A distributed algorithm for the coordination of dynamic barricades composed of autonomous mobile robots

Geunho Lee [1] , Xavier Defago [1,2] , and Nak Young Chong [1]

[1] School of Information Science, Japan Advanced Institute of Science and Technology, Ishikawa, Japan

(Tel : +81-761-51-1248; E-mail: geun-lee@jaist.ac.jp)

[1,2] PRESTO, Japan Science and Technology Agency (JST)

Abstract: In this paper, we study the distributed coordination of a set of synchronous, anonymous, memoryless mobile robots that can freely move on a two-dimensional plane but are unable to communicate directly. Based on this model, we analyze the application problem that consists in having a group of robots form a barricade line to protect from car traffic a crowd of demonstrators parading on the street. For the sake of robustness, we privilege fully decentralized solutions to the problem. In particular, we give a self-stabilizing distributed algorithm to address the problem, in this presentation

Keywords: *barricade robots, decentralized control, computation model, distributed algorithm, self-stabilizing*

1. INTRODUCTION

With advances in the robotics technology related to formation control of multiple robots, formation studies have advantages in efficiency, fault-tolerance, costs per robot, and generality comparing with one high-performance robot. They are expected to apply to a variety of areas, such as mine exploration, load carriage, scout, security, and rescue. In the formation control of multiple autonomous mobile robots, various researches mainly focus on achieving the specific formation and keeping the formation while moving in the task environment ([7] ~ [11]). Balch and Arkin [10] studied formation and navigation problems in multi-robot system based on the behavior-based control paradigm. Fredslund and Mataric studied the problem of achieving global behavior in group of distributed robot [11]. Moreover, Parker [9] developed the on-line distributed control strategies in order to accomplish global tasks. There are many applications for studying this problem using a distributed control method in cooperative multiple mobile robots. Unfortunately, only few researches address it from a computational standpoint which means that much remains to be done to develop its theoretical foundation such as [1],[4], and [6]. In this paper, we study the distributed coordination of a set of synchronous, anonymous, memoryless mobile robots that can freely move on a two-dimensional plane but are unable to communicate directly.

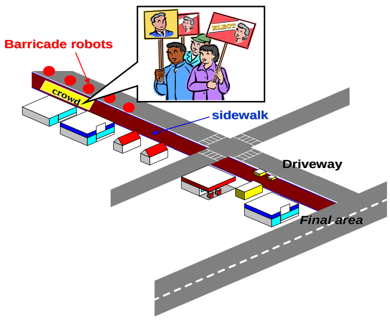


Fig. 1 Application Example

In this paper, we consider the problem of managing a barricade line: a group of robots are required to safely lead a crowd of demonstrators parading on the street in order to protect them from car traffic as shown Fig.1. For example, the barricade robots are required to form and maintain a barricade line. The barricade robots look like lively moving barricade line. Their direct task is to completely defend against dangerous articles according to the parading crowd. In particular, the robots must adapt their formation according to place and time because the crowd is changing both in length and velocity. In other words, this application is the robustly cooperative barricade problem of multiple mobile robots for moving and a variable-length target. In summary, the objective of this problem is to maintain a barricade line using multiple barricade robots while keeping pace with the variable length crowd.

We often encounter this kind of situation as real world application. For example, body guards have to protect their employer from an accident situation (e.g., flocking problem [5] – given set of n robots $\{r_1, r_2, \cdots, r_n\}$, the robots required to keep a given shape while moving). However, our main motivation for the flocking problem is a different example within the framework for a velocity matching of parading individuals as a crowd according to a variable- shaped target. The proposed application problem is to maintain the virtual defense line by a group of barricade robots against a crowd of demonstrators. Our application is similar to sheepdogs that safely conduct sheep to a more plentiful plain, under direction of a shepherd.

The main contribution of this paper is to propose a distributed algorithm by which a group of barricade robots maintain a robustly cooperative barricade line for a moving and a variable-length target. The algorithm forms a defense line with that of the crowd. Moreover, the algorithm does not require robots to memorize past actions and hence it is self-stabilizing [12]. An algorithm is said to be self-stabilizing if, starting from any arbitrary state, it always converges toward a desired behavior. In our case, this is ensured under the assumptions that no two robots have the same initial position.

