Compatibility of Convergence Algorithms for Autonomous Mobile Robots*

Yuichi Asahiro[a,*], Masafumi Yamashita[b]

[a] *Department of Information Science, Kyushu Sangyo University, 2-3-1 Matsukadai, Higashi-ku, 813-8503, Fukuoka, Japan*
[b] *Kyushu University*

Abstract

We investigate a swarm of anonymous oblivious mobile robots under the semi-synchronous ($\mathcal{SSYNC}$) scheduler. Each robot has a function called *target function* to decide the destination from the robots' positions, and operates in Look-Compute-Move cycles, i.e., identifies the robots' positions, computes the destination by the target function, and then moves there. Robots may have different target functions. Let Φ and Π be a set of target functions and a problem, respectively. If the robots whose target functions are chosen from Φ always solve Π, we say that Φ is compatible with respect to Π. If Φ is compatible with respect to Π, every target function $\phi \in \Phi$ is an algorithm for Π (in the conventional sense). Note that even if both ϕ and ϕ' are algorithms for Π, $\{\phi, \phi'\}$ may not be compatible with respect to Π.

From the view point of compatibility, we investigate the convergence, the fault tolerant (n, f)-convergence (FC(f)), the fault tolerant (n, f)-convergence to f points (FC(f)-PO), the fault tolerant (n, f)-convergence to a convex f-gon (FC(f)-CP), and the gathering problems, assuming crash failures. As a result, we see that these problems are classified into three groups: The convergence, the FC(1), the FC(1)-PO, and the FC(f)-CP compose the first group: **Every** set of target functions which always shrink the convex hull of a configuration is compatible. The second group is composed of the gathering and the FC(f)-PO for $f \geq 2$: **No** set of target functions which always shrink the convex hull of a configuration is compatible. The third group, the FC(f) for $f \geq 2$, is placed in between. Thus, the FC(1) and the FC(2), the FC(1)-PO and the FC(2)-PO, and the FC(2) and the FC(2)-PO are respectively in different groups, despite that the FC(1) and the FC(1)-PO are in the first group.

Keywords: autonomous mobile robot, compatibility, convergence, crash fault, gathering

---

*An extended abstract of this paper appeared in the Proc. International Colloquium on Structural Information and Communication Complexity (SIROCCO2023) [4].

*Corresponding author

1. Introduction

1.1. Convergence problem and compatibility

Over the last three decades, swarms of autonomous mobile robots, e.g., automated guided vehicles and drones, have obtained much attention in a variety of contexts [1, 2, 6, 10, 11, 22, 24, 26, 27, 28, 29, 31, 32, 34]. Among them is understanding solvable problems by a swarm consisting of many simple and identical robots in a distributed manner, which has been constantly attracting researchers in distributed computing society e.g., [1, 2, 3, 5, 7, 10, 13, 14, 15, 17, 18, 19, 20, 21, 22, 23, 24, 25, 30, 32, 33, 35, 36].

Many of the works mentioned above adopt the following robot model. The robots look identical and indistinguishable. Each robot is represented by a point that moves in the Euclidean plane. It lacks identifier and communication devices, and operates in Look-Compute-Move cycles. When a robot starts a cycle, it identifies the multiset of the robots' positions in its x-y local coordinate system (i.e., it has full visibility and the strong multiplicity detection capability), computes the destination point using a target function[1] based only on the multiset identified, and then moves towards the destination point. Here, the x-y local coordinate system is right-handed (i.e., it has the chirality) and its origin is always the position of the robot (i.e., it is self-centric), and all robots are typically requested to take the same target function.

If each cycle starts at a time t and finishes, reaching the destination, before (not including) $t+1$, for some integer t, the scheduler is said to be *semi-synchronous* ($\mathcal{SSYNC}$). If cycles can start and end any time (even on the way to the destination), it is *asynchronous* ($\mathcal{ASYNC}$).

This paper investigates several *convergence problems*, e.g., [2, 13, 14, 15, 18, 20, 21, 25, 32]. The simplest convergence problem requires the robots to converge to a single point. Under the $\mathcal{SSYNC}$ model, the problem is solvable for robots with unlimited visibility [32], and is also solvable for robots with limited visibility [2].

Under the $\mathcal{ASYNC}$ model, for robots with unlimited visibility, it is solvable by a target function called CoG, which always outputs the center of gravity of the robots' positions [13]. Finally, [25] gives a convergence algorithm for robots with limited visibility under the 1-bounded $\mathcal{ASYNC}$ model: from the moment one robot performs a Look-phase to the moment it finishes its Move-phase, at most one other robot performs a Look-phase. Unlike CoG, the algorithm does not assume the multiplicity detection capability.

The authors of [13] showed that CoG correctly works under the sudden-stop model, under which the movement of a robot towards the center of gravity might stop on the way after traversing at least some fixed distance. This implies that

[1] Roughly, a target function is a function that maps any multiset of points in the Euclidean plane to a point in the Euclidean plane. Later, we define a target function a bit more carefully.

the robots can correctly converge to a point, even if they take different target functions, as long as they always move robots towards the current center of gravity over distance at least some fixed constant. This idea is extended in [15]. The authors introduced the δ-*inner* property of target functions: Let P, D, and $\boldsymbol{o}$ be the multiset of robots' positions, the axes aligned minimum box containing P, and its center, respectively. Define $\delta * D = \{(1-2\delta)\boldsymbol{x} + 2\delta\boldsymbol{o} : \boldsymbol{x} \in D\}$. A function ϕ is said to be δ-inner, if $\phi(P) \in \delta * D$ for any P. Then, they showed that the robot system converges to a point, if all robots take δ-inner target functions, provided $\delta \in (0, 1/2]$.

Consider a problem Π and a set of target functions Φ. If the robots whose target functions are chosen from Φ always solve Π, we say that Φ is *compatible with respect to* Π. For example, every (non-empty) set of target functions satisfying the δ-*inner* property is compatible with respect to the convergence problem under the $\mathcal{ASYNC}$ model [15].

If a singleton $\{\phi\}$ is compatible with respect to Π, we abuse to say that target function ϕ is an *algorithm*[2] for Π. If a set Φ of target functions is compatible with respect to Π, every target function $\phi \in \Phi$ is an algorithm for Π by definition. (The converse is not always true.) Thus there is an algorithm for Π, if and only if there is a compatible set Φ with respect to Π. We sometimes say that a problem Π is *solvable*, if there is a compatible set Φ with respect to Π, which means that there is an algorithm for Π.

We would like to find a large compatible set Φ with respect to Π. That Π has a large compatible set with respect to Π implies that Π has many algorithms. The difficulty of problems might be compared in terms of the sizes of their compatible sets. A problem Π which has a large compatible set Φ seems to have some practical merits, as well. Two swarms both of which are controlled by target functions in Φ (which may be produced by different makers) can merge to form a larger swarm, keeping the correctness of solving Π. When a robot breaks down, we can safely replace it with another robot, as long as it has a target function from Φ.

1.2. Convergence problems in the presence of crash faults

This paper investigates three fault-tolerant convergence problems, besides the convergence and the gathering problems. This paper considers only crash faults: A faulty robot can stop functioning at any time, becoming permanently inactive. A faulty robot may not cause a malfunction, forever. We cannot distinguish such a robot from non-faulty ones. Let n and $f(\leq n-1)$ be the number of robots and the number of faulty robots.

The *fault-tolerant (n,f)-convergence problem* (FC(f)) is the problem to find an algorithm which ensures that, as long as at most f robots are faulty, *all*

[2] Here, we abuse term "algorithm," since an algorithm must have a finite description. A target function may not. (See Section 1.3 for the definition of a target function.) To compensate the abuse, when to show the existence of a target function, we will give a finite procedure to compute it. To show its non-existence, we will show the non-existence of a function (not only an algorithm).

non-faulty robots converge to a point.

The *fault-tolerant (n,f)-convergence problem to f points* (FC(f)-PO) is the problem to find an algorithm which ensures that, as long as at most f robots are faulty, *all robots* (including faulty ones) converge to at most f points. All non-faulty robots need not converge to the same point. If f faulty robots have crashed at different positions, each non-faulty robot must converge to one of the faulty robots.

The *fault-tolerant (n,f)-convergence problem to a convex f-gon* (FC(f)-CP) is the problem to find an algorithm which ensures that, as long as at most f robots are faulty, the convex hull of the positions of *all robots* (including faulty ones) converges to a convex h-gon CH for some $h \leq f$, in such a way that, for each vertex of CH, there is a robot that converges to the vertex.

Since an algorithm for the FC(1)-PO solves the FC(1), the former is not easier than the latter. (Note that for $f \geq 2$, an algorithm for the FC(f)-PO may not solve the FC(f).) Since an algorithm for the FC(f)-PO solves the FC(f)-CP, again the former is not easier than the latter. In [13], the authors showed that, for all $f \leq n-2$, CoG is an algorithm for the FC(f) under the $\mathcal{ASYNC}$ model. To the best of the authors' knowledge, the FC(f)-PO and the FC(f)-CP have not been investigated so far.

Gathering problems

The *gathering problem* is similar to the convergence problem. It requires the robots to gather in the exactly the same location. The gathering problem has been investigated under a variety of assumptions, e.g., [1, 7, 12, 17, 18, 20, 21, 32, 35, 36]. Under the $\mathcal{SSYNC}$ model, the gathering problem is not solvable if $n = 2$. If $n > 2$, it is solvable, provided that all robots initially occupy distinct positions [32]. Under the $\mathcal{ASYNC}$ model, the same results hold [12].

Many other works investigate the gathering problem in the presence of crash faults. See surveys [18, 20] for information on the fault-tolerant gathering problems. The *fault-tolerant (n, f)-gathering problem*, which is sometimes called the *weak gathering problem*, is the problem to find an algorithm which ensures, as long as at most f robots are faulty, all non-faulty robots gather at a point. In [1], the authors proposed a fault-tolerant $(n, 1)$-gathering algorithm, assuming that $n \geq 3$ and the robots initially occupy distinct positions. Provided the chirality, the fault-tolerant (n, f)-gathering problem can be solved for any $f < n$ except for the bivalent configuration. Here, a configuration is said to be bivalent, if it consists of two points with multiplicity $n/2$ [7].

Since the gathering problem is substantially harder than the convergence problem, it may not be a good idea to use a gathering algorithm to solve the convergence problem, especially when the size of solvable instances by a convergence algorithm is a major concern.

The gathering and convergence problems in the presence of Byzantine faults have also been investigated or surveyed, e.g., in [1, 8, 9, 19, 20].

1.3. Our contributions

Let R be the set of real numbers. By $\mathcal{P}$, we denote the set of all (non-empty) **multisets** P of points in R^2 such that $(0,0) \in P$. Then, a *target function* ϕ is a function from $\mathcal{P}$ to R^2.

Suppose that a robot r of a swarm of n robots identifies a multiset P of n points, which are the positions of the n robots in its x-y local coordinate system Z, in Look phase. Then, $(0,0) \in P$, and hence $P \in \mathcal{P}$, since Z is self-centric.[3] Using its target function ϕ, r computes the target point $\boldsymbol{x} = \phi(P)$, in Compute phase, and then it moves to $\boldsymbol{x}$ in Z, in Move phase.

Let the convex hull and the center of gravity of P be $CH(P)$ and $g(P)$, respectively. For any $0 \leq d$, let $d * CH(P) = \{d\boldsymbol{x} + (1-d)g(P) : \boldsymbol{x} \in CH(P)\}$. The *scale* $\alpha(\phi)$ of a target function ϕ is defined by

$$\alpha(\phi) = \sup_{P \in \mathcal{P}} \alpha(\phi, P),$$

where $\alpha(\phi, P)$ is the smallest d satisfying $\phi(P) \in d * (CH(P))$. Then, the scale of a set Φ of target functions ϕ is defined by

$$\alpha(\Phi) = \sup_{\phi \in \Phi} \alpha(\phi).$$

The only target function ϕ satisfying $\alpha(\phi) = 0$ is CoG. Thus the set Φ of target functions satisfying $\alpha(\Phi) = 0$ is a singleton {CoG}. The idea of scale is similar to that of δ-inner property in [15], and more directly embodies the idea behind the δ-inner target function, in the sense that ϕ never expands and tends to shrink $CH(P)$ if $\alpha(\phi) < 1$.

Our contributions can be summarized in Table 1. In Table 1, FC(f), FC(f)-PO, and FC(f)-CP are respectively abbreviations of the fault-tolerant (n,f)-convergence problem, the fault-tolerant (n,f)-convergence problem to f points, and the fault-tolerant (n,f)-convergence problem to a convex f-gon. Letter 'A' in the entry of a problem Π and a range of $\alpha(\Phi)$ means that every Φ such that $\alpha(\Phi)$ is in the range is compatible with respect to Π. Letter 'N' means that any Φ such that $\alpha(\Phi)$ is in the range is not compatible with respect to Π, which indicates the absence of an algorithm. Letter 'E' means some Φ is compatible, while some other is not, which indicates the existence of an algorithm. Letter '?' means that the answer is unknown.

For example, the entry of Convergence and $\alpha(\phi) = 0$ is A. Thus {CoG} is compatible with respect to the convergence problem, which is equivalent to say that CoG is an algorithm for the convergence problem, as [13] shows. Not only the case of $\alpha(\phi) = 0$, but also the case of $0 < \alpha(\Phi) < 1$, every Φ is compatible with respect to the convergence problem.

Table 1 shows that the problems we discuss in this paper can be classified into three groups, Group1, Group2, and Group3, from the view point of compatibility. The convergence, the FC(1), the FC(1)-PO, and the FC(f)-CP compose

[3] That $(0,0) \notin P$ means an error of eye sensor, which we assume will not occur, in this paper.

Table 1: The compatibility of a set Φ of target functions with respect to a problem Π, taking its scale $\alpha(\Phi)$ as a parameter. Each entry contains the status A, E, N, or ? of the compatibility of Φ with respect to Π (and the theorem/corollary/observation/citation number establishing the result in parentheses). Letter 'A' means that every Φ such that $\alpha(\Phi)$ is in the range is compatible with respect to Π. Letter 'N' means that any Φ such that $\alpha(\Phi)$ is in the range is not compatible with respect to Π, which indicates the absence of an algorithm. Letter 'E' means that some Φ is compatible, while some other is not, which indicates the existence of an algorithm. Letter '?' means that the answer is unknown.

problem Π	scale $\alpha(\Phi)$		
	$\alpha(\Phi) = 0$	$0 < \alpha(\Phi) < 1$	$\alpha(\Phi) = 1$
Convergence	A (Thm. 1 [13])	A (Thm. 2)	E (Thm. 3)
FC(1)	A ([13])	A (Cor. 2)	E (Thm. 5)
FC(1)-PO	A (Thm. 4)	A (Thm. 4)	E (Thm. 5)
FC(f)-CP ($f \geq 2$)	A (Thm. 8)	A (Thm. 8)	E (Thm. 9)
FC(f) ($f \geq 2$)	A (Thm. 6 [13])	E (Thm. 7)	E (Cor. 4)
FC(2)-PO	N (Thm. 10)	N (Thm. 10)	E (Obs. 2, Thm. 11)
FC(f)-PO ($f \geq 3$)	N (Thm. 10)	N (Thm. 10)	?
Gathering	N (Thm. 13)	N (Thm. 13)	E (Thm. 12 [32])

Group1: **Every** set of target functions which always shrink the convex hull of a configuration is compatible. For each of the problems in Group1, in Table 1, the entries for scales $\alpha(\Phi) = 0$, $0 < \alpha(\Phi) < 1$, and $\alpha(\Phi) = 1$ are A, A, E, respectively. That is, triple (A,A,E) characterizes Group1. The second group Group2 is composed of the gathering and the FC(f)-PO for $f \geq 2$: **No** set of target functions which always shrink the convex hull of a configuration is compatible. Then, triple (N,N,E/?) characterizes it. The third group Group3, the FC(f) for $f \geq 2$, is placed in between, and is characterized by triple (A,E,E). Thus, the FC(1) and the FC(2), the FC(1)-PO and the FC(2)-PO, and the FC(2) and the FC(2)-PO are respectively in different groups, despite that the FC(1) and the FC(1)-PO are in the first group. Figure 1 illustrates how the problems are related.

