

Distributed Stochastic Search Algorithm for n -Ship Collision Avoidance

Donggyun Kim, Katsutoshi Hirayama, and Tenda Okimoto

Graduate School of Maritime Sciences, Kobe University, Japan

In this paper, to prevent collisions between ships, we introduce the Distributed Stochastic Search Algorithm (DSSA), which allows each ship to change her next-intended course in a stochastic manner immediately after receiving all of the intentions from the neighboring ships. We also suggest a new cost function that considers both safety and efficiency. We empirically show that DSSA requires much fewer messages than distributed algorithms that we previously proposed.

1. Introduction

To prevent ship collisions, several methods suggested, such as ship domain [Fujii 71, Goodwin 75, Szlapczynski 06, Szlapczynski 07, Wang 09], and fuzzy theory [Hasegawa 89, Lee 04]. Most of these methods focus on one-to-one or one-to-few ship encountering situation. Very little research had been done for many-to-many situation. Therefore, we have previously proposed two distributed algorithms to prevent ship collision. With these algorithms, each individual ship can decide its safest course on its own. The first is the Distributed Local Search Algorithm (DLSA) [Kim 14], where each ship searches for a safer course within her own local view by exchanging information with neighboring ships. The second is the Distributed Tabu Search Algorithm (DTSA) [Kim 15], which enhances DLSA with the tabu search technique [Glover 89] to escape from a Quasi-Local Minimum (QLM) in which DLSA sometimes becomes trapped. One common drawback of these algorithms is that a relatively large number of messages need to be sent in order for the ships to coordinate their actions. Since message exchange accounts for the largest part of the cost of distributed algorithms, this could be fatal, especially in cases of emergency, where quick decisions should be made.

In this paper, we introduce the Distributed Stochastic Search Algorithm (DSSA) [Zhang 02, Zhang 05], where each ship changes her next-intended course in a stochastic manner immediately after receiving all of the intentions from the neighboring ships. DSSA enables ships to exchange significantly fewer messages than DLSA and DTSA, and its stochastic nature excludes the need for a specific method to escape from QLM. Along with our development of DSSA, we also suggest a new cost function that considers both safety and efficiency in our distributed algorithms.

2. Distributed Stochastic Search Algorithm

2.1 Framework and terminology

Distributed ship collision avoidance is made up of two procedures: control and search. A framework of these procedures is given in Fig. 1. When a ship arrives at its destination, this procedure is terminated. For the control procedure, a ship decides whether to proceed to the next position. If a ship does not have any neighboring ships within a certain area, namely detection range, and also hasn't yet arrived at its destination, it moves to the next position.

For the search procedure, a ship tries to avoid collision by distributed algorithms. If a ship confirms that there is a collision risk, she runs distributed algorithms to prevent the collision. If she finds a solution, or if the computation time exceeds a certain time limit, the ship moves to the next position. All ships go through this sequence until a termination condition is met. We set a time limit on the computational time that all ships exchange messages with each other to figure out safe courses. For example, if a time limit is set to three, all ships exchange information with neighboring ships for three seconds to search safe courses. When the time has elapsed, all ships move to the next position, then they check whether a collision happened on the spot. This process is repeated every three seconds until all ships arrive at their destinations.

Figure 2 illustrates basic terminologies we used in this paper:

- **T**: the maximum length of time for which the home ship plans her future positions.
- **Detection range**: the area in which the home ship can communicate with other ships.
- **Neighbor**: a ship located within the detection range. The home ship can exchange information only with its neighbors.
- **Safety domain**: the area that the home ship prohibits a neighboring ship from penetrating.