Cooperative robotics still lacks the rigorous theoretical foundation developed in other fields, such as distributed computing and concurrent systems. A few researches address the problem from a computational standpoint. Defago [2] provided a survey of researches about cooperative mobile robotics to represent the next logical step beyond mobile computing, namely cooperative robotics and nanorobotics. In particular, he proposed to address the problem in a way similar to what was done in the context of concurrent and distributed system. Among recurring problems found in literature on cooperative mobile robotics, the gathering problem is the one that has been studied most extensively. Suzuki and Yamashita [3~4] proposed an algorithm to deterministically gather robots with unlimited visibility. Prencipe [6] studied similar issues for the gathering problem and proposed the CORDA model with weaker assumptions on scheduling but with the ability of detecting multiplicity. On the problem of geometric pattern formation, Suzuki and Yamashita [3] studied the formation of geometric patterns based on the proposed algorithm in the gathering problem. Based on the same model, Defago and Konagaya [1] studied a self-stabilizing algorithm for the circle formation problem. Next, Gervasi and Prencipe [5] expressed the flocking problem in a leader-followers model. Their solutions of flocking are very useful primitive for larger task such as box pushing and cooperative manipulation. Recently, Pereira [8] proposed a general framework for motion planning of cooperative mobile robots.

The remainder of the paper is organized as follows. Section 2 presents the system model of the barricade robots and the crowd. In Section 3, the definitions of the problem and the expressed notations are explained. And we decompose into two sub-problems: forming the barricade line and keeping pace with the crowd. An intuition for the algorithm and its description are given in Section 4. In Section 5, we prove the correctness of the algorithm. Finally, Section 6 concludes the paper and presents future directions.

2. SYSTEM MODELS

In a task environment, there are two kinds of system models such as autonomous mobile barricade robots and an asynchronous crowd. The crowd independently acts from the other robots. About the crowd, we assume that no person secedes from the parading group and rushes at the robots among them. If, however, finding out a vulnerable point or area where a relatively long interval arises from an unequal distance between robots, the crowd intends to go out into the driveway through the point. The crowd is modeled as a rectangular type without distinction of the body as like a landmark or a marked point. Its shape is simple rectangular form which is varied its length according to moving (i.e., worm-like motion). The length's direction of the rectangle indicates its movement direction. However, the crowd merely moves forward in this direction. Namely, there exists only one direction for its motion. The pace of the crowd is defined as movement velocity varying.

The other system model considers a collection of anonymous barricade robots evolving synchronously, without common senses such as origin, direction, and unit distance. Each robot r_i is modeled as a point with computational capabilities without computation delays. Because robots are equipped with unlimited sensors, each robot are able to observe the behaviors of all other robots and the crowd with respect to its local $x-y$ coordinate systems. The local view of a robot includes a local Cartesian coordinate system having an origin, a unit distance, and the directions of two coordinate axes with their orientation identified as the positive (+) and negative (−) sides of the axes. In other words, there is no agreement on their coordinate systems among the robots. The robots have omni-directional wheel to enable to freely move on the 2-dimensional plan and never collide. Moreover, two or more robots may not simultaneously occupy the same physical location. During the computation, each robot r_i is anonymous in the sense that they are unable to uniquely identify themselves, neither with a unique identification number nor with some external distinctive mark. All robots execute the same algorithm and thus have no way to generate a unique identity for them. Besides, there are no explicit direct means of communication; hence the only way they have to acquire information is by observing behaviors of all others robots and the crowd.

There exist barricade robots and the crowd on the 2-dimensional space simultaneously. As illustrated Fig.2, robots that are located on the plan can find itself in either one of three types of situations as follows; first situation –

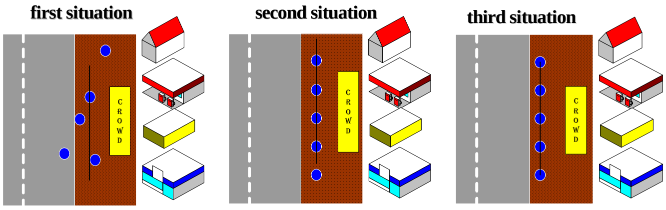
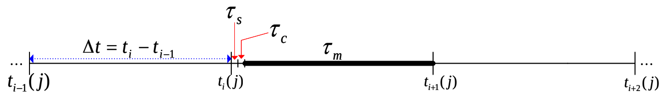


Fig. 2 Initial Configurations



τ_s := sensing time, τ_c := computation time, τ_m := motion time

Fig. 3 Synchronous Time

arbitrarily distributed, second situation – forming a barricade line without keeping pace, and third situation – forming barricade line. For the first situation, we assume that there exist barricade robots within the boundary of the maximum distance to be able to reach at less than one cycle of their activations. Moreover, barricade robots don't exist on the right side of the crowd because there are many buildings such stores. It is further assumed that initially all robots occupy a distinct position.