Organization.

After introducing the robot model we adopt in this paper in Section 2, we investigate the convergence problem, which has been studied extensively, in Section 3, from the new viewpoint of understanding its compatibility. In Section 4, we discuss the compatibilities of two convergence problems in the presence of at most one crash fault, i.e., the FC(1) and the FC(1)-PO, and show that they have the same property as the convergence problem. Sections 5 and 6 respectively investigate the compatibilities of the FC(f) and the FC(f)-CP for $f \geq 2$. Every set Φ of target functions such that $0 \leq \alpha(\Phi) < 1$ is compatible with respect to the FC(f)-CP (like the FC(1), the FC(1)-PO, and the convergence problem), while this property does not hold for the FC(f) for $f \geq 2$. Section 7 first shows that a target function ϕ is an algorithm for the FC(f)-PO for $f \geq 2$, only if $\alpha(\phi) \geq 1$. Thus any set Φ of target functions

Figure 1: Relationship among problems and their classification into three groups Group1, Group2, and Group3, which are characterized by triples (A,A,E), (N,N,E/?), and (A,E,E), respectively. Relation "$X \to Y$" between two problems X and Y represents that any algorithm for X also solves Y, i.e., Y is not harder than X.

such that $0 \leq \alpha(\Phi) < 1$ is **not** compatible with respect to the FC(f)-PO for $f \geq 2$, unlike the FC(f). We then present an algorithm $\psi_{(n,2)}$ for the FC(2)-PO. Section 8 investigates the gathering problem to show the difference between this and the convergence problems from the viewpoint of compatibility. We conclude the paper by presenting a list of open problems, in Section 9.

2. The model

Consider a robot system $\mathcal{R}$ consisting of n robots $r_1, r_2, \ldots, r_n$. Each robot r_i has its own unit of length, and a local compass defining an x-y local coordinate system Z_i, which is assumed to be right-handed and self-centric, i.e., its origin $(0,0)$ is always the position of r_i. We also assume that r_i has the strong multiplicity detection capability, i.e., it can count the number of robots resides at a point.

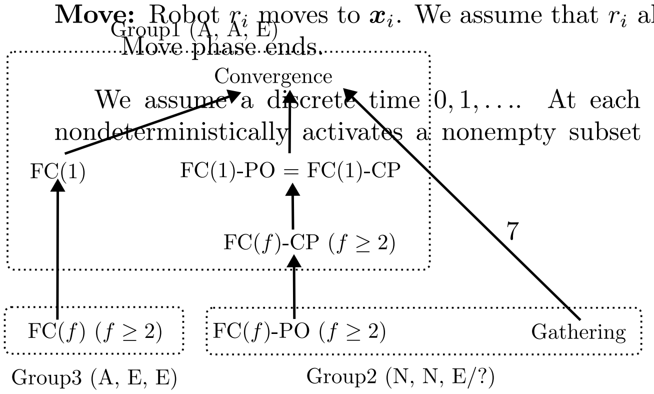
Given a target function ϕ_i, each robot $r_i \in \mathcal{R}$ repeatedly executes a Look-Compute-Move cycle:

Look: Robot r_i identifies the multiset P of the robots' positions (including the one of r_i) in Z_i. Since r_i has the strong multiplicity detection capability, it can identify P not only distinct positions of P.

Compute: Robot r_i computes $\boldsymbol{x}_i = \phi_i(P)$. (We do not mind even if ϕ_i is not computable. We simply assume that $\phi_i(P)$ is given by an oracle.)

Move: Robot r_i moves to $\boldsymbol{x}_i$. We assume that r_i always reaches $\boldsymbol{x}_i$ before this Move phase ends.

We assume a discrete time $0, 1, \ldots$. At each time $t \geq 0$, the scheduler nondeterministically activates a nonempty subset of robots. Then activated



robots execute a cycle which starts at t and ends before (not including) $t+1$, i.e., the scheduler is semi-synchronous ($\mathcal{SSYNC}$).

Let Z_0 be the x-y global coordinate system, which is right-handed and is not accessible by any robot r_i. The coordinate transformation from Z_i to Z_0 is denoted by γ_i. We use Z_0 and γ_i just for the purpose of explanation.

The position of robot r_i at time t in Z_0 is denoted by $\boldsymbol{x}_t(r_i)$. Then $P_t = \{\boldsymbol{x}_t(r_i) : 1 \le i \le n\}$ is a multiset representing the positions of all robots at time t, and is called the *configuration* at t.

Given an initial configuration P_0, an assignment $\mathcal{A}$ of a target function ϕ_i to each robot r_i, and an $\mathcal{SSYNC}$ schedule (produced by the $\mathcal{SSYNC}$ scheduler), which decides the set of robots activated (to start a new Look-Compute-Move cycle) at each time instant t, the execution of $\mathcal{R}$ is a sequence $\mathcal{E} : P_0, P_1, \dots, P_t, \dots$ of configurations starting from P_0. Here, for all r_i and $t \ge 0$, if r_i is not activated at t, $\boldsymbol{x}_{t+1}(r_i) = \boldsymbol{x}_t(r_i)$. Otherwise, if it is activated, r_i identifies $Q_t^{(i)} = \gamma_i^{-1}(P_t)$ in Z_i, computes $\boldsymbol{y} = \phi_i(Q_t^{(i)})$, and moves to $\boldsymbol{y}$ in Z_i. Then $\boldsymbol{x}_{t+1}(r_i) = \gamma_i(\boldsymbol{y})$. We assume that the scheduler is fair: It activates every robot infinitely many times. Throughout the paper, we regard the scheduler as an adversary.

Before closing this section, we introduce several notations which we will use in the following sections. Let $P \in \mathcal{P}$. The distinct points of P is denoted by $\overline{P}$. Then $|P|$ (resp. $|\overline{P}|$) denotes the number of points (resp. the number of distinct points) in P.

Let $CH(P)$ be the convex hull of P. While P is a multiset of n points, $CH(P)$ is a convex region (including its inside). We sometimes denote $CH(P)$ by a sequence of vertices of $CH(P)$ appearing on the boundary counter-clockwise. Obviously, $CH(P) = CH(\overline{P})$.

The center of gravity $g(P)$ of P is defined by $g(P) = \sum_{\boldsymbol{x} \in P} \boldsymbol{x}/n$. Note that $g(P) \ne g(\overline{P})$ in general.

For two points $\boldsymbol{x}$ and $\boldsymbol{y}$ in R^2, $dist(\boldsymbol{x}, \boldsymbol{y})$ denotes the Euclidean distance between $\boldsymbol{x}$ and $\boldsymbol{y}$. For a set $B (\subseteq R^2)$ of points and a point $\boldsymbol{a} \in R^2$, $dist(\boldsymbol{a}, B) = \min_{\boldsymbol{x} \in B} dist(\boldsymbol{a}, \boldsymbol{x})$.

In what follows, when we discuss a problem for a fixed size n of the robot swarm, the domain of a target function is the set $\mathcal{P}_n$ of all (non-empty) multisets P of n points such that $(0,0) \in P$. However, if it is obvious from the context, we omit n from $\mathcal{P}_n$ to write $\mathcal{P}$.

3. Convergence problem

We start our investigation with the convergence problem, provided that all robots are non-faulty. For any $0 \le \alpha \le 1$, consider a target function CoG_α defined by

$$\mathrm{CoG}_\alpha(P) = (1-\alpha) g(P),$$

for any $P \in \mathcal{P}$. The scale of CoG_α is α, and $\mathrm{CoG}_0 = \mathrm{CoG}$. The following theorem holds, since CoG works correctly under the sudden stop model.

Theorem 1 ([13]). *For any* $0 \leq \alpha < 1$*, let* $\Phi_\alpha = \{\mathrm{CoG}_\alpha\}$*. Then* Φ_α *is compatible with respect to the convergence problem, or equivalently,* CoG_α *is an algorithm for the convergence problem.*

We extend Theorem 1 to have the following theorem. Let $N_\epsilon(B)$ be the ϵ-neighbor of a set B, i.e., $N_\epsilon(B) = \{\boldsymbol{x} : dist(\boldsymbol{x}, \boldsymbol{b}) < \epsilon, \boldsymbol{b} \in B\}$. When B is a singleton $\{\boldsymbol{b}\}$, we denote $N_\epsilon(B)$ by $N_\epsilon(\boldsymbol{b})$.

Theorem 2. *Let* Φ *be a set of target functions such that* $0 \leq \alpha(\Phi) < 1$*. Then* Φ *is compatible with respect to the convergence problem.*

Proof. Let $\phi_i \in \Phi$ be the target function taken by robot r_i for $i = 1, 2, \ldots, n$. Let $\alpha(\phi_i) = \alpha_i$ and $\alpha = \max_{1 \leq i \leq n} \alpha_i$. Then $\alpha \leq \alpha(\Phi) < 1$.

Consider any execution $\mathcal{E} : P_0, P_1, \ldots$ starting from any initial configuration P_0. We show that P_t converges[4] to a point.

Suppose that $P_t = \{\boldsymbol{x}, \boldsymbol{x}, \ldots, \boldsymbol{x}\}$ at some time t, i.e., $|\overline{P_t}| = 1$. Since $\boldsymbol{g}_t = g(P_t) = \boldsymbol{x}$, $P_{t+1} = P_t$. Thus convergence has already been achieved. We assume without loss of generality that $|\overline{P_t}| \geq 2$ for all $t \geq 0$.

Let $A_t \subseteq \mathcal{R}$ be the set of robots activated at time t. If $\boldsymbol{x}_t(r) = \boldsymbol{g}_t$ for all $r \in A_t$, $P_{t+1} = P_t$ holds. However, there is a robot r such that $\boldsymbol{x}_t(r) \neq \boldsymbol{g}_t$ since $|\overline{P_t}| \geq 2$, and r is eventually activated by the fairness of scheduler. Thus, without loss of generality, we assume that there is a robot $r \in A_t$ such that $\boldsymbol{x}_t(r) \neq \boldsymbol{g}_t$, and that $P_{t+1} \neq P_t$ holds for all $t \geq 0$.[5]

We denote $CH(P_t)$ by CH_t. Since $\alpha < 1$, $CH_{t+1} \subseteq CH_t$, [6] which implies that CH_t converges from outside to a convex k-gon CH (including a point and a line segment) for some positive integer k. We show that CH is a point, i.e., $k = 1$.

Let $\boldsymbol{p}_0, \boldsymbol{p}_1, \ldots, \boldsymbol{p}_{k-1}$ be the vertices of CH aligned counter-clockwise on the boundary. To derive a contradiction, we assume that $k \geq 2$. For any pair (i, j) $(0 \leq i < j \leq k-1)$, let $L_{(i,j)} = dist(\boldsymbol{p}_i, \boldsymbol{p}_j)$, and $L = \min_{0 \leq i < j \leq k-1} L_{(i,j)}$. Since CH_t converges to CH from outside, for any $0 < \epsilon \ll (1-\alpha)L/n$,[7] there is a time instant t_0 such that, for all $t \geq t_0$, $CH \subseteq CH_t \subseteq N_\epsilon(CH)$. Observe

[4] Formally, you should read as "a sequence $\{P_t : t = 0, 1, \ldots\}$ converges." This convention applies in what follows. Observe that if P_t converges to a point, then each robot converges to the point. Later, when we discuss the convergence to multiple points, the convergence of P_t is not sufficient to show the convergence of each robot.

[5] Formally, let $\mathcal{E}'$ be a sequence constructed from $\mathcal{E}$ by removing all subsequences $P_{t+1}, P_{t+2}, \ldots, P_{t'}$ such that $P_{t-1} \neq P_t$, $P_t = P_{t+1} = \ldots = P_{t'}$, and $P_{t'} \neq P_{t'+1}$, where we assume that $P_{-1} \neq P_0$. Then $\mathcal{E}'$ is also an execution, i.e., there is an $\mathcal{SSYNC}$ schedule that produces $\mathcal{E}'$, and $\mathcal{E}$ converges if and only if $\mathcal{E}'$ does. We consider $\mathcal{E}'$ instead of $\mathcal{E}$. This is what this and similar assumptions in this paper formally mean.

[6] Note that $CH_{t+1} = CH_t$ happens, when every robot activated at t is located strictly inside CH_t.

[7] In the rest of proof, all arguments below hold, e.g., for any ϵ such that $0 < \epsilon < (1 - \alpha)L/((3-\alpha)n)$. In what follows, like in this inequality, we use notation "$\ll$ (much less than)" or "$\gg$ (much greater than)" if deriving a bound is obvious and is not our concern.

that for any vertex $\boldsymbol{p}$ of CH,

$$dist(\boldsymbol{p}, \alpha * CH_t) > (1-\alpha)\left(\frac{L}{n} - \epsilon\right) - \epsilon \gg \epsilon,$$

because $dist(\boldsymbol{p}, \boldsymbol{g}_t) > L/n - \epsilon$.

Let r be any robot, and suppose that r is activated at some time $t \geq t_0$ (by the fairness of the scheduler). Then $\boldsymbol{x}_{t+1}(r) \in \alpha * CH_t$, which implies that $\boldsymbol{x}_{t+1}(r) \notin N_\epsilon(\boldsymbol{p})$, for any vertex $\boldsymbol{p}$ of CH. If r is reactivated at some time $t' > t$ for the first time after t (by the fairness of the scheduler), since $CH_{t'} \subseteq CH_t$ and $\boldsymbol{x}_{t'+1}(r) \in \alpha * CH_{t'}$, $\boldsymbol{x}_{t'+1}(r) \notin N_\epsilon(\boldsymbol{p})$, for any vertex $\boldsymbol{p}$ of CH. Therefore, for any $t' > t$ and any vertex $\boldsymbol{p}$ of CH, $\boldsymbol{x}_{t'}(r) \notin N_\epsilon(\boldsymbol{p})$. It is a contradiction to the assumption that CH_t converges to CH. □

The following corollary holds by Theorem 2.

Corollary 1. 1. *Let ϕ be a target function such that $0 \leq \alpha(\phi) < 1$. Then $\Phi = \{\phi\}$ is compatible with respect to the convergence problem, or equivalently, ϕ is a convergence algorithm.*

2. *Let Φ and Φ' be two sets of target functions such that $0 \leq \alpha(\Phi) < 1$ and $0 \leq \alpha(\Phi') < 1$ hold. Then $\Phi \cup \Phi'$ is also compatible with respect to the convergence problem, not only Φ and Φ'.*

Corollary 1 states that every target function ϕ such that $\alpha(\phi) < 1$ is a convergence algorithm. However, some target function ϕ such that $\alpha(\phi) = 1$ is not a convergence algorithm. Indeed, CoG_1 is an example as the following proposition states.

Proposition 1. *Target function CoG_1 is not a convergence algorithm. Thus there is a set Φ of target functions such that $\alpha(\Phi) = 1$ and that it is not compatible with respect to the convergence problem.*

Proof. Consider any execution for two robots starting from configuration $P_0 = \{(0,0), (1,0)\}$ (in Z_0). Since both robots take CoG_1 as their target functions and CoG_1 does not move any robot in P_0, it is not a convergence algorithm. □

Let Φ and Φ' be two sets of target functions. If $0 \leq \alpha(\Phi) < 1$ and $0 \leq \alpha(\Phi') < 1$, Φ, Φ', and $\Phi \cup \Phi'$ are all compatible with respect to the convergence problem by Corollary 1. However, the following claim does **not** hold:

> If both of Φ and Φ' are compatible with respect to the convergence problem, so is $\Phi \cup \Phi'$.

To observe this fact, examine the following two target functions ϕ_T and ϕ_S. For a configuration P, define a condition Ψ as follows:

Ψ: $|P| = 7$, $(0,0) \in P$, $P = T \cup S$, $CH(T)$ is an equilateral triangle, $CH(S)$ is a square, $CH(T)$ and $CH(S)$ have the same side length, and finally $CH(T)$ and $CH(S)$ do not overlap each other.