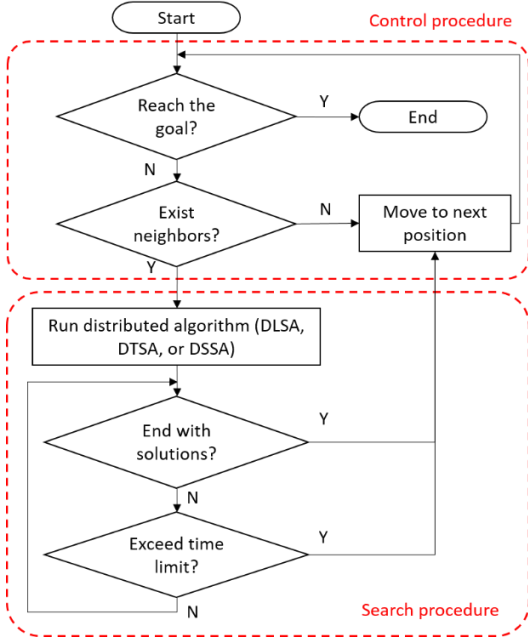


Figure 1: Framework of Distributed Ship Collision Avoidance

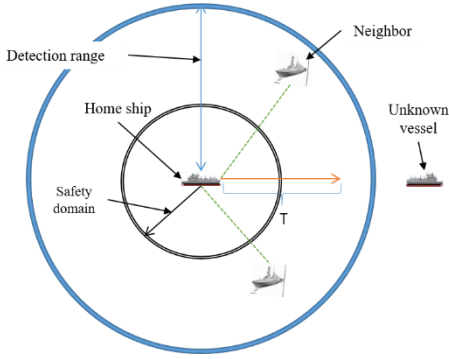


Figure 2: Description of the variables

The home ship located at the center has a detection range to detect neighboring ships. The home ship can exchange information with a neighboring ship, but not with a ship located outside the detection range. The home ship tries to keep a safety domain between itself and a neighboring ship. If that safety domain is penetrated, we consider they collide with each other.

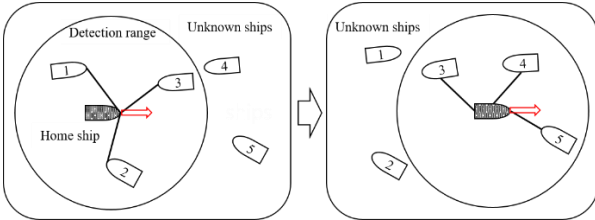


Figure 3: Progress of the movement of home ship

Figure 3 shows the progress of the movement of home ship. After finding a safe course by distributed algorithms, home ship proceeds to the next position. Then, ships 1 and 2 are going out of

the detection range, but ships 4 and 5 are entering the range to become its new neighboring ships. This process is repeated until a ship arrives at its destination.

2.2 Cost and Improvement

Equation 1 shows the Collision Risk (CR), where crs and j mean a candidate course and a neighboring ship, respectively, and $self$ means the home ship. If a neighboring ship j exists in T for crs , T is divided by $TCPA$ (Time to Closest Point of Approach). $TCPA$ is the remaining time to reach the closest point of approach for two ships. CR_{self} is only computed when collision risk exists; otherwise, 0. Equation 2 describes the cost function, which is made up of two parts: first, the sum of CR_{self} over the neighboring ships at risk for crs , and second, the relative degree between crs and a destination. The sum of CR_{self} and the relative degree is the cost for crs . The α is a weight factor. It can control the relationship between safety and efficiency. If α is bigger than one, a ship places more emphasis on safety than efficiency. Therefore, a ship can consider safety and efficiency at the same time. In this work, we set the value of α to one. The θ_{self_dest} is the relative degree between a ship's heading and a destination, as shown in Eq. 4. Due to the restriction on ship movement, we limit the maximum altering course ($MaxDegForAltCrs$).