Time for the crowd is asynchronous and unpredictable. During which the crowd becomes active, it moves independently. Conversely, when the crowd is inactive, there is no action. Moreover, time is represented as an infinite sequence of time instants $t_0, t_1, t_2, \cdots$ for all robots. At each time instant, each robot computes a new position using a given algorithm, which takes as input the sensed positions of the crowd and the robots, and moves toward its target position.
In our proposed model, robots are synchronized because all of them become active at every time instant. The activation of robots is determined by an activation schedule that is composed as the cycle of SENSING – COMPUTATION – MOTION. (See Fig.3 – Synchronous time for barricade robots)

(1) SENSING – The robot observes the task environment by its sensor detecting the positions of other robots and the movement points of moving target. And then the robot's sensor will return the detected snapshot with respect to its local coordinate system. The result, each robots viewed as a point and the moving target as a rectangle, of the SENSING is just the pair of their coordinates and the set of movement positions.

(2) COMPUTATION – After having observed, the robot performs a local computation according to its algorithm. Here, the local computation is based on only current locations of sensed robots and the crowd. The result of the computation is a destination point.

(3) MOTION – After the robot executes the algorithm, if a target position is equal to the current location, the robot stay still. Otherwise, the robot moves toward a computed destination. The barricade robots synchronously go at computed positions for moving of the crowd with individual velocities. After the completion of the motion time (τ_m), robots go back to the SENSING state.

As shown Fig.3, the time for the robot to complete a SENSING-COMPUTATION-MOTION cycle is neither infinite nor extremely small. The elapsed time in the SENSING and the COMPUTATION is negligible compared to the time required in the MOTION. The life of a robot consists in repeating an endless cycle of states (1) – (3).

Based on our proposed model, the new algorithm consists of a function φ that is executed by a robot r_i at every time instant. The arguments of φ consist of the current position for the robot and a set of positions for all robots at the corresponding time instant. All positions are represented in terms of the local coordinate system of r_i. The returned value by φ is the new position and the velocity for r_i which must be within one distance unit of the previous position, as measured by r_i's coordinate system. For simplicity, it is assumed that obtaining information about the system model, computing the new position and velocity, and moving toward it are instantaneous. Note that arguments to φ include only current information and thus the algorithm can use no knowledge of the past. The robots are oblivious because they are unable to remember any past actions or sensing. As a result of [3], it can be discussed that the algorithm defined φ is self-stabilizing. However, this is not necessarily true, as this largely depends on the exact definition of self-stabilization.

3. PROBLEM DEFINITIONS

In this section, we formally define collective problems of the barricade line. Based on this model, we analyze the application problem that consists in having a group of robots form a safe defense line to protect from car traffic a crowd of demonstrators parading on the street. For the sake of robustness, we privilege fully decentralized solutions to the problem. Thus, in this presentation, we give a self-stabilizing distributed algorithm to address the problem.

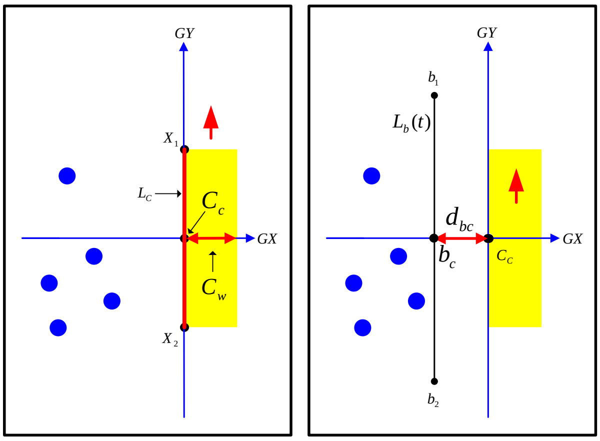


Fig. 4 Definitions of Crowd & Barricade Line

3.1 Definitions

Given a robot r_i, $p_i(t)$ denotes its position according to global $x-y$ coordination systems at time t, $p_i(0)$ is its initial position at time t_0. A configuration $P(t)=\{p_i(t)|1\le i\le n\}$ at time t denotes a set of distinct points for a group of n robots $r_1,\cdots,r_n$ with distinct positions and arbitrarily located. Configurations are used to model the positions of barricade robots and to represent target points that organize a desired formation. A formation $L_b(t)=\{p_i(t+1)|1\le i\le n\}$ is a target configuration with a uniform interval. We call the formation the barricade line of the group of barricade robots. We suppose that the barricade line is expressed in a coordinate system having the crowd as origin and oriented according to the movement direction of the crowd.