[Target function ϕ_T]

1. If P satisfies Ψ:
   (a) If $(0,0) \in T$, $\phi_T(P) = g(T)/2$, which is the middle point on the line segment connecting $(0,0)$ and $g(T)$.
   (b) If $(0,0) \in S$, $\phi_T(P) = g(P)$.
2. If P does not satisfy Ψ: $\phi_T(P) = g(P)$.

[Target function ϕ_S]

1. If P satisfies Ψ:
   (a) If $(0,0) \in S$, $\phi_S(P) = g(S)/2$, which is the middle point on the line segment connecting $(0,0)$ and $g(S)$.
   (b) If $(0,0) \in T$, $\phi_S(P) = g(P)$.
2. If P does not satisfy Ψ: $\phi_S(P) = g(P)$.

Recall that $g(P)$, $g(T)$, and $g(S)$ are the gravity centers of P, T, and S, respectively, and that when a robot identifies P in Look phase, $(0,0)$ always in P, which corresponds to its current position.

Let us observe that $\alpha(\phi_T) = 1$. Since $\phi_T(P) \in CH(P)$ for all P, $\alpha(\phi_T) \leq 1$. To see that $\alpha(\phi_T) \geq 1$, for any number $0 < a < 1$, consider the following configuration Q_a (see Figure 2 for illustration):

- Q_a satisfies Ψ,
- the side length of each $CH(T)$ and $CH(S)$ is 1,
- $(0,0) \in T$,
- $(0,0)$, $g(T)$, $g(Q_a)$, and $g(S)$ appear on a line ℓ in this order, and
- $dist(g(T), g(Q_a)) = a/(1-a)$.

Then, since $dist((0,0), g(Q_a)) < a/(1-a) + 1 = 1/(1-a)$,

$$\alpha(\phi_T) \geq \frac{dist(g(T)), g(Q_a)}{dist((0,0), g(Q_a))} > \frac{a}{1-a} \cdot \frac{1-a}{1} = a.$$

Thus, $\alpha(\phi_T) = 1$ by the definition of α. By the same argument, $\alpha(\phi_S) = 1$.

Let $\Phi_T = \{\phi_T\}, \Phi_S = \{\phi_S\}$, and $\Phi = \Phi_T \cup \Phi_S = \{\phi_T, \phi_S\}$. Then $\alpha(\Phi_T) = \alpha(\Phi_S) = \alpha(\Phi) = 1$.

Theorem 3. *Both Φ_T and Φ_S are compatible with respect to the convergence problem, but $\Phi = \Phi_T \cup \Phi_S$ is not.*

Proof. We continue to use notations we introduced earlier.

(I) Let us start with showing that Φ_T is compatible with respect to the convergence problem. A proof that Φ_S is also compatible with respect to the convergence problem is similar.

Figure 2: A configuration Q_a which achieves $\alpha(\phi_T) > a$. Unfilled circles represent robot positions and filled circles are the gravity centers of T, S, and Q_a.

All robots take ϕ_T as their target functions. Let $\mathcal{E} : P_0, P_1, \ldots$ be any execution starting from any initial configuration P_0. Since $CH_{t+1} \subseteq CH_t$ by the definition of ϕ_T, CH_t converges from outside to a convex k-gon for some positive integer k. We show $k = 1$. When the number n of robots is not 7, ϕ_T = CoG, and P_t converges to a point by Theorem 1. Thus, we concentrate on the case $n = 7$.

The proof for $n = 7$ is by a contradiction. We assume $k \geq 2$ and derive a contradiction. Let L be the minimum distance between two (distinct) vertices of CH. Then, for any $0 < \epsilon \ll L$, there is a time t_0 such that, for any time $t \geq t_0$, $CH \subseteq CH_t \subseteq N_\epsilon(CH)$.

Suppose that P_t $(t \geq t_0)$ satisfies Ψ. By definition $P_t = T_t \cup S_t$, where $CH(T_t)$ is an equilateral triangle and $CH(S_t)$ is a square. If P_{t+1} satisfies Ψ, at t, no robots in S_t are activated (to maintain a square at P_{t+1}), and all robots in T_t are activated (to maintain an equilateral triangle at P_{t+1}). Then, the side lengths of $CH(T_{t+1})$ and $CH(S_{t+1})$ are different, which implies that P_{t+1} does not satisfy Φ.

Suppose that P_t $(t \geq t_0)$ does not satisfy Ψ. If $P_{t'}$ does not satisfy Ψ for all $t' \geq t$, then P_t converges to a point by Theorem 1, which is a contradiction. Thus, there is an $t'(> t)$ such that $P_{t'}$ satisfies Ψ (and $P_{t'+1}$ does not). Hence, there are infinitely many time instances $s \geq t_0$ such that P_s does not satisfy Ψ, but P_{s+1} does.

First, consider the case $k = 2$. That is, CH is a line segment $\overline{\boldsymbol{pq}}$ of length L connecting distinct points $\boldsymbol{p}$ and $\boldsymbol{q}$. We derive a contradiction by showing that such an $s(\geq t_0)$ never exists.

Since $P_s \neq P_{s+1}$ and P_s does not satisfy Ψ, there is a robot r who moves to $\boldsymbol{g_s} = g(P_s)$ at s. Since P_{s+1} satisfies Ψ, in P_{s+1}, r at $\boldsymbol{g_s}$ is a part of an equilateral triangle $CH(T_{s+1})$ or a square $CH(S_{s+1})$, whose side has length at most 2ϵ. Since $CH \subseteq CH_{s+1} \subseteq N_\epsilon(CH)$, there are robots $r_{\boldsymbol{p}} \in N_\epsilon(\boldsymbol{p})$ and $r_{\boldsymbol{q}} \in N_\epsilon(\boldsymbol{q})$, each of which is a part of T_{s+1} or S_{s+1} whose side has length at most 2ϵ, since P_{s+1} satisfies Ψ. However, it is a contradiction, since $\min\{dist(\boldsymbol{p}, \boldsymbol{g}_s), dist(\boldsymbol{q}, \boldsymbol{g}_s)\} > L/7 - \epsilon \gg 4\epsilon$.

Next consider the case of $3 \leq k \leq 7$. Let $\boldsymbol{p}_0, \boldsymbol{p}_1, \ldots, \boldsymbol{p}_{k-1}$ be the vertices of CH aligned counter-clockwise on the boundary. For any pair (i, j) $(0 \leq i < j \leq k-1)$, let $L_{(i,j)} = dist(\boldsymbol{p}_i, \boldsymbol{p}_j)$, and $L = \min_{0 \leq i < j \leq k-1} L_{(i,j)}$. We derive a contradiction by showing that there are only a finite number of such time instances s (even if there are).

Consider any vertex $\boldsymbol{p}$ of CH. For any $t > t_0$, there is a robot $r \in N_\epsilon(\boldsymbol{p})$. Suppose that r is activated at t. By the definition of ϕ_T, r moves either to $\boldsymbol{g}_t (= g(P_t))$ or to $g(T_t)$ (when P_t satisfies Ψ and r is a part of T_t). Since $\min_{0 \leq i \leq k-1}\{dist(\boldsymbol{p}_i, \boldsymbol{g}_t)\} > L/7 - \epsilon \gg \epsilon$, $\boldsymbol{g}_t \notin N_\epsilon(\boldsymbol{p})$. Suppose that it moves to $\boldsymbol{g}_t$. Since the side length of $CH(T_t)$ is at least $L - 2\epsilon$, $dist(g(T_t), \boldsymbol{p}) \gg \epsilon$. Hence, $g(T_t) \notin N_\epsilon(\boldsymbol{p})$. Thus r is not in $N_\epsilon(\boldsymbol{p})$ at $t+1$.

We next show that any robot r not in $N_\epsilon(\boldsymbol{p})$ at t will never be in $N_\epsilon(\boldsymbol{p})$, forever. To derive a contradiction, suppose that there is a $t_1 > t$ such that r is not in $N_\epsilon(\boldsymbol{p})$ at t_1, but is in $N_\epsilon(\boldsymbol{p})$ at t_1+1. By definition, r moved either to $\boldsymbol{g}_{t_1}$ or to $g(T_{t_1})$ at t_1. Obviously, r did not move to $\boldsymbol{g}_{t_1}$, since $\boldsymbol{g}_{t_1} \notin N_\epsilon(\boldsymbol{p})$. Thus, P_{t_1} satisfies Ψ, and r moved to $g(T_{t_1})$. However, it is a contradiction, since, as mentioned, $dist(g(T_{t_1}), \boldsymbol{p}) \gg \epsilon$, and hence r is not in $N_\epsilon(\boldsymbol{p})$ at $t_1 + 1$.

Now, we conclude that if there are infinite number of such time instances s, then eventually there will be no robots in $N_\epsilon(\boldsymbol{p})$. It is a contradiction.

(II) We show that Φ is not compatible with respect to the convergence problem. Suppose that $P_0 = T_0 \cup S_0$ satisfies Ψ, and that the robots in T_0 (resp. S_0) take ϕ_T (resp. ϕ_S) as their target functions. Suppose that the scheduler is $\mathcal{FSYNC}$, i.e., all robots are activated at every time t. (Note that the $\mathcal{FSYNC}$ scheduler always produce an $\mathcal{SSYNC}$ schedule.) Then it is easy to observe that P_t converges to $\{g(T_0), g(S_0)\}$, and does not converge to a point since $g(T_0) \neq g(S_0)$. □

4. Convergence in the presence of at most one crash failure

We consider the convergence problem in the presence of at most one crash failure in this section. We investigate two problems, the fault-tolerant $(n,1)$-convergence problem (FC(1)) and the fault-tolerant $(n,1)$-convergence problem to a point (FC(1)-PO). There is an algorithm for the FC(1) [13]. Obviously, the FC(1)-PO is not easier than the FC(1), since if all robots (including a faulty one) converge to a point, then all non-faulty robots converge to a point. We have the following theorem, which implies that there is an algorithm for the FC(1)-PO.

Theorem 4. *Let Φ be a set of target functions, and assume that $0 \leq \alpha(\Phi) < 1$. Then Φ is compatible with respect to the* FC(1)-PO*.*

Proof. Let $\phi_i \in \Phi$ be the target function taken by robot r_i for $i = 1, 2, \ldots, n$. Let $\alpha(\phi_i) = \alpha_i$ and $\alpha = \max_{1 \leq i \leq n} \alpha_i$. Then $\alpha \leq \alpha(\Phi) < 1$. Let $\mathcal{E} : P_0, P_1, \ldots$ be any execution starting from any initial configuration P_0. Since $\alpha \leq \alpha(\Phi) < 1$, $CH_{t+1} \subseteq CH_t$, which implies that CH_t converges to a convex k-gon CH for some $k \geq 1$, regardless of whether or not there is a faulty robot. We assume that $k \geq 2$, and derive a contradiction. Since CH_t converges to CH, for any small number $0 < \epsilon \ll (1-\alpha)L/n$, there is a time t_0 such that for all $t > t_0$, $CH \subseteq CH_t \subseteq N_\epsilon(CH)$, where L is the minimum distance between two (distinct) vertices of CH.

By repeating the proof of Theorem 2, we can show that there is a time $t' > t$ such that, for any non-faulty robot r and any vertex $\boldsymbol{p}$ of CH, $\boldsymbol{x}_{t'}(r) \notin N_\epsilon(\boldsymbol{p})$. There is at most one faulty robot, and $N_\epsilon(\boldsymbol{p}) \cap N_\epsilon(\boldsymbol{p}') = \emptyset$ for any two distinct vertices $\boldsymbol{p}$ and $\boldsymbol{p}'$ of CH. Since $k \geq 2$, there is a vertex $\boldsymbol{p}$ such that, for any robot r, $\boldsymbol{x}_{t'}(r) \notin N_\epsilon(\boldsymbol{p})$, which is a contradiction. □

Immediately, we have the following corollary.

Corollary 2. *Let Φ be a set of target functions, and assume that $0 \leq \alpha(\Phi) < 1$. Then Φ is compatible with respect to the* FC(1).

Next we reconsider the target functions ϕ_T and ϕ_S which we introduced in Section 3. Recall that $\alpha(\Phi_T) = \alpha(\Phi_S) = \alpha(\Phi) = 1$.

Theorem 5. *Both $\Phi_T = \{\phi_T\}$ and $\Phi_S = \{\phi_S\}$ are compatible with respect to the* FC(1)-PO*. However, $\Phi = \Phi_T \cup \Phi_S$ is not.*

Proof. (I) We show that Φ_T is compatible with respect to the FC(1)-PO. A proof for Φ_S is similar. If there is no faulty robot, all robots converge to a point by Theorem 3. Suppose that a robot r_F is faulty at time t. The proof is almost the same as the proof (I) of Theorem 3.

Let $\mathcal{E} : P_0, P_1, \ldots$ be any execution starting from any initial configuration P_0. Then CH_t converges to a convex k-gon CH for some positive integer k, even if there is a faulty robot r_F and $\alpha(\Phi_T) = 1$, since $CH_{t+1} \subseteq CH_t$ holds for all $t \geq 0$. Let L be the minimum distance between two (distinct) vertices of CH. Then, any $0 < \epsilon \ll L$, there is a time t_0 such that, for any time $t \geq t_0$, $CH \subseteq CH_t \subseteq N_\epsilon(CH)$. Following the proof (I) of Theorem 3, if P_t $(t \geq t_0)$ satisfies Ψ, then P_{t+1} does not satisfy Φ, and for infinitely many time instants $s > t_0$, P_s does not satisfy Ψ, but P_{s+1} does.

First consider the case $k = 2$. Since there is a time $s > t_0$ such that P_s does not satisfy Ψ, but P_{s+1} does. At s, a robot $r (\neq r_F)$ must move to $\boldsymbol{g}_s = g(P_s)$. Then we can derive a contradiction, as in the proof of Theorem 3.

Next consider the case $3 \leq k \leq 7$. Following the proof (I) of Theorem 3, we conclude that, if a vertex $\boldsymbol{p}_i$ of CH does not include r_F at some $t > t_0$, then eventually no robots will be in $N_\epsilon(\boldsymbol{p}_i)$, provided that there are infinitely many time instants $s > t_0$ such that P_s does not satisfy Ψ, but P_{s+1} does. It is a contradiction, since $k \geq 3$ and such a vertex $\boldsymbol{p}_i$ exists.

(II) To observe that Φ is not compatible with respect to FC(1)-PO, consider the case in which no faulty robot exists. Then all robots need to converge to a point. However, the execution of Φ converges to two points when P_0 satisfies Ψ, the robots in T take ϕ_T, the robots in S take ϕ_S, and the scheduler is $\mathcal{FSYNC}$. □

Observe that Φ is compatible with respect to the FC(1)-PO, if a robot certainly crashes.

5. FC(f) for $f \geq 2$

We go on the fault tolerant (n, f)-convergence problem (FC(f)) for $f \geq 2$. Since CoG is a fault tolerant (n, f)-convergence algorithm for all $f \leq n-2$ [13], and the problem is obviously solvable by CoG when exactly $f = n-1$ robots crash, the next theorem holds.

Theorem 6 ([13]). *Suppose that $f \leq n-1$. Set $\Phi_0 = \{\mathrm{CoG}\}$ is compatible with respect to the* FC(f)*, or equivalently, a set Φ of target functions is compatible with respect to the* FC(f)*, if $\alpha(\Phi) = 0$.*

We show how the FC(f) for $f \geq 2$ is different from the FC(1) (and the FC(1)-PO) from the viewpoint of compatibility. Corollary 2 states that every set Φ of target functions such that $0 \leq \alpha(\Phi) < 1$ is compatible with respect to the FC(1). We show that, for any $2 \leq f \leq n-1$ and $0 < \alpha < 1$, there is a set Φ of two target functions such that (1) $\alpha(\Phi) = \alpha$, (2) each target function in Φ is compatible with respect to the FC(f), but (3) Φ is not compatible with respect to the FC(f).