Each candidate course has a number indicating how much cost is reduced from the currently selected next-intended course. A ship chooses the largest one as $improve_{self}$. Thus, a ship can find the safest next-intended course at that time. In DLSA and DTSa, only the ship that has larger $improve_{self}$ than those of neighboring ships can change its next-intended course to prevent an endless loop.

$$CR_{self}(crs, j) \equiv \begin{cases} \frac{T}{TCPA_{self}}, & \text{if } self \text{ will collide with ship } j \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

$$COST_{self}(crs) \equiv \alpha \sum_{j \in Neighbors} CR_{self}(crs, j) \quad (2)$$

$$+ \frac{\theta_{self_dest} - \theta_{self}(crs)}{180^\circ}$$

$$improve_{self} \equiv \max_{crs} \{ COST_{self}(CurrentlySelected) - COST_{self}(NextIntendedCourse) \} \quad (3)$$

$$\theta_{self_dest} \quad (4)$$

$$\equiv \begin{cases} \theta_{dest} - \theta_{heading}, & \text{if } |\theta_{dest} - \theta_{heading}| < |MaxDegForAltCrs| \\ \text{empty}, & \text{otherwise} \end{cases}$$

where $crs \in \{-45^\circ, \dots, -5^\circ, 0^\circ, +5^\circ, \dots, +45^\circ\} \cup \{\theta_{self_dest}\}$

$\theta_{self}(crs)$ returns $heading + crs$

2.3 Procedure for DSSA

In DSSA, the next-intended course is chosen stochastically as follows. A certain ship, which depends on rule A or B, chooses the course giving $improve_{self}$ with probability p , but does not change with probability p .

In DSSA-A, only the ships with positive $improve_{self}$ can change the next-intended courses stochastically. On the other hand, in DSSA-B, the ships with zero $improve_{self}$ can also change the next-intended courses if they have positive costs. This is because the next-intended course of a ship may produce better results at the next step, even if it does not satisfy the constraints presently. Therefore, the new next-intended course may be chosen with the probability p . Figure 4 shows the procedure for DSSA. First, a ship selects its current course as the next-intended course. After exchanging the next-intended course, then she compute $COST_{self}$ and $improve_{self}$. If a ship does not satisfy with the next-intended course, she searches new intended-course by DSSA-A or B. This process is repeated until all ships are satisfied with their current next-intended courses. In DSSA, multiple ships can change their next-intended courses simultaneously even if they are mutually neighbors. In the previous algorithms [Kim 14, Kim 15], only one ship among the one and its neighbors can do that to avoid endless loop. This might cause significant increase in the number of messages being exchanged.