When observed a crowd with the rectangular body as like Fig.4, we formalize the crowd line (L_C) that the length's direction of the rectangular body indicates the movement direction. In detail, the line is measured from one vertex to the other vertex along the longest line adjacent to a driveway as illustrated Fig.4. The center point (C_C) of the crowd is the midpoint of the L_C. We assume that the C_C is expressed as common origin in the global coordinate system. In addition, the global coordination is agreed on as follows; First, GY (Y-axis of common coordination) passed through the C_C, is parallel for the L_C, and indicates that (+) positive direction of the common Y-axis is equal to the movement direction of the crowd. Second, GX (X-axis of common coordination) is defined as passing through the C_C and vertical direction for the GY, whose (+) positive direction turns a counterclockwise rotation. As seen the Fig.4, the C_W (width of the crowd) is perpendicular to the L_C in the rectangular body. More specially, this C_W is defined as unit distance of the common coordinate system. Using C_C, $GY-GX$ coordination, and C_W, all robots may reach an agreement on common origin, common direction, and the shared unit of measure because they are able to know when they take a snapshot of the positions with respect to their own local coordination.

Where does the barricade line (L_b) prefer to be located in order to maintain the barricade line for the parading crowd? In detail, how long distance from the crowd is favorably located in the barricade line? As shown Fig.4, the L_b is located in the distance (d_{bc}) which is equals to C_W, runs parallel with GY, and vertically passes through a point b_c on the GX. To begin with, we define that C_W and the distance d_{bc} between C_C and the point b_c are the same distance. At the point b_c, the length of the barricade line is bisected. Next, let the stable length of the barricade line for maximum parading distance per one activation schedule be $f_b(t)=f_C(t)+\Delta X_1^{\max}+\Delta X_2^{\max}=f_C(t)+2C_W$ where let f_b and f_C be the length of a barricade line and the length of crowd line, respectively. In addition, C_W is equal to $\Delta X_1^{\max}$ or $\Delta X_2^{\max}$ ($\because$ $C_W=\Delta X_1^{\max}=\Delta X_2^{\max}$). Here, we additionally define a condition for $f_C(t)$ (the length of the crowd line) if $\sum d_u(t)\le f_C(t)$ where there are two assumptions related to the $f_C(t)$; first, however much the barricade line may reduce, the length of the crowd cannot become a point. Second, the barricade line is limited to finite length. Moreover, the endpoints of the barricade line indicate the point b_1 located in (+) direction of GY and the point b_2 in (−) direction of GY, respectively. More specially, let $d_u(t)$ (the uniform distance) at time t be the distance between p_i and p_{i-1}, when the barricade robots are individually positioned on the barricade line. In order to defend the crowd that moves on the street through a vulnerable area, the robots must maintain the barricade line with a uniform interval $d_u(t)$.

3.2 Conditions

When the crowd parades on the street, two endpoints for movement of the crowd are modeled as X_1, X_2 of L_C. Let $v_{X_1}^{\max}$ and $v_{X_2}^{\max}$ be the maximum velocities for X_1, X_2, and let v_b be the velocity of a barricade robot. Let's image the worst possible case related to X_1, X_2 - The crowd is moving away from the barricade robots at the same time. In other words, the crowd must not be too fast otherwise; the barricade robots will lag behind it and will not be able to maintain the barricade line. In order to be solved for the problem, the following velocity condition must be required:

$$\max(v_{X_1}^{\max}, v_{X2}^{\max}) < \min_j v_b \ . \tag{1}$$

Next, let ΔX_1 and ΔX_2, which is moving distance per one activation time, be the displacement for X_1, X_2, and let $d_{\min}$ be the minimum traveling distance for a robot. In order to maintain the barricade line, the elapsed time of a MOTION state in the robot's activation schedule must not be too long. Otherwise, the crowd could travel away from barricade robots for the time instant between two consecutive SENSING states. In order to protect from the previous mentioned case, the following condition must be met:

$$\max(\Delta X_1^{\max}, \Delta X_2^{\max}) < C_W < \min_j d_{\min} \ . \tag{2}$$