Consider first the following target function $\xi_{(\alpha,4)}$ for four robots, where $0 < \alpha < 1$. For a configuration P, define a condition Ψ by a conjunction of two conditions (i) and (ii):

Ψ: (i) $P = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3, \boldsymbol{p}_4\} \subseteq \overline{\boldsymbol{p}_1\boldsymbol{p}_4}$, where $\boldsymbol{p}_1$, $\boldsymbol{p}_2$, $\boldsymbol{p}_3$, and $\boldsymbol{p}_4$ are distinct and aligned on $\overline{\boldsymbol{p}_1\boldsymbol{p}_4}$ in this order.

(ii) $dist(\boldsymbol{p}_1, \boldsymbol{p}_2) = L/2$ and $dist(\boldsymbol{p}_2, \boldsymbol{p}_3) = 2\alpha L/(\alpha + 3)$, where $L = dist(\boldsymbol{p}_1, \boldsymbol{p}_4)$.

[Target function $\xi_{(\alpha,4)}$]

1. Suppose that P satisfies Ψ.
   - **(a)** $\xi_{(\alpha,4)}(P) = \boldsymbol{p}_3$, if $\boldsymbol{p}_2 = (0, 0)$,
   - **(b)** $\xi_{(\alpha,4)}(P) = (0, 0)$, if $\boldsymbol{p}_3 = (0, 0)$, and
   - **(c)** $\xi_{(\alpha,4)}(P) = g(P)$, otherwise.
2. Suppose that P does not satisfy Ψ. Then $\xi_{(\alpha,4)}(P) = g(P)$.

The following lemma holds.

Lemma 1. *Set $\Phi = \{\xi_{(\alpha,4)}\}$ is compatible with respect to the fault tolerant $(4, f)$-convergence problem, where $0 < \alpha(\Phi) = \alpha < 1$ and $f \leq 3$.*

Proof. We first show $\alpha(\xi_{(\alpha,4)}) = \alpha$, and hence $\alpha(\Phi) = \alpha$. If P does not satisfy ψ, all activated robots move to $g(P) \in 0 * CH(P)$. If P satisfies Ψ, a robot moves either to $\boldsymbol{p}_3$ or to $g(P)$. The distance between $g(P)$ and $\boldsymbol{p}_1$ is

$$\frac{L}{4}\left(\frac{1}{2} + \left(\frac{1}{2} + \frac{2\alpha}{\alpha+3}\right) + 1\right) = \frac{L}{2} + \frac{L}{4} \cdot \frac{2\alpha}{2\alpha+6}.$$

Hence,

$$\alpha(\xi_{(\alpha,4)}) = \frac{dist(g(P), \boldsymbol{p}_3)}{dist(g(P), \boldsymbol{p}_4)} = \frac{\frac{3}{4} \cdot \frac{2\alpha}{\alpha+3}}{\frac{1}{2} - \frac{1}{4} \cdot \frac{2\alpha}{2\alpha+6}} = \alpha.$$

Let $\mathcal{E} : P_0, P_1, \ldots$ be any execution starting from any initial configuration P_0. If CH_t is a convex k-gon with $k \geq 3$, then $\xi_{(\alpha,4)}(P_t) = \mathrm{CoG}(P_t)$. Thus if there is no time t such that CH_t is a convex k-gon with $k \leq 2$, all non-faulty robots converge to a point by Theorem 6. On the other hand, if CH_t is a line segment for some time t, then $CH_{t+1} \subseteq CH_t$ by the definition of $\xi_{(\alpha,4)}$, which implies that $CH_{t'}$ is also a line segment for all $t' \geq t$.

We assume without loss of generality that CH_0 is a line segment. Since $CH_{t+1} \subseteq CH_t$, CH_t converges to a line segment $CH = \overline{\boldsymbol{pq}}$. We denote the length of CH_t (resp. CH) by L_t (resp. L). That is, L_t is non-increasing, i.e., $L_{t+1} \leq L_t$,[8] and converges to L. Without loss of generality, we may assume that all faulty robots have crashed at time 0.

(I) Suppose that at most one robot is faulty. Then CH is a point, and hence all (non-faulty) robots converge to the point. To show this, we assume that CH is not a point, i.e., $\boldsymbol{p} \neq \boldsymbol{q}$ and $L > 0$, and derive a contradiction.

Since CH_t converges to CH, for any $0 < \epsilon \ll (1-\alpha)L$, there is a time $t_0 \geq 0$ such that, for all $t > t_0$, $CH \subseteq CH_t \subseteq N_\epsilon(CH)$. Let $P_t = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3, \boldsymbol{p}_4\}$. Without loss of generality, we assume that $\boldsymbol{p}_1$, $\boldsymbol{p}_2$, $\boldsymbol{p}_3$, and $\boldsymbol{p}_4$ are aligned in $\overline{\boldsymbol{p}_1\boldsymbol{p}_4}$ in this order, where $\boldsymbol{p}_1 \in N_\epsilon(\boldsymbol{p})$ and $\boldsymbol{p}_4 \in N_\epsilon(\boldsymbol{q})$.

Suppose that $\boldsymbol{x}_t(r) = \boldsymbol{p}_1$ and $\boldsymbol{x}_t(r') = \boldsymbol{p}_4$. Since either r or r' is non-faulty, we assume that r is non-faulty without loss of generality. Since r is non-faulty, at some time $t' \geq t$, it is activated and move to $g(P_{t'}) \notin N_\epsilon(\boldsymbol{p}) \cup N_\epsilon(\boldsymbol{q})$. Furthermore, it will never return to $N_\epsilon(\boldsymbol{p}) \cup N_\epsilon(\boldsymbol{q})$ again. It is because, for all $t'' \geq t'$, (i) $g(P_{t''}) \notin N_\epsilon(\boldsymbol{p}) \cup N_\epsilon(\boldsymbol{q})$, and (ii) if $P_{t''}$ satisfies Ψ, and $\boldsymbol{x}_{t''}(r)$ is one of the middle points of CH_t ($\boldsymbol{p}_2$ or $\boldsymbol{p}_3$ in the definition of $\xi_{(\alpha,4)}$), the target point of r ($\boldsymbol{p}_3$ in the definition of $\xi_{(\alpha,4)}$) does not belong to $N_\epsilon(\boldsymbol{p}) \cup N_\epsilon(\boldsymbol{q})$, since $\epsilon \ll (1-\alpha)L$.

Thus all non-faulty robots eventually do not exist in $N_\epsilon(\boldsymbol{p}) \cup N_\epsilon(\boldsymbol{q})$, which contradict to the assumption that CH_t converges to $\overline{\boldsymbol{pq}}$.

(II) Suppose next that at least two robots are faulty. We show that all non-faulty robots converge to a point. If more than one robot has crashed at the same position at time 0, P_t does not satisfy Ψ for all $t > 0$, and thus all non-faulty robots converge to a point by Theorem 6.

Consider the case in which all faulty robots have crashed at distinct positions. Suppose that CH_t converges to $CH = \overline{\boldsymbol{pq}}$ for some distinct points $\boldsymbol{p}$ and $\boldsymbol{q}$, since CH cannot be a point. By the same argument as in (I), we can assume without loss of generality that $\boldsymbol{p}$ and $\boldsymbol{q}$ are the positions of faulty robots at time 0, and that $P_0 = \{\boldsymbol{p}, \boldsymbol{p}_2, \boldsymbol{p}_3, \boldsymbol{q}\}$, where $\boldsymbol{p}$, $\boldsymbol{p}_2$, $\boldsymbol{p}_3$, and $\boldsymbol{q}$ are distinct and aligned on $\overline{\boldsymbol{pq}}$ in this order, that is, $CH_t = CH$ for all $t \geq 0$.

[8] If P satisfies Ψ and no robots at $\boldsymbol{p}_1$ and $\boldsymbol{p}_4$ are activated, $L_{t+1} = L_t$ holds. That is, L_t is not strictly decreasing.

There are two robots r and r' such that $\boldsymbol{x}_0(r) = \boldsymbol{p}_2$ and $\boldsymbol{x}_0(r') = \boldsymbol{p}_3$, one of which may be faulty. If P_t does not satisfy Ψ for all $t \geq 0$, all non-faulty robots converge to a point by Theorem 6.

Without loss of generality, we assume that P_0 satisfies Ψ. If r is faulty, since r' does not move by the definition of $\xi_{(\alpha,4)}$, $P_0 = P_t$ for all $t \geq 0$, i.e., all non-faulty robots converges to $\boldsymbol{p}_3$. Otherwise, if r is non-faulty, r is eventually activated and moves to $\boldsymbol{p}_3$ at some time $t \geq 0$.

We observe that $P_{t'}$ does not satisfy Ψ for all $t' > t$. In P_{t+1}, $\boldsymbol{x}_{t+1}(r)$ and $\boldsymbol{x}_{t+1}(r')$ are located in $\overline{\boldsymbol{hq}}$ (excluding $\boldsymbol{h}$), where $\boldsymbol{h} = (\boldsymbol{p} + \boldsymbol{q})/2$ is the middle point of $CH = \overline{\boldsymbol{pq}}$. Then P_{t+1} does not satisfy Φ. Since $g(P_{t+1})$ is also in $\overline{\boldsymbol{hq}}$ (excluding $\boldsymbol{h}$), by a simple induction, for all $t' > t$, $\boldsymbol{x}_{t'}(r)$ and $\boldsymbol{x}_{t'}(r')$ are located in $\overline{\boldsymbol{hq}}$, and hence $P_{t'}$ does not satisfy Ψ. We can conclude that all non-faulty robots converge to a point by Theorem 6. $\Box$

For $0 < \alpha < 1$, we next consider the following target function $\xi'_{(\alpha,4)}$ for four robots. We use the condition Ψ defined above.

[Target function $\xi'_{(\alpha,4)}$]

1. Suppose that P satisfies the condition Ψ.
   - **(a)** $\xi'_{(\alpha,4)}(P) = (0,0)$, if $\boldsymbol{p}_2 = (0,0)$,
   - **(b)** $\xi'_{(\alpha,4)}(P) = \alpha\boldsymbol{p}_1 + (1-\alpha)g(P)$, if $\boldsymbol{p}_3 = (0,0)$, and
   - **(c)** $\xi'_{(\alpha,4)}(P) = g(P)$, otherwise.
2. Suppose that P does not satisfy Ψ. Then $\xi'_{(\alpha,4)}(P) = g(P)$.

By an argument similar to the proof of Lemma 1, we can show that the set $\Phi' = \{\xi'_{(\alpha,4)}\}$ is compatible with respect to the fault tolerant $(4, f)$-convergence problem, and that $\alpha(\Phi') = \alpha(\xi'_{(\alpha,4)}) = \alpha < 1$. The proof only differs from that for Lemma 1 in the last paragraph; we observe that $P_{t'}$ does not satisfy Ψ for all $t' > t$, since in P_{t+1}, $\boldsymbol{x}_{t+1}(r)$ and $\boldsymbol{x}_{t+1}(r')$ are located in $\overline{\boldsymbol{ph}}$ (excluding $\boldsymbol{h}$), where $\boldsymbol{h}$ corresponds to $\boldsymbol{p}_3$ in Ψ.

Corollary 3. *Set $\Phi' = \{\xi'_{(\alpha,4)}\}$ is compatible with respect to the fault tolerant $(4, f)$-convergence problem, where $0 < \alpha(\Phi) = \alpha < 1$ and $f \leq 3$.*

Lemma 2. *For any $2 \leq f \leq 3$, $\Phi \cup \Phi' = \{\xi_{(\alpha,4)}, \xi'_{(\alpha,4)}\}$ is not compatible with respect to the fault tolerant $(4, f)$-convergence problem. Here, $\alpha(\Phi \cup \Phi') = \alpha(\Phi) = \alpha(\Phi') = \alpha < 1$.*

Proof. We assume the followings: P_0 satisfies the condition Ψ, the target function of the robots at $\boldsymbol{p}_1$, $\boldsymbol{p}_3$, and $\boldsymbol{p}_4$ is $\xi_{(\alpha,4)}$, the one of the robot at $\boldsymbol{p}_2$ is $\xi'_{(\alpha,4)}$, the scheduler is $\mathcal{FSYNC}$, and the robots at $\boldsymbol{p}_1$ and $\boldsymbol{p}_4$ have already crashed at time 0.

Then by the definitions of $\xi_{(\alpha,4)}$ and $\xi'_{(\alpha,4)}$, $P_t = P_0$ for all $t \geq 0$. Thus the lemma holds. $\Box$

Let us extend this lemma to general f. We use target functions $\xi_{(\alpha,n)}$ and $\xi'_{(\alpha,n)}$ which are easy extensions of $\xi_{(\alpha,4)}$ and $\xi'_{(\alpha,4)}$. Let $\ell = \lfloor (n-2)/2 \rfloor$ and

$\ell' = \lceil (n-2)/2 \rceil$. Thus, $\ell + \ell' = n-2$. For a configuration P, define a condition Ψ^+ as the conjunction of two conditions (i) and (ii).

Ψ^+: (i) $P = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \dots, \boldsymbol{p}_n\} \subseteq \overline{\boldsymbol{p}_1\boldsymbol{p}_n}$, where $\boldsymbol{p}_1$, $\boldsymbol{p}_{\ell+1}$, $\boldsymbol{p}_{\ell+2}$, $\boldsymbol{p}_{\ell+3}$ are distinct and aligned on $\overline{\boldsymbol{p}_1\boldsymbol{p}_n}$ in this order, $\boldsymbol{p}_1 = \boldsymbol{p}_2 = \cdots = \boldsymbol{p}_\ell$, i.e., the multiplicity of $\boldsymbol{p}_1$ is ℓ, and $\boldsymbol{p}_{\ell+3} = \boldsymbol{p}_{\ell+4} = \cdots = \boldsymbol{p}_n$, i.e., the multiplicity of $\boldsymbol{p}_{\ell+3}$ is ℓ'.

(ii) Let $L = dist(\boldsymbol{p}_1, \boldsymbol{p}_n)$. Then $dist(\boldsymbol{p}_1, \boldsymbol{p}_{\ell+1}) = L/2$. If n is even, $dist(\boldsymbol{p}_{\ell+1}, \boldsymbol{p}_{\ell+2}) = \alpha n L/(2(\alpha + n - 1))$; otherwise, if it is odd, $dist(\boldsymbol{p}_{\ell+1}, \boldsymbol{p}_{\ell+2}) = (\alpha(n-1)+1)L/(2(\alpha + n - 1))$.

[Target function $\xi_{(\alpha,n)}$]

1. Suppose that P satisfies Ψ^+.
   (a) $\xi_{(\alpha,n)}(P) = \boldsymbol{p}_{\ell+2}$, if $\boldsymbol{p}_{\ell+1} = (0,0)$,
   (b) $\xi_{(\alpha,n)}(P) = (0,0)$, if $\boldsymbol{p}_{\ell+2} = (0,0)$, and
   (c) $\xi_{(\alpha,n)}(P) = g(P)$, otherwise.
2. Suppose that P does not satisfy Ψ^+. Then $\xi_{(\alpha,n)}(P) = g(P)$.

[Target function $\xi'_{(\alpha,n)}$]

1. Suppose that P satisfies Ψ^+.
   (a) $\xi'_{(\alpha,n)}(P) = (0,0)$, if $\boldsymbol{p}_{\ell+1} = (0,0)$,
   (b) $\xi'_{(\alpha,n)}(P) = \alpha \boldsymbol{p}_1 + (1-\alpha) g(P)$, if $\boldsymbol{p}_{\ell+2} = (0,0)$, and
   (c) $\xi'_{(\alpha,n)}(P) = g(P)$, otherwise.
2. Suppose that P does not satisfy Ψ^+ Then $\xi'_{(\alpha,n)}(P) = g(P)$.