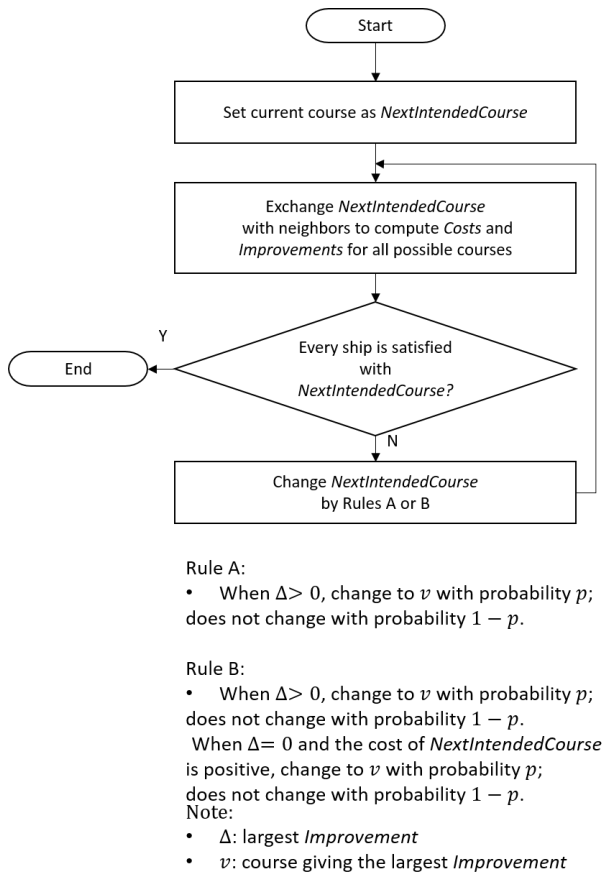


Figure 4: Procedure for DSSA

3. Experiments

Table 1 shows the experimental conditions. All ships have the same and constant speed of 12 nautical miles per hour. Each ship moves to its next position every 0.6 miles, i.e., a ship proceeds 0.6 nautical miles every three minutes. We assume that a ship cannot decrease or increase its speed to avoid collision. The detection range and safety domain are set as 12 and 0.5 nautical miles, respectively. If a target ship is located within detection range, the home ship exchanges information with it. The safety domain refers to the safe distance that must be maintained from the target ship. Probability p used in DSSA is set to 0.5 and 0.7. To evaluate DSSA, we compared its performance with that of DLSA and DTSA. We used MATLAB for the experiments.

Table 1. Variables and value for experiments

VARIABLES	VALUES
Number of ships	100
Speed	12 knots
Detection range	12 nautical miles
Safety domain	0.5 nautical miles
Probability	0.5, 0.7

For a better understanding, Figure 5 showed the result for the simulated encounters among 12 ships by DSSA-A. This demonstrates an overall situation in which ships can encounter each other without collisions. Even though the method may seem simple, it can significantly ease the burden on an officer. It shows how much a ship's decision affects the target ships. The ships in the middle of the figure altered their courses significantly while other ships altered their courses only a little. All ships changed their courses depending on the course of the other ships.

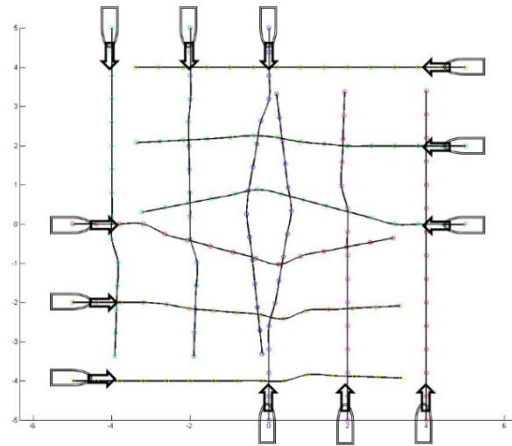


Figure 5: Simulated encounters among 12 ships by DSSA-A

We used 100 ships in this experiment. The blue and red circles mean the origin and destination for ships, respectively, as shown in Fig. 6(a). All ship's headings and positions were initialized randomly. Figure 6(b) shows the trajectories computed by DSSA-A for this problem instance. Figure 7 indicates both the average distance of trajectories and the number of messages exchanged by DLSA, DTSA and DSSA when time limit is change from one

through five in step of one. Note that no collision occurs in this experiment. The numbers in parentheses of DSSA-A and B mean the probability. In terms of average distance shown in bars in Figure 7, all algorithms showed a similar result. We can say there is little difference among these distributed algorithms. In terms of number of messages shown in lines in Figure 7, DLSA recorded the highest. DTSA had better results than DLSA. DSSA performed better well than DLSA and DTSA, irrespective of values for time limit.

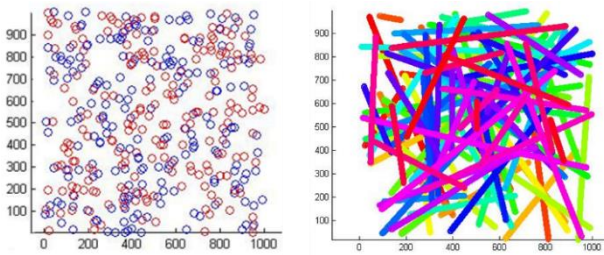


Figure 6: Initialization (a, left) and trajectories (b, right)