As the definition of Fig.3 in the Section 2, τ_s, τ_c, and τ_m are sensing, computation, and motion time at each time instant, respectively. We assumed that the elapsed time of the SENSING and the COMPUTATION is negligible compared to the time required in the MOTION. The above condition is intuitive because we consider the duration of the MOTION at same time instant of all the robots. In detail, let Δt be the time interval between t_i and t_{i-1}. Using this condition, we formally define the time interval of synchronous time as follows:

$$\tau_s,\ \tau_c << \tau_m \cong t_i - t_{i-1} = \Delta t \ . \tag{3}$$

For example, the velocity for the moving endpoint X_1 is resolved into the displacement ΔX_1 and the synchronous time instant.

$$v_{X_1} = \frac{\Delta X_1}{\tau_m} \cong \frac{\Delta X_1}{\Delta t} = \frac{\Delta X_1}{t_i - t_{i-1}} \ . \tag{4}$$

Analogously, the minimum movement distance of any barricade robot may be analyzed into

$$d_{\min} = \min_j v_b \times \tau_m = \min_j v_b \times (t_i - t_{i-1}) = \min_j v_b \times \Delta t \ . \tag{5}$$

3.3 Problem Definitions

The problem addressed in this paper is the maintenance of the barricade line by a group of mobile robots. We define the barricade problem as follows:

PROBLEM 1 (BARRICADE PROBLEM)

*Let $r_1, \cdots r_n$ be a group of barricade robots with distinct and arbitrarily distributed positions, let a crowd be a group of a parading demonstrators modeled as variable-length rectangular shape with movement velocities varying according to each time instant, and let $L_b(t)$ be a barricade line given in input to $r_1, \cdots r_n$. The barricade robots are able to find out a solution for the **BARRICADE PROBLEM** if, starting at time instant t_0, $\exists t_1 \ge t_0$ such that, $\forall t \ge t_1$ the robots completely maintain the barricade line against the crowd.*

In the reminder of this paper, we decompose Problem 1 into two subproblems. The first subproblem (Problem 2) is a weaker version of Problem 1, wherein a barricade line is formed without keeping pace with a crowd. The second subproblem (Problem 3) consists in transforming a configuration, in which robots are uniformly arranged on a barricade line, into one where the robots are arranged uniformly according to a crowd of each length-varied shape with each time-varying movement velocities.

PROBLEM 2 (BARRICADE FORMATION PROBLEM)

*Let $r_1, \cdots r_n$ be a group of barricade robots with distinct and arbitrarily distributed positions, let a crowd be a group of a parading demonstrators modeled as variable-length rectangular shape according to each time instant, and let $L_b(t)$ be a barricade line given in input to $r_1, \cdots r_n$. The barricade robots are able to find out a solution for the **BARRICADE FORMATION PROBLEM** if, starting at time instant t_0, $\exists t_1 \ge t_0$ such that, $\forall t \ge t_1$ the robots completely form the barricade line against the crowd.*

PROBLEM 3 (VELOCITY MATCHING PROBLEM)

*Let $r_1, \cdots r_n$ be a group of barricade robots of distinct positions with a uniform interval on a barricade line, let the crowd be a group of a parading demonstrators modeled as variable-length rectangular shape with movement velocities varying according to each time instant, and let $L_b(t)$ be a barricade line given in input to $r_1, \cdots r_n$. The barricade robots are able to find out a solution for the **VELOCITY MATCHING PROBLEM** if, starting at time instant t_0, $\exists t_1 \ge t_0$ such that, $\forall t \ge t_1$ the robots keep pace with the crowd.*

4. ALGORITHM DESCRIPTIONS

Given an initial configuration where a collection of robots are arbitrarily distributed on the 2-dimensional plan, our proposed algorithm ensures that the system model migrates toward a configuration of robots that is a valid solution to the BARRICADE PROBLEM. This algorithm actually builds upon functions under hypothesis that the barricade robots (1) are oblivious in the sense that they are unable to recall past behaviors and observations, (2) share no common sense such as a common coordination and a unit distance, (3) are anonymous in the point of view that cannot be distinguished from each others, and (4) have no direction communication only through observing positions of other robots and movement positions of the crowd. Besides, the proposed algorithm is self-stabilizing because of starting from any arbitrary state always converges towards the barricade line. It takes an arbitrary configuration in which all robots have distinct positions and regardless of their activation, eventually brings the system toward a configuration in which all robots are uniformly distributed on the barricade line. Informally, the algorithm relies on the fact that the task environment observed by all robots in the same while they have their different own local coordination system. Moreover, the barricade line is unique and depends only on the relative velocities and the length of the crowd. Therefore, the algorithm makes sure that the barricade line remains unchanged and uses it as a configuration at each time instant.