Theorem 7. *For any $2 \le f \le n-1$ and $0 < \alpha < 1$, there are two target functions $\xi_{(\alpha,n)}$ and $\xi'_{(\alpha,n)}$ such that (1) $\alpha(\xi_{(\alpha,n)}) = \alpha(\xi'_{(\alpha,n)}) = \alpha$, (2) both of $\Phi = \{\xi_{(\alpha,n)}\}$ and $\Phi' = \{\xi'_{(\alpha,n)}\}$ are compatible with respect to the FC(f), but (3) $\Phi \cup \Phi'$ is not.*

Proof. First confirm that $\xi_{(\alpha,n)}$ (resp. $\xi'_{(\alpha,n)}$) is an extension of $\xi_{(\alpha,4)}$ (resp. $\xi'_{(\alpha,4)}$) in the sense that $\xi_{(\alpha,n)}$ (resp. $\xi'_{(\alpha,n)}$) for $n = 4$ is exactly the same as $\xi_{(\alpha,4)}$ (resp. $\xi'_{(\alpha,4)}$) defined earlier.

(I) We first show that $\alpha(\xi_{(\alpha,n)}) = \alpha$ and Φ is compatible with respect to the FC(f). When n is even, by the definition of $g(P)$, $dist(\boldsymbol{p}_{\ell+1}, g(P)) = \alpha L/(2(\alpha + n-1))$ and $dist(g(P), \boldsymbol{p}_{\ell+2}) = \alpha(n-1)L/(2(\alpha+n-1))$. Since $dist(\boldsymbol{p}_{\ell+1}, g(P)) < dist(g(P), \boldsymbol{p}_{\ell+2})$, and $dist(\boldsymbol{p}_1, g(P)) > dist(g(P), \boldsymbol{p}_n)$, following the proof of $\alpha(\xi_{(\alpha,4)}) = \alpha$ in Lemma 1,

$$\alpha(\xi_{(\alpha,n)}) = \frac{dist(g(P), \boldsymbol{p}_{\ell+2})}{dist(g(P), \boldsymbol{p}_n)} = \alpha.$$

When n is odd, by a similar argument, again,

$$\alpha(\xi_{(\alpha,n)}) = \frac{dist(g(P), \boldsymbol{p}_{\ell+2})}{dist(g(P), \boldsymbol{p}_n)} = \alpha.$$

Thus, $\alpha(\Phi) = \alpha$.

Let $\mathcal{E} : P_0, P_1, \ldots$ be any execution starting from any initial configuration P_0, where at most f robots crash. If there is a $t_0 \geq 0$ such that P_t does not satisfy Ψ^+ for all $t \geq t_0$, then $\xi_{(\alpha,n)}(P_t) = \mathrm{CoG}(P_t)$, and all non-faulty robots converge to a point by Theorem 6. Thus, without loss of generality, we can assume that P_0 satisfies Ψ^+. That is, CH_0 is a line segment.

By the definition of $\xi_{(\alpha,n)}$, for any $t \geq 0$, CH_t is a line segment and converges to a line segment $CH = \overline{\boldsymbol{pq}}$, since $CH_{t+1} \subseteq CH_t$. We denote the length of CH_t (resp. CH) by L_t (resp. L). That is, L_t is non-increasing, and converges to L. Without loss of generality, we can also assume that all faulty robots have already crashed at time 0.

Let $P_0 = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_n\}$, where the multiplicities of $\boldsymbol{p}_1$ and $\boldsymbol{p}_n$ are ℓ and ℓ', respectively. Then, all of the ℓ robots at $\boldsymbol{p}_1$ are either non-faulty or crashed, since otherwise, P_t does not satisfy Ψ^+ for all $t \geq t_0$, where at t_0, a robot at $\boldsymbol{p}_1$ is activated to move to to $\boldsymbol{g}_{t_0}(= g(P_{t_0})$. By the same reason, all of the ℓ' robots at $\boldsymbol{p}_n$ are either non-faulty or crashed.

If $L = 0$, the FC(f) has been solved. Thus, we assume $L > 0$, and let $CH = \overline{\boldsymbol{pq}}$ for some $\boldsymbol{p} \neq \boldsymbol{q}$. Then, there is a $t > 0$ such that P_t contains $\boldsymbol{p}$ with multiplicity ℓ and $\boldsymbol{q}$ with multiplicity ℓ, and all of the robots at $\boldsymbol{p}$ and $\boldsymbol{q}$ are crashed.

Since there are at most two non-faulty robots in CH, they eventually converges to a point by the definition of $\xi_{(\alpha,n)}$. Thus Φ is compatible with respect to the FC(f).

By the same argument, $\alpha(\xi'_{(\alpha,n)}) = \alpha$, and Φ' is compatible with respect to the FC(f).

(II) We next show that $\Phi \cup \Phi'$ is not compatible with respect to the FC(f). The proof is similar to that of Lemma 2.

Assume the followings: P_0 satisfies the condition Ψ^+, the target function of the robots except the one at $\boldsymbol{p}_{\ell+1}$ is $\xi_{(\alpha,n)}$, and the target function of the robot at $\boldsymbol{p}_{\ell+1}$ is $\xi'_{(\alpha,n)}$, the scheduler is $\mathcal{FSYNC}$, and all robots except the two robots at $\boldsymbol{p}_{\ell+1}$ and $\boldsymbol{p}_{\ell+2}$ have already crashed at time 0.

Then by the definitions of $\xi_{(\alpha,n)}$ and $\xi'_{(\alpha,n)}$, $P_t = P_0$ for all $t \geq 0$. □

Recall Corollary 2. Theorem 7 states an interesting difference between the FC(f) for $f \geq 2$ and the FC(1). Before closing this section, we examine the case $\alpha = 1$ by using the above theorem.

Corollary 4. *For any $2 \leq f \leq n-1$, there are two target functions $\xi_{(1,n)}$ and $\xi'_{(1,n)}$ such that (1) $\alpha(\xi_{(1,n)}) = \alpha(\xi'_{(1,n)}) = 1$, (2) both of $\Phi = \{\xi_{(1,n)}\}$ and $\Phi' = \{\xi'_{(1,n)}\}$ are compatible with respect to the* FC(f)*, but (3) $\Phi \cup \Phi'$ is not.*

Proof. We construct target functions $\xi_{(1,n)}$ and $\xi'_{(1,n)}$ as follows: First, consider the following target function $\xi^*_{(1,3)}$ for three robots.

[Target function $\xi^*_{(1,3)}$]

1. If $P = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3\}$ satisfies (i) $P \subseteq \overline{\boldsymbol{p}_1\boldsymbol{p}_3}$, where $\boldsymbol{p}_1$, $\boldsymbol{p}_2$, $\boldsymbol{p}_3$ are distinct and aligned on $\overline{\boldsymbol{p}_1\boldsymbol{p}_3}$ in this order, and (ii) $dist(\boldsymbol{p}_1, \boldsymbol{p}_2) = 9L/10$ and $dist(\boldsymbol{p}_2, \boldsymbol{p}_3) = L/10$, where $dist(\boldsymbol{p}_1, \boldsymbol{p}_3) = L$,
   (a) $\xi^*_{(1,3)}(P) = \boldsymbol{p}_3$, if $\boldsymbol{p}_1 = (0,0)$,
   (b) $\xi^*_{(1,3)}(P) = g(P)$, otherwise.
2. Otherwise, $\xi^*_{(1,3)}(P) = g(P)$.

By Step 1(a), $\alpha(\xi^*_{(1,3)}) = 1$, and $\{\xi^*_{(1,3)}\}$ is obviously an algorithm for the fault tolerant $(3,2)$-convergence problem.

Let $\xi_{(1,n)}$ and $\xi'_{(1,n)}$ be respectively constructed from $\xi_{(\alpha,n)}$ and $\xi'_{(\alpha,n)}$ by replacing $\xi_{(\alpha,3)}$ and $\xi'_{(\alpha,3)}$ with $\xi^*_{(1,3)}$. Then $\alpha(\xi_{(1,n)}) = \alpha(\xi'_{(1,n)}) = 1$. By Theorem 7, $\xi_{(1,n)}$ and $\xi'_{(1,n)}$ are algorithms for the FC(f), but $\{\xi_{(1,n)}, \xi'_{(1,n)}\}$ is not compatible with respect to the FC(f). □

6. FC(f)-CP for $f \geq 2$

We next investigate the fault tolerant (n,f)-convergence problem to a convex f-gon (FC(f)-CP). The FC(f)-CP is the problem to ensure that, as long as at most f robots crash, in every execution $\mathcal{E} : P_0, P_1, \ldots$, $CH_t = CH(P_t)$ converges to a convex h-gon CH for some $h \leq f$, in such a way that for each vertex of CH there is a robot that converges to the vertex. A convex 2-gon is a line segment. The FC(1)-CP is the FC(1)-PO. The FC(f)-CP seems to be substantially easier than the FC(f) (and the FC(f)-PO), since the convergence of CH_t to a convex f-gon does not always mean the convergence of P_t; a robot may not converge to a point. We have the following theorem.

Theorem 8. *Let Φ be any set of target functions such that $0 \leq \alpha(\Phi) < 1$. Then Φ is compatible with respect to the* FC(f)-CP *for any $2 \leq f \leq n-1$.*

Proof. Let $\phi_i \in \Phi$ be the target function taken by robot r_i, for $i = 1, 2, \ldots, n$. Let $\alpha(\phi_i) = \alpha_i$ and $\alpha = \max_{1 \leq i \leq n} \alpha_i$. Then $\alpha \leq \alpha(\Phi) < 1$.

Let $\mathcal{E} : P_0, P_1, \ldots$ be any execution of Φ starting from any initial configuration P_0. We show that CH_t converges to a convex k-gon CH for some $k \leq f$, provided that at most f robots will crash. Since $\alpha < 1$, CH_t converges to a convex k-gon CH for some $k \geq 1$. We show that $k \leq f$.

To derive a contradiction, we assume that $k \geq f+1$. Let $\boldsymbol{p}_0, \boldsymbol{p}_1, \ldots, \boldsymbol{p}_{k-1}$ be the vertices of CH, and they appear in this order on the boundary of CH counter-clockwise. For any pair $(i,j) (0 \leq i < j \leq k-1)$, let $L_{(i,j)} = dist(\boldsymbol{p}_i, \boldsymbol{p}_j)$ and $L = \min_{0 \leq i < j \leq k-1} L_{(i,j)}$. For any $0 < \epsilon \ll (1-\alpha)L/n$, there is a time t_0 such that, for all $t > t_0$, $CH \subseteq CH_t \subseteq N_\epsilon(CH)$. By definition, for all (i,j), $N_\epsilon(\boldsymbol{p}_i) \cap N_\epsilon(\boldsymbol{p}_j) = \emptyset$.

There is a time t and a vertex $\boldsymbol{p}$ such that $N_\epsilon(\boldsymbol{p})$ does not include a faulty robot in CH_t. Certainly such a $\boldsymbol{p}$ exists, since there are at most f faulty robots, a robot cannot belong to the neighbors of two vertices simultaneously, and $k \geq f+1$.

Consider any non-faulty robot r in $N_\epsilon(\boldsymbol{p})$ at t. Then r is eventually activated at time $t' > t$, and moves inside $\alpha * CH_{t'} \subseteq \alpha * CH_t$. Thus $dist(\boldsymbol{x}_{t'}(r), \boldsymbol{x}_{t'+1}(r)) > (1-\alpha)L/n \gg 2\epsilon$, which implies that there is no time $t'' > t'$ such that $\boldsymbol{x}_{t''}(r) \in N_\epsilon(\boldsymbol{p}_i)$ for any $0 \leq i \leq k-1$. It contradicts to the assumption that CH_t converges to CH, since there is a time such that no robot is in $N_\epsilon(\boldsymbol{p})$.

We next show that for each vertex $\boldsymbol{p}$ of CH, there is a robot r that converges to $\boldsymbol{p}$. By the argument above, if r is non-faulty, it eventually leaves $N_\epsilon(\boldsymbol{p})$ and will never return in $N_\epsilon(\boldsymbol{p}_i)$ for any $0 \leq i \leq k-1$. Thus r is faulty and it crashes at $\boldsymbol{p}$, i.e., r converges to $\boldsymbol{p}$. □

Observation 1. *Let* Φ *and* Φ' *be any set of target functions such that* $\alpha(\Phi) < 1$ *and* $\alpha(\Phi') < 1$ *hold. Then all of* Φ, Φ', *and* $\Phi \cup \Phi'$ *are compatible with respect to the* FC(f)-CP *for all* $2 \leq f \leq n-1$, *since* $\alpha(\Phi \cup \Phi') < 1$.

However, we cannot extend Observation 1 to include the case $\alpha = 1$. as we shall see below.

Consider the following two target functions τ and τ' for four robots. For a configuration P, define a condition Ψ:

Ψ: $P = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3, \boldsymbol{p}_4\}$, and $CH(P)$ is a convex quadrilateral such that $\angle \boldsymbol{p}_1 < \angle \boldsymbol{p}_2 < \angle \boldsymbol{p}_3 < \angle \boldsymbol{p}_4$, where $\angle \boldsymbol{p}_i$ is the angle of vertex $\boldsymbol{p}_i$ of the quadrilateral.

[Target function τ]

1. If $|\overline{P}| \leq 2$, then $\tau(P) = (0,0)$.
2. If P satisfies Ψ, then $\tau(P) = \boldsymbol{p}_1$.
3. Otherwise, $\tau(P) = g(P)$.

[Target function τ']

1. If $|\overline{P}| \leq 2$, then $\tau(P) = (0,0)$.
2. If P satisfies Ψ, then $\tau'(P) = \boldsymbol{p}_4$.
3. Otherwise, $\tau'(P) = g(P)$.

The following theorem holds.

Theorem 9. *Let* $\Phi = \{\tau\}$ *and* $\Phi' = \{\tau'\}$. *Then* $\alpha(\Phi) = \alpha(\Phi') = 1$. *Sets* Φ *and* Φ' *are compatible with respect to the fault tolerant* $(4,2)$*-convergence problem to a line segment, but* $\Phi \cup \Phi'$ *is not.*

Proof. Obviously, $\alpha(\tau) = \alpha(\tau') = 1$. We first show that Φ is compatible with respect to the fault tolerant $(4,2)$-convergence problem to a convex 2-gon, i.e., to a line segment. A proof that Φ' is also compatible with respect to the problem is similar.

Suppose that all robots take τ as their target functions, and let $\mathcal{E} : P_0, P_1, \ldots$ be any execution starting from any initial configuration P_0. If P_t does not satisfy Ψ for all $t \geq 0$, then $\mathcal{E}$ converges to a line segment by Theorem 8, since $\tau(P_t) = g(P_t)$ for all $t \geq 0$.

Without loss of generality, suppose that P_0 satisfies Ψ. If all robots activated at time 0 are either a robot at $\boldsymbol{p}_1$, or a robot which has been crashed, then $P_1 = P_0$, and hence, by the fairness of the scheduler, a robot such that it is not at $\boldsymbol{p}_1$ and is not crashed is eventually activated. Thus, without loss of generality, we assume that at least one such robot is activated at time 0. (It is possible, since there are at most two faulty robots.)

If two or more such robots are activated, then they move to $\boldsymbol{p}_1$, and $|\overline{P_1}| \leq 2$ holds. Since $\tau(P_t) = (0, 0)$ for $t \geq 1$, the execution converges to a line segment.

Suppose that exactly one such robot is activated. Then, $CH(P_1)$ is a triangle, in which two robots are at a vertex of $CH(P_1)$. By a simple induction on t, for all $t \geq 1$, $CH(P_t)$ is either a line segment or a triangle, and does not satisfy Ψ. Thus, $\mathcal{E}$ converges to a line segment, again by Theorem 8.