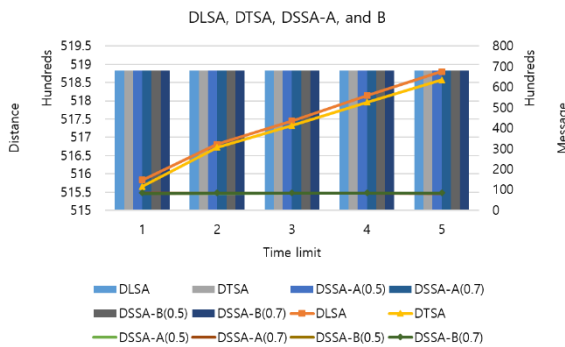


Figure 7: Simulation results

4. Conclusion

We introduced DSSA to prevent ship collision at sea. DSSA enables ships to exchange significantly fewer messages than DLSA and DTSA; furthermore, its stochastic nature excludes the need for a specific method to escape from QLM. Through developing DSSA, we also suggested a new cost function that considers both safety and efficiency in our distributed algorithms. Compared to DLSA and DTSA, DSSA produced good results, such as decreasing the number of messages. By adjusting the probability, DSSA is applicable to various situations.

For future work, we need to consider the characteristics of individual ships. The detection range and safety domain depending on the size of the ship may be sensitive to a distributed system for ship collision avoidance. It needs to be able to cope with various situations, such as when there are obstacles or when it is impossible for a ship to communicate.

References

- [Fujii 71] Fujii, Y., & Tanaka, K. *The Journal of Navigation*, 24(4): 543-552 (1971)
- [Glover 89] Glover, F. *Tabu Search-Part I. ORSA. Journal on Computing*, 1(3): 190-206 (1989)
- [Goodwin 75] Goodwin, E.M. *The Journal of Navigation*, 28(3): 328-344 (1975)
- [Hasegawa 89] Hasegawa, K., Kouzuki, A., Muramatsu, T., Komine, H., & Watabe, Y. *Journal of the Society of Naval Architecture of Japan*, 166: 445-452 (1989)
- [Kim 14] Kim, D., Hirayama, K., & Park, G. *Collision Avoidance in Multiple-ship Situations by Distributed Local Search. Journal of Advanced Computational Intelligence and Intelligent Informatics* 18(5):839-848 (2014)
- [Kim 15] Kim, D., Hirayama, K., & Okimoto, T. *Ship Collision Avoidance by Distributed Tabu Search. The International Journal on Marine Navigation and Safety of Sea Transportation*, 9(1): 23–29 (2015)
- [Lee 04] Lee, S., Kwon, K., & Joh, J. *A Fuzzy Logic for Autonomous Navigation of Marine Vehicles Satisfying COLREG Guidelines. International Journal of Control, Automation and Systems*, 2: 171–181 (2004)
- [Szlapczynski 06] Szlapczynski, R. *A Unified Measure Of Collision Risk Derived From The Concept Of A Ship Domain. The Journal of Navigation*, 59(3): 477–490 (2006)
- [Szlapczynski 07] Szlapczynski, R. *Determining the Optimal Course Alteration Manoeuvre in a Multi-Target Encounter Situation for a Given Ship domain Model. Annual of Navigation*, 12: 75–85 (2007)
- [Zhang 02] Zhang, W., Wang, G., & Wittenburg, L. *Distributed Stochastic Search for Distributed Constraint Satisfaction and Optimization: Parallelism, Phase Transitions and Performance Proc. AAAI-02 Workshop on Probabilistic Approaches in Search* 53-59. (2002)
- [Zhang 05] Zhang, W., Wang, G., Xing, Z., & Wittenburg, L. *Distributed Stochastic Search and Distributed Breakout: Properties, Comparison and Applications to Constraint Optimization Problems in Sensor Networks. Artificial Intelligence*, 161:55-87 (2005)
- [Wang 09] Wang, N., Meng, X., Xu, Q., & Wang, Z. *A Unified Analytical Framework for Ship Domains. The Journal of Navigation*, 62(4):643-655 (2009)