The intuition behind the algorithm is simple and we briefly describe it here (See Fig.5 and Algorithm 1.). Whenever all robots become active at each time instant, they consider their current positions as a configuration by new updated common $GY-GX$ coordinate systems. (We explained agreement on common sense with respect to robots' own local coordination in Section 3.) And then robots decide their rankings as sorting in increasing order by each position value with respect to

GY -axis and GX -axis, respectively. In particular, after the sorting, it is guaranteed that

$$\forall p_i, p_j,\ \exists p_i = (x_i, y_i), p_j = (x_j, y_j)$$
$$\text{if } p_i < p_j \Leftrightarrow \{(y_i < y_j) \vee \{(y_i = y_j) \wedge (x_i < x_j)\}\} \quad (6)$$

where p_i and p_j are points of r_i and r_j in a configuration. The ranking m of the configuration it belongs to is computed, (i.e., the position that the current configuration occupies in the sorting.), by $\text{Sort}(p_i, \cdot)$. Next, the m -th robots consider the crowd as their target, and compute a virtual barricade line according to states, which represent f_C, C_C, and C_W, of the crowd retrieved in the current SENSING state. Then, the robots know they are uniformly distributed to the end point b_2 by an interval of d_u on the barricade line originated from a start point b_1. At this point, the positions returned by this execution are the next configuration, which indicates positions on the barricade line defined by $L_b(t)$, that all robots will try to reach. After having computed the $L_b(t)$, a robot find itself in either one of three of situations. First, the simplest situation occurs when a current configuration is exactly equal to $L_b(t)$. In this case, the robots will stay still because the crowd doesn't move. The second situation arises when robots exist on a $L_b(t)$ without keeping pace. Even though the conditions $d_{bc} = C_W$ and $k = dist(p_m, p_{m-1})$ (Here, the distance between p_m and p_{m-1} is considered as one or two direct neighbor(s) of a robot) is satisfied, the midpoints of $P(t)$ and $L_b(t)$ doesn't coincide. In this case, a robot estimates displacements of the crowd for an activation schedule as illustrated Algorithm 1 - ($\Delta X_1 = dist_{GY}(b_1, p_1)$: = the displacement of X_1 = distance between endpoint b_1 of $L_b(t)$ and a position of the highest ranking robot on the barricade line at each activation time). Therefore, all robots made to move toward the target points on the barricade line at the velocity v_m. The third situation arises when robots are arbitrarily distributed. In this case, robots must form a barricade line before everything else. The robots go to target positions with maximum velocity $v_b^{\max}$.

5. ALGORITHM CORRECTNESS

LEMMA 1. *Two or more robots do not simultaneously occupy the same location.*

PROOF. If starting at time instant t_0, $\exists t_1 \geq t_0$ such that, $\forall t \geq t_1$, Two or more robots do not simultaneously occupy the same location. The proof is by induction. By assumption on the initial configurations that no two robots occupy the same location initially (Section 2), it is true that two or more robots have the distinct positions at t_0. Algorithm 1 gives all configurations where robots are located can be decided by ranking at time instant t. Therefore, if two or more robots have the distinct positions at some time instant, then they always have distinct positions afterward. □

LEMMA 2. *An arbitrarily located robot r_k moves toward a point that is located on a barricade line.*

PROOF. Robots moves only at lines 19, 22, 24 of Algorithm 1. Let us consider each case for an arbitrary positioned robot.

1. **At line 19**, r_k stays still, so it obvious does not move away from L_b
2. **At line 22**, r_k moves from the interior of the barricade line toward a point located on a barricade line. Therefore, r_k actually exists on the only barricade line.
3. **At line 24**, r_k moves to a point in its barricade line according to the decided ranking. Since a point always belongs to its barricade line according to the initial assumption, the new position must be within the boundary of the maximum distance to be able to reach at less than one cycle of activation.

The robot r_k is unable to move away from the boundary of a barricade line in any of the three cases. □

LEMMA 3. *All robots located on a barricade line always remain on the barricade line.*

PROOF. If arbitrary robot r_j on the barricade line at time t is selected, r_j exists on a barricade line for $\forall t' \geq t$. When a configuration including an arbitrary robot r_j that exists only on a barricade line at time t, there are two cases to be considered.