Next, we show that $\Phi \cup \Phi'$ is not compatible with respect to the fault tolerant $(4, 2)$-convergence problem to a line segment. Suppose that P_0 satisfies Ψ. Consider the case that the robot at $\boldsymbol{p}_1$ (resp. $\boldsymbol{p}_4$) takes τ (resp. τ') as its target function, and the robots at $\boldsymbol{p}_2$ and $\boldsymbol{p}_3$ crash at time 0. Since $\tau(P_0) = (0, 0)$ and $\tau'(P_0) = (0, 0)$, $P_t = P_0$ for all $t \geq 1$, i.e., the execution does not converge to a line segment. □

7. FC(f)-PO for $f \geq 2$

This section investigates the fault tolerant (n, f)-convergence problem to f points (FC(f)-PO) for $f \geq 2$. At a glance, the FC(f)-PO looks to have properties similar to the FC(f), and readers might consider that the former would be easier than the latter, since in the former, all non-faulty robots are not requested to converge to a single point. On the contrary, we shall see that the FC(f)-PO is a formidable problem even when $f = 2$.

7.1. Compatibility

We show a difference between the FC(f) and the FC(f)-PO for $f \geq 2$, from the viewpoint of compatibility.

Theorem 10. *Let $f \geq 2$. Any target function ϕ is not an algorithm for the* FC(f)-PO*, if $0 \leq \alpha(\phi) < 1$, or equivalently, Φ is not compatible with the* FC(f)-PO*, if $0 \leq \alpha(\Phi) < 1$.*

Proof. To derive a contradiction, we assume that there is an algorithm ϕ for the FC(f)-PO such that $\alpha(\phi) < 1$.

Let $\mathcal{R} = \{r_1, r_2, \dots, r_n\}$, where $n \geq f + 1$. Consider an initial configuration P_0 such that points $\boldsymbol{x}_0(r_1), \boldsymbol{x}_0(r_2), \dots, \boldsymbol{x}_0(r_f)$ form a regular f-gon B, and $\boldsymbol{x}_0(r_i)$ is the center of gravity of B, where $i = f + 1, f + 2, \dots, n$.

Consider any execution $\mathcal{E} : P_0, P_1, \dots$, assuming that the robots $r_1, r_2, \dots, r_f$ have already crashed at time 0. Then r_{f+1} must converge to one of the vertices of B. Since $CH_t = B$ for all $t \geq 0$, it is a contradiction since $\alpha(\phi) < 1$. □

Recall that $\Phi = \{\xi_{(\alpha,n)}\}$ and $\Phi' = \{\xi'_{(\alpha,n)}\}$ are compatible with respect to the FC(f) for all $2 \leq f \leq n - 1$ and $0 \leq \alpha < 1$ by Theorem 7. Since $\alpha(\Phi) = \alpha$, by Theorem 10, we have:

Corollary 5. *Neither Φ nor Φ' is compatible with respect to the* FC(f)-PO*, for all $f \geq 2$ and $0 \leq \alpha < 1$.*

It goes without saying that not all target functions ϕ with $\alpha(\phi) = 1$ are algorithms for the FC(2)-PO. For example,

Observation 2. *CoG_1 is not an algorithm for the* FC(2)-PO*.*

7.2. Algorithm for FC(2)-PO

In Section 7.1, we showed that, for any $f \geq 2$, there is no FC(f)-PO algorithm whose scale is less than 1. It is a clear difference between the FC(f)-PO and the FC(f), which is solved, e.g., by CoG$_\alpha$ for any $0 \leq \alpha < 1$. This section proposes an algorithm $\psi_{(n,2)}$ with $\alpha(\psi_{(n,2)}) = 1$ for the FC(2)-PO, and shows its correctness. Unfortunately, proposing an algorithm for FC(f)-PO for an $f \geq 3$ is left as a future work.

7.2.1. Algorithm $\psi_{(3,2)}$

We start with proposing an algorithm $\psi_{(3,2)}$ for solving the fault tolerant $(3, 2)$-convergence problem to two points. Let $>$ be a lexicographic order on R^2 defined as follows: For two distinct points $\boldsymbol{p} = (p_x, p_y)$ and $\boldsymbol{q} = (q_x, q_y)$ in R^2, $\boldsymbol{p} > \boldsymbol{q}$ if and only if either (i) $p_x > q_x$, or (ii) $p_x = q_x$ and $p_y > q_y$ holds.[9]

We classify configurations $P = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3\}$ such that $(0, 0) \in P$ into three types G, L, and T:

G(oal): There are $i, j (i \neq j)$ such that $\boldsymbol{p}_i = \boldsymbol{p}_j$.

L(ine): Three points $\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3$ are distinct, and $\boldsymbol{p}_2 \in \overline{\boldsymbol{p}_1\boldsymbol{p}_3}$. We assume $\boldsymbol{p}_1 > \boldsymbol{p}_3$.

T(riangle): $CH(P)$ is a triangle (not including a line and a point). We assume that $\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3$ appear in this order counter-clockwise on the boundary of the triangle.

[9] We use the same notation $>$ to denote the lexicographic order on R^2 and the order $>$ on R to save the number of notations.

We describe target function $\psi_{(3,2)}$.

[Target function $\psi_{(3,2)}$]

1. If P is type G, $\psi_{(3,2)}(P) = (0,0)$.
2. When P is type L:
   (a) If $\boldsymbol{p}_1 = (0,0)$ or $\boldsymbol{p}_3 = (0,0)$, then $\psi_{(3,2)}(P) = \boldsymbol{p}_2/2$.
   (b) If $\boldsymbol{p}_2 = (0,0)$, $\psi_{(3,2)}(P) = \boldsymbol{p}_1/2$.
3. When P is type T: If $\boldsymbol{p}_i = (0,0)$, then $\psi_{(3,2)}(P) = \boldsymbol{p}_{i+1}/2$, where $\boldsymbol{p}_4 = \boldsymbol{p}_1$.

We show the correctness of $\psi_{(3,2)}$. Let $\mathcal{R} = \{r_1, r_2, r_3\}$. Recall that the x-y local coordinate system Z_i of each robot r_i is right-handed.

Lemma 3. *Target function $\psi_{(3,2)}$ satisfies $\alpha(\psi_{(3,2)}) = 1$, and is an* FC(2)-PO *algorithm for $n = 3$.*

Proof. We observe $\alpha(\psi_{(3,2)}) = 1$. Since $\psi_{(3,2)}(P) \in CH(P)$, $\alpha(\psi_{(3,2)}) \leq 1$. For any $0 < a < 1$, there is a configuration P of type L such that $\boldsymbol{p}_1 > \boldsymbol{p}_3$, $\boldsymbol{p}_2 = (0,0)$ (in Z_0), $dist(\boldsymbol{p}_1, \boldsymbol{p}_3) = 1$, and $dist(\boldsymbol{p}_1, \boldsymbol{p}_2) = a$ hold. Then $dist(\boldsymbol{p}_1, g(P)) = (1+a)/3$, and $\alpha(\psi_{(3,2)}, P) = 1 - 3a/(2(1+a))$. By definition, $\alpha(\psi_{(3,2)}) \geq \lim_{a\to 0}(1 - 3a/(2(1+a))) = 1$. Thus $\alpha(\psi_{(3,2)}) = 1$.

Let $\mathcal{E} : P_0, P_1, \ldots$ be any execution starting from any initial configuration P_0. We show that each robot converges to one of at most two convergence points.

Let $Q_t^{(i)}$ be the multiset of the robots' positions that r_i identifies in Look phase in Z_i at time t. That is, $\gamma_i(Q_t^{(i)}) = P_t$, where γ_i is the coordinate transformation from Z_i to Z_0. Of course, $Q_t^{(i)}$ and P_t are similar, and have the same type. Keep in mind that r_i computes $\psi_{(3,2)}(Q_t^{(i)})$, not $\psi_{(3,2)}(P_t)$, and $Q_t^{(i)} \neq Q_t^{(j)}$ occurs in general.

If P_t (and $Q_t^{(i)}$) is type G, $\psi_{(3,2)}(Q_t^{(i)}) = (0,0)$ for all robot r_i, which means that r_i does not move, since $(0,0)$ is the current position of r_i in Z_i. Since $|\overline{P_t}| \leq 2$ by the definition of type G, and each robot r_i does not move after t (regardless of whether or not it is faulty), it converges to one of at most two convergence points.

Suppose that P_t (and $Q_t^{(i)}$) is type L. Without loss of generality, we assume $P_t = \{\boldsymbol{x}_t(r_1), \boldsymbol{x}_t(r_2), \boldsymbol{x}_t(r_3)\}$, where $\boldsymbol{x}_t(r_1)$, $\boldsymbol{x}_t(r_2)$, $\boldsymbol{x}_t(r_3)$ are distinct, and $\boldsymbol{x}_t(r_2) \in \overline{\boldsymbol{x}_t(r_1)\boldsymbol{x}_t(r_3)}$.

Suppose that $\gamma_i^{-1}(\boldsymbol{x}_t(r_j)) = \boldsymbol{y}_t^{(i)}(r_j)$, for all $i, j = 1, 2, 3$. That is, $Q_t^{(i)} = \{\boldsymbol{y}_t^{(i)}(r_1), \boldsymbol{y}_t^{(i)}(r_2), \boldsymbol{y}_t^{(i)}(r_3)\}$, where $\boldsymbol{y}_t^{(i)}(r_i) = (0,0)$.

The target position of r_1 (resp. r_3) is the middle point of $\boldsymbol{x}_t(r_1)$ (resp. $\boldsymbol{x}_t(r_3)$) and $\boldsymbol{x}_t(r_2)$, and that of r_2 is the middle point of $\boldsymbol{x}_t(r_2)$ and $\boldsymbol{x}_t(r_j)$, where j is either 1 or 3, and is determined from $Q_t^{(2)}$ (not from P_t), since r_2 computes $\psi_{(3,2)}(Q_t^{(2)})$ (not $\psi_{(3,2)}(P_t)$) in Compute phase; $j = 1$ if $\boldsymbol{y}_t^{(2)}(r_1) > \boldsymbol{y}_t^{(2)}(r_3)$, and otherwise, $j = 3$. (Since P_t is type L (but not G), $\boldsymbol{y}_t^{(2)}(r_1) = \boldsymbol{y}_t^{(2)}(r_3)$ does not occur.)

We observe the following: $P_{t'}$ is type L for all $t' \geq t$. Moreover, $\boldsymbol{x}_{t'}(r_2) \in \overline{\boldsymbol{x}_{t'}(r_1)\boldsymbol{x}_{t'}(r_3)}$, i.e., the relative positions of r_1, r_2, and r_3 do not change, unless some of them match and the type changes to G. Thus if $\boldsymbol{y}_t^{(2)}(r_1) > \boldsymbol{y}_t^{(2)}(r_3)$, then $\boldsymbol{y}_{t'}^{(2)}(r_1) > \boldsymbol{y}_{t'}^{(2)}(r_3)$ for all $t' > t$. Therefore, if the target position of r_2 at time t is the middle point of $\boldsymbol{x}_t(r_2)$ and $\boldsymbol{x}_t(r_j)$, so is at time t' for all $t' > t$. Thus r_2 converges either to r_1 or to r_3, as long as r_2 is not faulty.

If r_2 is faulty, either r_1 or r_3 is not faulty, and it converges to r_2.

Suppose that P_t is type T. For $i = 1, 2, 3$, let $Q_t^{(i)} = \{\boldsymbol{q}_1^{(i)}, \boldsymbol{q}_2^{(i)}, \boldsymbol{q}_3^{(i)}\}$, where we assume that $\boldsymbol{q}_1^{(i)}$, $\boldsymbol{q}_2^{(i)}$, $\boldsymbol{q}_3^{(i)}$ appear in this order counter-clockwise on the boundary of the triangle.[10] Since Z_i is right-handed, once activated, a robot r_i recognizes j such that $\boldsymbol{q}_j^{(i)} = (0, 0)$, and moves to the middle point of the current position $(0, 0)$ and the next point $\boldsymbol{q}_{j+1}^{(i)}$ in Z_i, which is, in Z_0, the middle point of the current position of r_i and the next point counter-clockwise in P_t. Thus $P_{t'}$ is type T for all $t' \geq t$.

Now, it is obvious that (1) if there is no faulty robot, then all robots converge to a point, (2) if exactly one robot crashes at a point $\boldsymbol{p}$, then the other two robots converge to $\boldsymbol{p}$, (3) if exactly two robots crash at points $\boldsymbol{p}$ and $\boldsymbol{p}'$, then the third robot converges either to $\boldsymbol{p}$ or $\boldsymbol{p}'$. □

7.2.2. Algorithm $\psi_{(n,2)}$

We propose an algorithm $\psi_{(n,2)}$ to solve the FC(2)-PO for $n \geq 4$, and show its correctness. By combining $\psi_{(3,2)}$ and $\psi_{(n,2)}$, we obtain an algorithm for the FC(2)-PO.

Let $\mathrm{LN}_{(n,2)}$ (for $n \geq 4$) be an algorithm to solve the FC(2)-PO in such a way that $CH(P_t) \subseteq CH(P_0)$ for all $t \geq 0$, provided that $CH(P_0)$ is a line segment. Assuming the existence of $\mathrm{LN}_{(n,2)}$, we propose $\psi_{(n,2)}$ and show its correctness. Later, in Section 7.2.3, we propose $\mathrm{LN}_{(n,2)}$ and show its correctness.

Rotation group, view, and order.. We first prepare some notions necessary to describe $\psi_{(n,2)}$.

In $\psi_{(3,2)}$, we used lexicographic order $>$ to compare positions $\boldsymbol{p}$ and $\boldsymbol{q}$. Let $\sqsubset$ be a lexicographic order on $\mathcal{P}$ defined as follows: For distinct multisets of n points $P = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_n\}$ and $Q = \{\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_n\}$, where for all $i = 1, 2, \ldots, n-1$, $\boldsymbol{p}_i \leq \boldsymbol{p}_{i+1}$ and $\boldsymbol{q}_i \leq \boldsymbol{q}_{i+1}$ hold, $P \sqsubset Q$, if and only if there is an $i (1 \leq i \leq n-1)$ such that (i) $\boldsymbol{p}_j = \boldsymbol{q}_j$ for all $j = 1, 2, \ldots, i-1$,[11] and (ii) $\boldsymbol{p}_i < \boldsymbol{q}_i$.

Although these orders are easy to understand, they have a drawback for our purpose, since robots cannot consistently compute lexicographic orders $<$ and $\sqsubset$. To see this fact, let $\boldsymbol{x}$ and $\boldsymbol{y}$ be distinct points in $\overline{P}$ in Z_0. Then both $\gamma_i^{-1}(\boldsymbol{x}) < \gamma_i^{-1}(\boldsymbol{y})$ and $\gamma_i^{-1}(\boldsymbol{x}) > \gamma_i^{-1}(\boldsymbol{y})$ can occur, depending on Z_i. Thus robots cannot consistently compare $\boldsymbol{x}$ and $\boldsymbol{y}$ using $>$. And it is true for $\sqsubset$, as

[10] $\boldsymbol{q}_j^{(i)}$ may not be the position $\boldsymbol{y}_t^{(i)}(r_j)$ of robot r_j at time t observed by r_i in Z_i.
[11] We assume $\boldsymbol{p}_0 = \boldsymbol{q}_0$.

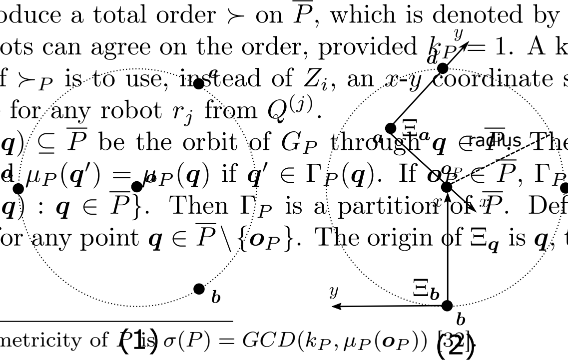


Figure 3: (1) A configuration P, where $\overline{P} = \{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{o}\}$. If $\mu_P(\boldsymbol{a}) = \mu_P(\boldsymbol{b}) = \mu_P(\boldsymbol{c}) = i$ for an integer $i > 0$, then $k_P = 3$, regardless of $\mu_P(\boldsymbol{o})$. (2) A configuration P, where $\overline{P} = \{\boldsymbol{o}_P, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{d}\}$. In Z_0, $\boldsymbol{o}_P = (0,0)$, $\boldsymbol{a} = (-1/2, 1/2)$, $\boldsymbol{b} = (0,-1)$, $\boldsymbol{c} = (1,0)$, and $\boldsymbol{d} = (0,1)$. Then the center of the smallest enclosing circle C is $\boldsymbol{o}_P$, and its radius is 1. Solid arrows represent the x- and y-axes of $\Xi_{\boldsymbol{a}}$ and $\Xi_{\boldsymbol{b}}$, whose lengths are 1, i.e., the radius of C. In $\Xi_{\boldsymbol{b}}$, $\boldsymbol{o}_P$, $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$, and $\boldsymbol{d}$ are $(1,0)$, $(3/2,1/2)$, $(0,0)$, $(0,-1)$, and $(2,0)$, respectively, and thus $\gamma_{\boldsymbol{b}}^{-1}(P) = V_P(\boldsymbol{b}) = \{(1,0), (3/2,1/2), (0,0), (0,-1), (2,0)\}$.

well. In $\psi_{(n,2)}$, we introduce and use an order $\succ$ that all robots can consistently compute.

