use of quadtrees for a simplified ant algorithm. XIV International Scientific and Technical Conference on Marine Traffic Engineering MTE2011, Świnoujście 2011, 319–328.
2. DORIGO M., STUTZLE T.: Ant Colony Optimization. MIT Press 2004.
3. DRAMSKI M., MĄKA M.: Algorithm of Solving Collision Problem of Two Objects in Restricted Area. Communications in Computer and Information Science 395, Springer 2013, 251-257.
4. DE BERG M., VAN KREVELD M., OVERMARS M., SCHWARZKOPF O.: Geometria obliczeniowa – algorytmy i zastosowania. Wydawnictwa Naukowo-Techniczne, Warszawa 2007.
5. DRAMSKI M., MĄKA M.: Selected shortest path in the graph algorithms with a use of trapezoidal grid in navigation in restricted area. Transport System Telematics (Archives of Transport System Telematics vol. 5, 2012, 3–7).
6. MĄKA M.: The recurrent algorithm for area discretization using the trapezoidal mesh method. Scientific Journals of Maritime University of Szczecin 29(101), 2012, 134–139.
7. DRAMSKI M.: Shortest path problem in static navigation situations. Metody Informatyki Stosowanej 5, 2011.
8. DENEBOURG J.L et al.: The self-organizing exploratory pattern of the Argentine ant. Journal of Insect Behavior 3, 159–168.