1. **At line 18,** This configuration is stationary and not removable from a barricade line because the crowd does not move. Namely, all robots must exactly keep the current configuration at time t'. No robot moves away from the barricade line.
2. **At line 20,** In this case, an arbitrary robot r_j does not satisfy the condition $P(t) = L_b(t)$ at time t. However, by the definition of the barricade line, the configuration included in an arbitrary robot r_j indicates a type of a barricade line slow than the crowd. So, all robots of this configuration move from a point located in the current line toward a new point in barricade line. No robot move away from the barricade line. □

LEMMA 4. *If the crowd moves at any velocity, then the barricade robots keep pace with the crowd.*

PROOF. Let a velocity of the crowd be v_{X_j}. The proof is by induction, where the induction step directly follows from the mentioned condition $v_{X_j} \leq v_{X_j}^{\max} < \min_i v_b$ and the definitions of each velocity for the crowd and the robots. Indeed, according to Lemma 2 and Lemma 3, no crowd is faster than the barricade robots. □

LEMMA 5. *If the crowd does not move, eventually the barricade line remains unchanged.*

PROOF. This follows trivially from lemma 2 and lemma 3. □

6. CONCLUSION

In this paper, we proposed the BARRICADE PROBLEM as the first step toward the application of recurrent problems found in the literature on the cooperative robotics. Our algorithm ensured that a group of mobile barricade robots will eventually form a safe defense line moving together with the crowd, and adapt its moving velocity to that crowd. We showed that the algorithm forms a barricade line and maintains its pace. In addition, our algorithm is decentralized in the sense that each robot need only know the position of neighboring robots and the crowd, but no explicit inter-robot communication or centralized control is required.

We indeed intend to use this problem as a starting point for studying the role and strengths of several different communication models. For instance, the algorithm presented in the Section 4 relies exclusively on the fact that robots can detect each others position, as is the case with GPS or RFID. It is now interesting to see whether replacing GPS or RFID with other communication models (e.g., ad hoc networking with directional antennas) still allows for solving the barricade line problem. However, this question is not addressed here and left for later studies.

ACKNOWLEDGMENTS

This research is conducted as a program for the "Fostering Talent in Emergent Research Fields" in Special Coordination Funds for Promoting Science and Technology by Ministry of Education, Culture, Sports, Science and Technology. The authors would like to thank Rami Yared for his valuable comments as well as his encouragements.

REFERENCES

[1] X. Defago and A. Konagaya, “Circle formation for oblivious anonymous mobile robots with no common sense of orientation”, Proc. 2nd ACM Intl. Workshop on Principles of Mobile Computing (POMC’02), pp.97-104, 2002.

[2] X. Defago, “Distributed computing on the move: From mobile computing to cooperative robotics and nanorobotics (a position paper)”, Proc. 1st ACM Annual Workshop on Principles of Mobile Computing (POMC’01), pp. 49-55, 2001.

[3] I. Suzuki and M. Yamashita, “Distributed anonymous mobile robot: Formation of geometric patterns”, SIAM Journal of Computing 28(4), 1347-1363, 1999.

[4] I. Suzuki and M. Yamashita, “Agreement on a common x-y coordination system by a group of mobile robots”, Proc. Dagstuhl Seminar on Modeling and Planning for Sensor-Based Intelligent Robots, 1996.

[5] V. Gervasi and G. Prencipe, “Coordination without Communication: The Case of The Flocking Problem”, Discrete Applied Mathematics, Vol 143, pp. 203-223, 2004.

[6] G. Prencipe, “CORDA: Distributed Coordination of a Set of Autonomous Mobile Robots”, Proc. Fourth European Research Seminar on Advances in Distributed Systems (ERSADS 2001), pp. 185-190, 2001.

[7] G. A. S. Pereira, V. Kumar, and M. F. M. Campos, "Decentralized Algorithms for Multirobot Manipulation via Caging," Proceedings of the Fifth International Workshop on Algorithmic Foundations of Robotics (WAFR'02), pp. 242-258, 2002.

[8] X. Yun, G. Alptekin, and O. Albayrak, “Line and Circle Formation of Distributed Physical Mobile Robots”, Journal of Robotic Systems 14(2), pp. 63-76, 1997.

[9] Lynne E. Parker, “Distributed Algorithms for Multi-Robot Observation of Multiple Moving Targets”, Autonomous Robots, 12 (3), pp. 231-255, 2002.

[10] Tucker Balch and Ronald C. Arkin, “Behavior-based Formation Control for Multi-robot Teams”, IEEE Transactions on Robotics and Automation, 14(6), pp. 926-939, 1998.

[11] Jakob Fredslund and Maja J Mataric, "A General, Local Algorithm for Robot Formations",IEEE Transactions on Robotics and Automation, special issue on Advances in Multi-Robot Systems, 18(5), pp. 837-846, Oct 2002.

[12] S. Dolev, “Self-Stabilization”, MIT Press, 2000

.