Let $P = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_n\}$ be a multiset of n points, and $\overline{P} = \{\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_m\}$ be the set of distinct points in P. We denote the multiplicity of $\boldsymbol{q}$ in P by $\mu_P(\boldsymbol{q})$, i.e., $\mu_P(\boldsymbol{q}) = |\{i : \boldsymbol{p}_i = \boldsymbol{q} \in P\}|$. We identify P with a pair $(\overline{P}, \mu_P)$, where μ_P is a labeling function to associate label $\mu_P(\boldsymbol{q})$ with each element $\boldsymbol{q} \in \overline{P}$. Let G_P be the rotation group $G_{\overline{P}}$ of $\overline{P}$ about $\boldsymbol{o}_P$ preserving μ_P, where $\boldsymbol{o}_P$ is the center of the smallest enclosing circle of P. The order $|G_P|$ of G_P is denoted by k_P. We define $k_P = 0$, if $|\overline{P}| = 1$, i.e., if $\overline{P} = \{\boldsymbol{o}_P\}$.[12]

For example, let $P_1 = \{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}\}$, $P_2 = \{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{c}\}$, $P_3 = \{\boldsymbol{a}, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{c}\}$, and $P_4 = \{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{o}, \boldsymbol{o}\}$, where a triangle $\boldsymbol{abc}$ is equilateral, and $\boldsymbol{o}$ is the center of the smallest enclosing circle of triangle $\boldsymbol{abc}$. Then $k_{P_1} = k_{P_3} = k_{P_4} = 3$ and $k_{P_2} = 1$. (Figure 3(1) illustrates an example similar to P_4.)

Suppose that P is a configuration in Z_0. When activated, a robot r_i identifies the robots' positions $Q^{(i)} = \gamma_i^{-1}(P)$ in Z_i in Look phase. Since P and $Q^{(i)}$ are similar, $k_P = k_{Q^{(i)}}$, i.e., all robots can consistently compute k_P.

We introduce a total order $\succ$ on $\overline{P}$, which is denoted by $\succ_P$, in such a way that all robots can agree on the order, provided $k_P = 1$. A key idea behind the definition of $\succ_P$ is to use, instead of Z_i, an x-y coordinate system Ξ_i which is computable for any robot r_j from $Q^{(j)}$.

Let $\Gamma_P(\boldsymbol{q}) \subseteq \overline{P}$ be the orbit of G_P through $\boldsymbol{q} \in \overline{P}$. Then $|\Gamma_P(\boldsymbol{q})| = k_P$ if $\boldsymbol{q} \neq \boldsymbol{o}_P$, and $\mu_P(\boldsymbol{q}') = \mu_P(\boldsymbol{q})$ if $\boldsymbol{q}' \in \Gamma_P(\boldsymbol{q})$. If $\boldsymbol{o}_P \in \overline{P}$, $\Gamma_P(\boldsymbol{o}_P) = \{\boldsymbol{o}_P\}$. Let $\Gamma_P = \{\Gamma_P(\boldsymbol{q}) : \boldsymbol{q} \in \overline{P}\}$. Then Γ_P is a partition of $\overline{P}$. Define x-y coordinate system $\Xi_{\boldsymbol{q}}$ for any point $\boldsymbol{q} \in \overline{P} \setminus \{\boldsymbol{o}_P\}$. The origin of $\Xi_{\boldsymbol{q}}$ is $\boldsymbol{q}$, the unit distance is

[12] The symmetricity of P is $\sigma(P) = GCD(k_P, \mu_P(\boldsymbol{o}_P))$ [32].

the radius of the smallest enclosing circle of P, the x-axis is taken so that it goes through $\boldsymbol{o}_P$, and it is right-handed. Let $\gamma_{\boldsymbol{q}}$ be the coordinate transformation from $\Xi_{\boldsymbol{q}}$ to Z_0. Then the view $V_P(\boldsymbol{q})$ of $\boldsymbol{q}$ is defined to be $\gamma_{\boldsymbol{q}}^{-1}(P)$. Obviously $V_P(\boldsymbol{q}') = V_P(\boldsymbol{q})$ (as multisets), if and only if $\boldsymbol{q}' \in \Gamma_P(\boldsymbol{q})$. Let $View_P = \{V_P(\boldsymbol{q}) : \boldsymbol{q} \in \overline{P} \setminus \{\boldsymbol{o}_P\}\}$. (See Figure 3(2) for an example.)

Any robot r_i, in Compute phase from $Q^{(i)}$, can compute $\Xi_{\boldsymbol{q}}$ and $V_{Q^{(i)}}(\boldsymbol{q})$ for each $\boldsymbol{q} \in \overline{Q^{(i)}} \setminus \{\boldsymbol{o}_{Q^{(i)}}\}$, and thus $View_{Q^{(i)}}$. Since P and $Q^{(i)}$ are similar, by the definition of $\Xi_{\boldsymbol{q}}$, $View_P = View_{Q^{(i)}}$, which implies that all robots r_i can consistently compute $View_P$.

We define $\succ_P$ on Γ_P using $View_P$. For any distinct orbits $\Gamma_P(\boldsymbol{q})$ and $\Gamma_P(\boldsymbol{q}')$, $\Gamma_P(\boldsymbol{q}) \succ_P \Gamma_P(\boldsymbol{q}')$, if and only if one of the following conditions hold:

1. $\mu_P(\boldsymbol{q}) > \mu_P(\boldsymbol{q}')$.
2. $\mu_P(\boldsymbol{q}) = \mu_P(\boldsymbol{q}')$ and $dist(\boldsymbol{q}, \boldsymbol{o}_P) < dist(\boldsymbol{q}', \boldsymbol{o}_P)$ hold, where $dist(\boldsymbol{x}, \boldsymbol{y})$ is the Euclidean distance between $\boldsymbol{x}$ and $\boldsymbol{y}$.
3. $\mu_P(\boldsymbol{q}) = \mu_P(\boldsymbol{q}')$, $dist(\boldsymbol{q}, \boldsymbol{o}_P) = dist(\boldsymbol{q}', \boldsymbol{o}_P)$, and $V_P(\boldsymbol{q}) \sqsupset V_P(\boldsymbol{q}')$ hold. [13]

Then $\succ_P$ is a total order on Γ_P. If $k_P = 1$, since $\Gamma_P(\boldsymbol{q}) = \{\boldsymbol{q}\}$ for all $\boldsymbol{q} \in \overline{P}$, we regard $\succ_P$ as a total order on $\overline{P}$ by identifying $\Gamma_P(\boldsymbol{q})$ with $\boldsymbol{q}$. For a configuration P (in Z_0), from $Q^{(i)}$ (in Z_i), each robot r_i can consistently compute $k_P = k_{Q^{(i)}}$, $\Gamma_P = \Gamma_{Q^{(i)}}$, and $View_P = View_{Q^{(i)}}$, and hence $\succ_P = \succ_{Q^{(i)}}$. Thus, all robots can agree on, e.g., the largest point $\boldsymbol{q} \in \overline{P}$ with respect to $\succ_P$.

Configuration type.. Since $k_P = 0$ implies $m_P = 1$, i.e., the FC(2)-PO has been solved, we may assume $k_P \geq 1$ in what follows. We partition the set of all multisets $P = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_n\}$ for all $n \geq 4$ into six types G, L, T, I, S, and Z. Let $m_P = |\overline{P}|$.

G(oal): $m_P \leq 2$.

L(ine): $CH(P)$ is a line segment.

T(riangle): $m_P = 3$ and $CH(P)$ is a triangle.

I(nside): $m_P = 4$, $CH(P)$ is a triangle, and $\boldsymbol{o}_P \in P$.

S(ide): $m_P = 4$, $CH(P)$ is a triangle, and $\boldsymbol{M}_P \in P$, where $\boldsymbol{M}_P$ is the middle point of a longest side of $CH(P)$.

Z: P does not belong to the above five types.

Target function $\psi_{(n,2)}$.. Now we define target function $\psi_{(n,2)}$. Let $\mathcal{P}$ be the set of multisets that contains at least one $(0,0)$, which is the domain of a target function $\psi_{(n,2)}$. Algorithm $\mathrm{LN}_{(n,2)}$, which we assume to exist, transforms any configuration of type L into a configuration of type G, without going through a configuration whose type is not L.

[Target function $\psi_{(n,2)}$]

[13]Since $dist(\boldsymbol{o}_P, \boldsymbol{o}_P) = 0$, $V_P(\boldsymbol{q})$ is not compared with $V_P(\boldsymbol{o}_P)$ with respect to $\sqsupset$.

1. When P is type Z:
   (a) If $k_P \geq 2$, $\psi_{(n,2)}(P) = \boldsymbol{o}_P$.
   (b) If $k_P = 1$, $\psi_{(n,2)}(P) = \boldsymbol{a}_P$, where $\boldsymbol{a}_P \in \overline{P}$ is the largest point with respect to $\succ_P$, which is well-defined since $k_P = 1$.
2. If P is type L, invoke $\mathrm{LN}_{(n,2)}$.
3. When P is type T, let $\overline{P} = \{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}\}$.
   (a) If triangle $\boldsymbol{abc}$ is equilateral, $\psi_{(n,2)}(P) = \boldsymbol{o}_P$.
   (b) If triangle $\boldsymbol{abc}$ is not equilateral, $\psi_{(n,2)}(P) = \boldsymbol{M}_P$, where $\boldsymbol{M}_P$ is the middle point of the longest side. If there are two longest sides, $\boldsymbol{M}_P$ is the middle point of the side next to the shortest side counter-clockwise.
4. If P is type I, $\psi_{(n,2)}(P) = \boldsymbol{o}_P$.
5. If P is type S, $\psi_{(n,2)}(P) = \boldsymbol{M}_P$ (which is defined in the definition of type S).

Correctness.. To show that target function $\psi_{(n,2)}$ is an algorithm to solve the FC(2)-PO for $n \geq 4$, provided the existence of $\mathrm{LN}_{(n,2)}$, we need the following technical lemma. A multiset P is said to be *linear* if $CH(P)$ is a line segment.

Lemma 4. *Let A be a* **set** *(not a multiset) of points satisfying (1) A is not linear, (2) $k_A \geq 2$, and (3) $\boldsymbol{o}_A \notin A$. For any point $\boldsymbol{a} \in A$, let $B = (A \cup \{\boldsymbol{o}_A\}) \setminus \{\boldsymbol{a}\}$, i.e., B is constructed from A by replacing $\boldsymbol{a} \in A$ with $\boldsymbol{o}_A$. Then $k_B = 1$.*

Proof. See Appendix A. □

Based on the above lemma, we can show the correctness of $\psi_{(n,2)}$ provided the existence of $\mathrm{LN}_{(n,2)}$.

Lemma 5. *Target function $\psi_{(n,2)}$ is an algorithm to solve the* FC(2)-PO *for $n \geq 4$, provided the existence of $LN_{(n,2)}$.*

Proof. See Appendix B. □

7.2.3. Algorithm $LN_{(n,2)}$

We propose the target function $\mathrm{LN}_{(n,2)}$ and show the following: $\mathrm{LN}_{(n,2)}$ is an FC(2)-PO algorithm for any initial configuration P_0 such that CH_0 is a line segment, where $CH_t = CH(P_t)$. Moreover, $CH_t \subseteq CH_0$ holds for all $t \geq 0$. We borrow some symbols and notations from the last section.

Let $P = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_n\} \in \mathcal{P}$ be a configuration of type L, which may be a configuration that a robot identifies in Look phase. We identify a point $\boldsymbol{p}_i$ in R^2 with a point in R: Since $(0,0) \in P$, we rotate P about $(0,0)$ counter-clockwise so that the resultant P becomes the multiset of points in the x-axis. Then we denote $(p,0)$ by p. In what follows in this section, a configuration P is thus regarded as a multiset of n real numbers, including at least one 0. We assume $p_1 \leq p_2 \leq \cdots \leq p_n$. By $\overline{P} = \{b_1, b_2, \ldots, b_{m_P}\}$, we denote the set of distinct real numbers in P, where m_P is the size $|\overline{P}|$ of $\overline{P}$, and $b_1 < b_2 < \cdots < b_{m_P}$. The length of $CH(P)$ is denoted by $L_P = b_{m_P} - b_1 = p_n - p_1$. Let $\lambda_P = \max_{p \in P} \min\{p - p_1, p_{m_P} - p\} \leq L_P/2$. Define j^* by $b_{j^*} = 0$. (Thus the current position of a robot r_i who identifies P in Look phase is b_{j^*} in Z_i.) Since P is type L, $k_P \leq 2$. We denote the middle point of x and y by M_{xy}, i.e., $M_{xy} = (x+y)/2$.

Like $\psi_{(n,2)}$, we consider ten types, which we define as follows:

G: $m_P \leq 2$.

B$_3$: $m_P = 3$ and $k_P = 2$.

B$_4$: $m_P = 4$ and $k_P = 2$.

B$_5$: $m_P = 5$ and $k_P = 2$.

B$_6$: $m_P = 6$ and $k_P = 2$.

B: $m_P \geq 7$ and $k_P = 2$.

U$_3$: $m_P = 3$ and $k_P = 1$.

W: $m_P = 4$, $k_P = 1$, and $\overline{P} = \{b_1, b_2, b_3, b_4\}(b_1 < b_2 < b_3 < b_4)$ satisfies either (a) $2(b_2 - b_1) = b_3 - b_2$ and $b_3 \leq M_{b_1 b_4}$, or (b) $2(b_4 - b_3) = b_3 - b_2$ and $b_2 \geq M_{b_1 b_4}$.

U$_4$: $m_P = 4$, $k_P = 1$, and P is not type W.

U: $m_P \geq 5$ and $k_P = 1$.

We now give target function $\mathrm{LN}_{(n,2)}$.

[Target function LN$_{(n,2)}$]

1. If P is type G, $\mathrm{LN}_{(n,2)}(P) = 0$.
2. When P is type B: If $j^* \leq \lceil m_P/2 \rceil$, $\mathrm{LN}_{(n,2)}(P) = b_1$. Otherwise if $j^* > \lceil m_P/2 \rceil$, $\mathrm{LN}_{(n,2)}(P) = b_{m_P}$.
3. When P is type B$_3$: $m_P = 3$. If $j^* \leq 2$, $\mathrm{LN}_{(n,2)}(P) = M_{b_1 b_2}$. Otherwise if $j^* = 3$, $\mathrm{LN}_{(n,2)}(P) = M_{b_2 b_3}$.
4. When P is type B$_4$: $m_P = 4$. If $j^* \leq 2$, $\mathrm{LN}_{(n,2)}(P) = M_{b_1 b_2}$. Otherwise if $j^* \geq 3$, $\mathrm{LN}_{(n,2)}(P) = M_{b_3 b_4}$.
5. When P is type B$_5$: $m_P = 5$. If $j^* \leq 3$, $\mathrm{LN}_{(n,2)}(P) = b_2$. Otherwise if $j^* \geq 4$, $\mathrm{LN}_{(n,2)}(P) = b_4$.
6. When P is type B$_6$: $m_P = 6$. If $j^* \leq 3$, $\mathrm{LN}_{(n,2)}(P) = b_2$. Otherwise if $j^* \geq 4$, $\mathrm{LN}_{(n,2)}(P) = b_5$.
7. When P is type U: Since $k_P = 1$, either $b_1 \succ_P b_{m_P}$ or $b_{m_P} \succ_P b_1$ holds. If $b_1 \succ_P b_{m_P}$, then $\mathrm{LN}_{(n,2)}(P) = b_1$. Otherwise if $b_{m_P} \succ_P b_1$, $\mathrm{LN}_{(n,2)}(P) = b_{m_P}$.