Algorithm 1 Combining the parts – BARRICADE PROBLEM

Input: $p_1, \cdots p_n$ are a group of positions for $r_1, \cdots, r_n$ retrieved in the current SENSING state. At the same time, f_C, C_C, and C_W represent the length of the crowd line, the center point of the crowd line, and the width for the crowd, respectively.

Function $\varphi_{barricade}(P, p_i)$

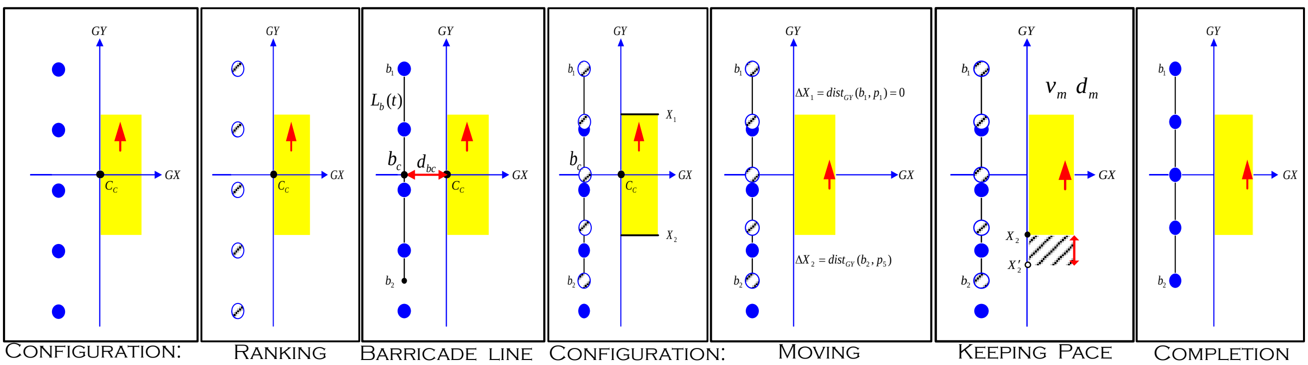
Agreement_Common(f_C) ; // *common sense*
For All $i = 1,2,\cdots,n$ **Do**
  Sort(p_i, GY) ; // *selecting candidate with greatest* GY *-coord.*
  If exactly one candidate of each ranking exists **Then**
    m := ranking. // $m = 1,\cdots,n$
  Else {Several candidates exist}
    Sort(p_i, GY) ; // *Selecting candidate with greatest* GY *-coord.,*
    Sort(p_i, GX) ; // *and then* GX *-coord.*
    m := ranking.
  End If
End For
Barricade_Line(d_{bc}, b_c, f_b) ; // *favorable location, center point, and length of* $L_b(t)$
b_1, b_2 :=Barricade_Line(d_{bc}, b_c, f_b) ; // *endpoints of barricade line*
d_u :=Barricade_Line(d_{bc}, b_c, f_b) ; // *uniform interval* : Eq.(1)
$L_b(t)$:=Next positions; // i.e., *next configuration* $L_b(t) \Rightarrow P(t+1) = \{p_1, \cdots, p_n\}$
    // *target positions at a distance from point* b_1 : Eq.(2)
    // m *-th set of target point* v_m : Eq.(3)
If The crowd doesn’t move **Then** $\{P(t) = L_b(t)\}$
  No Action. // i.e., *robot* r_i *stay still*
Else If robots exist on the L_b without keeping pace **Then**
    $\Delta X_1 = dist_{GY}(b_1, p_1)$ and $\Delta X_2 = dist_{GY}(b_2, p_n)$; // *displacements*
    Go to d_m with v_m . // *velocity of* r_m : : Eq.(4)
    Else {Robots are arbitrarily distributed}
      Go to target position.
    End If
End If

//: Eq.(1) $d_u = \frac{1}{n-1} \times f_b$

//: Eq.(2) $b_{1GY} - \left(\frac{m-1}{n-1} \times f_b \right)$

//: Eq.(3) $\left(b_{1GX}, b_{1GY} - \left(\frac{m-1}{n-1} \times f_b \right) \right)$

//: Eq.(4) $v_m = v_{X_1} \times \frac{n-m}{n-1} + v_{X_2} \times \frac{m-1}{n-1}$



CONFIGURATION: $P(t)$ | RANKING | BARRICADE LINE | CONFIGURATION: $P(t+1) = L(t)$ | MOVING VELOCITY | KEEPING PACE | COMPLETION

Fig. 5 Example of How the Algorithms Work