8. When P is type U_3: Since $k_P = 1$ and $m_P = 3$, if $b_2 = M_{b_1 b_3}$, then $\mu_P(b_1) \neq \mu_P(b_3)$. If $b_2 < M_{b_1 b_3}$ or $(b_2 = M_{b_1 b_3}) \wedge (\mu_P(b_1) > \mu_P(b_3))$, then $\mathrm{LN}_{(n,2)}(P) = (2b_1 + b_2)/3$. Otherwise, if $b_2 > M_{b_1 b_3}$ or $(b_2 = M_{b_1 b_3}) \wedge (\mu_P(b_1) < \mu_P(b_3))$, then $\mathrm{LN}_{(n,2)}(P) = (b_2 + 2b_3)/3$.
9. When P is type W: $k_P = 1$ and $m_P = 4$, and P satisfies either condition (a) or (b) (of the definition of type W).
   (a) If $2(b_2 - b_1) = b_3 - b_2$ and $b_3 \leq M_{b_1 b_4}$, then $\mathrm{LN}_{(n,2)}(P) = b_2$.
   (b) If $2(b_4 - b_3) = b_3 - b_2$ and $b_2 \geq M_{b_1 b_4}$, then $\mathrm{LN}_{(n,2)}(P) = b_3$.
10. When P is type U_4: $k_P = 1$, $m_P = 4$, and P is not type W. Suppose that $\mu_P(b_1) \geq \mu_P(b_4)$ holds. (The case P satisfies $\mu_P(b_1) < \mu_P(b_4)$ is symmetric, and we omit it.)
    (a) If $\mu_P(b_1) \geq \mu_P(b_3)$, then $\mathrm{LN}_{(n,2)}(P) = b_1$.
    (b) If $(\mu_P(b_1) < \mu_P(b_3)) \wedge (\mu_P(b_3) \geq 3)$, $\mathrm{LN}_{(n,2)}(P) = b_1$, if $b_3 = 0$, and $\mathrm{LN}_{(n,2)}(P) = 0$, otherwise if $b_3 \neq 0$.
    (c) Otherwise if $(\mu_P(b_1) < \mu_P(b_3)) \wedge (\mu_P(b_3) < 3)$, $\mu_P(b_1) = \mu_P(b_4) = 1$ and $\mu_P(b_3) = 2$. $\mathrm{LN}_{(n,2)}(P) = b_1$, if $(b_2 = 0) \vee (b_3 = 0)$, and $\mathrm{LN}_{(n,2)}(P) = 0$, otherwise if $(b_1 = 0) \vee (b_4 = 0)$.

We can show the following lemma.

Lemma 6. *Target function $LN_{(n,2)}$ is an algorithm to solve the* FC(2)-PO *for any configuration of type L, without reaching a configuration not in type L.*

Proof. See Appendix Appendix C. □

It is easy to see that $\alpha(\psi_{(n,2)}) = 1$. By Lemmas 5 and 6, we have the following theorem.

Theorem 11. *Target function $\psi_{(n,2)}$, which satisfies $\alpha(\psi_{(n,2)}) = 1$, is an algorithm for the* FC(2)-PO.

8. Gathering problem

We finally investigate the gathering problem, provided that there are no faulty robots, to emphasize that the gathering and the convergence problems have completely different properties from the viewpoint of compatibility.

Since the gathering problem is not solvable if $n = 2$ [32], we assume $n \geq 3$ in this section. Moreover, we assume that the robots initially occupy distinct points, as all proposed gathering algorithms assume. There are many gathering algorithms. The following algorithm GAT [32] is one of them.

Let $P = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_n\}$ be a multiset of $n \geq 3$ points. We use concepts $\overline{P}, \mu_P, \boldsymbol{o}_P, k_P$, and $\succ_P$ introduced in Section 7.2.2.

[Target function GAT]

1. If there is a unique $\boldsymbol{p} \in P$ such that $\mu_P(\boldsymbol{p}) > 1$, $\mathrm{GAT}(P) = \boldsymbol{p}$.

2. Otherwise, if $\mu_P(\boldsymbol{p}) = 1$ for all $\boldsymbol{p} \in P$:

   (a) If $k_P \leq 1$, $\mathrm{GAT}(P) = \boldsymbol{p}$, where $\boldsymbol{p}$ is the largest point in P with respect to $\succ_P$.

   (b) If $k_P > 1$, $\mathrm{GAT}(P) = \boldsymbol{o}_P$.

Observe that $\alpha(\mathrm{GAT}) = 1$. A sketch of the correctness proof of GAT is as follows: Since the robots initially occupy distinct points by assumption, and by the definition of Step 2 of GAT, in any execution of GAT, there must be a time such that a unique point $\boldsymbol{q}$[14] satisfying $\mu(\boldsymbol{q}) > 1$ occurs in the configuration for the first time. Then by the definition of Step 1, $\mu(\boldsymbol{q})$ monotonically increases (while $\mu(\boldsymbol{x})$ of the other points $\boldsymbol{x} \in P$ monotonically decreases), and eventually $\mu(\boldsymbol{q}) = n$ is satisfied.

We can modify Step 2(a) of GAT to obtain another algorithm GAT$'$. For example, $\mathrm{GAT}'(P)$ can be the smallest point $\boldsymbol{p}'$ in P with respect to $\succ_P$, instead of the largest point $\boldsymbol{p}$. Then indeed GAT$'$ is also a gathering algorithm with $\alpha(\mathrm{GAT}') = 1$, but obviously $\Phi = \{\mathrm{GAT}, \mathrm{GAT}'\}$ is not compatible with respect to the gathering problem; if some robots take GAT and some others GAT$'$, then a configuration P such that $\mu_P(\boldsymbol{p}), \mu_P(\boldsymbol{p}') \geq 2$ may yield. Let us summarize.

Theorem 12. [32] *Let $\Phi = \{\mathrm{GAT}\}$ and $\Phi' = \{\mathrm{GAT}'\}$. Then Φ and Φ' are compatible with respect to the gathering problem, but $\Phi \cup \Phi'$ is not.*

Here, $\alpha(\Phi) = \alpha(\Phi') = \alpha(\Phi \cup \Phi') = 1$.

Theorem 13. *Any target function ϕ is not a gathering algorithm if $\alpha(\phi) < 1$, or equivalently, any set Φ of target functions such that $\alpha(\Phi) < 1$ is not compatible with respect to the gathering problem.*

Proof. Consider any gathering algorithm ϕ and show that $\alpha(\phi) = 1$.

Suppose that $n = 3$. For any initial configuration P_0 satisfying that all robots occupy distinct positions, we investigate any execution $\mathcal{E} : P_0, P_1, \ldots$, assuming that the scheduler is central, i.e., exactly one robot is activated each time.

Since $\mathcal{E}$ achieves the gathering, there is a time instant t such that $|\overline{P_t}| = 2$ and $|\overline{P_{t+1}}| = 1$. Since exactly one robot, say r, is activated at t, it moves to the position of the other robots (since they occupy the same position). Hence $\alpha(\phi) = 1$. □

9. Conclusions

We introduced the concept of compatibility and investigated the compatibilities of several convergence problems. A compatible set Φ of target functions with respect to a problem Π is an extension of an algorithm ϕ for Π, in the

[14] Here, $\boldsymbol{q}$ is either $\boldsymbol{p}$ or $\boldsymbol{o}_P$ in Step 2 of GAT.

sense that every target function $\phi \in \Phi$ is an algorithm for Φ, although a set of algorithms for Π is not always a compatible set with respect to Π.

The problems we investigated are the convergence problem, the fault tolerant (n, f)-convergence problem (FC(f)), the fault tolerant (n, f)-convergence problem to a convex f-gon (FC(f)-CP), and the fault tolerant (n, f)-convergence problem to f points (FC(f)-PO), for crash faults. The gathering problem was also investigated. The results are summarized in Table 1 and Figure 1. Main observations we would like to emphasize are:

1. The convergence problem, the FC(1), the FC(1)-PO, and the FC(f)-CP share the same property: Every set Φ of target functions such that $0 \le \alpha(\Phi) < 1$ is compatible.
2. The gathering problem and the FC(f)-PO for $f \ge 2$ share the same property: Any set Φ of target functions such that $0 \le \alpha(\Phi) < 1$ is **not** compatible.
3. FC(f) ($f \ge 2$) is in between the FC(f)-CP and the FC(f)-PO.
4. The FC(1)-PO and the FC(2)-PO are completely different problems from the viewpoint of compatibility.

A compatible set Φ with respect to a problem Π could be regarded as an "algorithm scheme" $\mathcal{S}$ for Π such that every algorithm in Φ is an instantiation of $\mathcal{S}$. For example, for any fixed $0 \le \alpha < 1$, the set $\Phi_\alpha = \{\phi : \alpha(\phi) \le \alpha\}$ of target functions ϕ is compatible with respect to the convergence problem. Then an algorithm scheme $\mathcal{S}$ defining Φ_α (as the set of its instantiations) is

$$\phi(P) \in \alpha * CH(P) \text{ for all } P \in \mathcal{P}.$$

It is not an algorithm, since it does not specify a concrete value of $\phi(P)$. It however captures the essence how all algorithms in common solve the convergence problem, and we can show the correctness of each algorithm from this description.

For any $0 < \delta \le 1/2$, the set $\Lambda_\delta = \{\phi : \phi \text{ is } \delta\text{-inner}\}$ is compatible with respect to the convergence problem [15]. For any $0 \le \alpha < 1$, there is a $0 < \delta \le 1/2$ such that $\Phi_\alpha \subset \Lambda_\delta$, which implies that there is another algorithm scheme for the convergence problem which is more general than algorithm scheme: $\phi(P) \in \alpha * CH(P)$ for all $P \in \mathcal{P}$. Investigation of an algorithm scheme would lead us a deeper understanding of robot algorithms.

Note that the concept of algorithm scheme is not new: In Chapter 21 of [16], for example, the authors first present GENERIC-MST, which is an algorithm scheme to construct a minimum spanning tree, show its correctness, and then derive Kruscal's and Prim's algorithms as its instantiations.

Before closing the paper, we list some open problems:

1. Extend Table 1 to contain the results for $\alpha(\Phi) > 1$.
2. Suppose that ϕ and ϕ' are algorithms for the convergence problem. Find a necessary and/or a sufficient condition for $\Phi = \{\phi, \phi'\}$ to be compatible with respect to the convergence problem.

More generally, if a set Φ of target functions is compatible with respect to the convergence problem, then every target function $\phi \in \Phi$ is a convergence algorithm. What is a sufficient condition for a set of convergence algorithms to be compatible with respect to the convergence algorithm?

3. Is there an FC(3)-PO algorithm?
   More generally, is there an algorithm for the FC(f)-PO for any $3 \leq f \leq n-1$? We conjecture that there is an FC(f)-PO algorithm.
4. Is the next statement correct? If a set Φ of target functions is compatible with respect to the convergence problem, so is $\beta\Phi = \{\beta\phi : \phi \in \Phi\}$ for any real number $0 < \beta \leq 1$.
5. Let $ALG_{\mathrm{FC}(1)}$ (resp. $ALG_{\mathrm{FC}(1)\text{-PO}}$) be the set of all algorithms for the FC(1) (resp. the FC(1)-PO). Does $ALG_{\mathrm{FC}(1)\text{-PO}} = ALG_{\mathrm{FC}(1)}$ hold?
6. Investigate the compatibility of convergence problems under the $\mathcal{ASYNC}$ model.
7. Investigate the compatibility of convergence problems in the presence of Byzantine failures.
8. Investigate the compatibility of fault tolerant gathering problems.
9. Find interesting problems with a large compatible set.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Numbers JP22K11915 and JP24K02902.

References

[1] N. Agmon and D. Peleg. Fault-tolerant gathering algorithms for autonomous mobile robots. In *Proc. 15th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1063–1071, 2004.

[2] H. Ando, Y. Oasa, I. Suzuki, and M. Yamashita. A distributed memoryless point convergence algorithm for mobile robots with limited visibility. *IEEE Trans. Robotics and Automation*, 15:818–828, 1999.

[3] Y. Asahiro, I. Suzuki, and M. Yamashita. Monotonic self-stabilization and its application to robust and adaptive pattern formation. *Theoretical Computer Science*, 934:21–46, 2022.

[4] Y. Asahiro and M. Yamashita. Compatibility of convergence algorithms for autonomous mobile robots (extended abstract). In *Proc. 30th International Colloquium on Structural Information and Communication Complexity (SIROCCO2023), Lecture Notes in Computer Science*, volume 13892, pages 149–164, 2023.

[5] Y. Asahiro and M. Yamashita. Minimum algorithm sizes for self-stabilizing gathering and related problems of autonomous mobile robots (extended abstract). In *Proc. 25th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS2023), Lecture Notes in Computer Science*, volume 14310, pages 312–327, 2023.

[6] T. Balch and R. Arkin. Behavior-based formation control for multi-robot teams. *IEEE Trans. Robotics and Automation*, 14:926–939, 1998.

[7] Z. Bouzid, S. Das, and S. Tixeuil. Gathering of mobile robots tolerating multiple crash faults. In *Proc. IEEE 33rd International Conference on Distributed Computing Systems*, pages 337–346, 2013.

[8] Z. Bouzid, M. G. Potop-Butucaru, and S. Tixeuil. Byzantine convergence in robot networks. In *Proc. 13th International Symposium on Stabilization, Safety, and Security of Distributed Systems*, pages 52–70, 2009.

[9] Z. Bouzid, M. G. Potop-Butucaru, and S. Tixeuil. Optimal byzantine-resilient convergence in uni-directional robot networks. *Theoretical Computer Science*, 411:3154–3168, 2010.

[10] K. Buchin, P. Flocchini, I. Kostitsyana, T. Peters, N. Santoro, and K. Wada. On the computational power of energy-constrained mobile robots: algorithms and cross-model analysis. In *Proc. 29th International Colloquium on Structural Information and Communication Complexity (SIROCCO2022), Lecture Notes in Computer Science*, volume 13298, pages 42–61, 2022.

[11] Y. U. Cao, A. S. Fukunaga, and A. B. Kahng. Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*, 4:7–23, 1997.

[12] M. Cieliebak, P. Flocchini, G. Prencipe, and N. Santoro. Distributed computing by mobile robots: gathering. *SIAM J. Computing*, 41:829–879, 2012.

[13] R. Cohen and D. Peleg. Convergence properties of the gravitational algorithm in asynchronous robot systems. *SIAM J. Computing*, 34:1516–1528, 2005.

[14] R. Cohen and D. Peleg. Convergence of autonomous mobile robots with inaccurate sensors and movements. *SIAM J. Computing*, 38:276–302, 2008.

[15] A. Cord-Landwehr, B. Degener, M. Fischer, M Hüllman, B. Kempkes, A. Klaas, P. Kling, S. Kurras, M. Märtens, F. Meyer auf der Heide, C. Raupach, K. Swierkot, D. Warner, C. Weddemann, and D. Wonisch. A new approach for analyzing convergence algorithms for mobile robots. In *Proc. 38th International Colloquimum on Automata, Languates, and Programming(ICALP2011), Part II, Lecture Notes in Computer Science*, volume 6756, pages 650–661, 2011.

[16] T. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms (4th edition)*. MIT Press, 2022.

[17] S. Das, P. Flocchini, N. Santoro, and M Yamashita. Forming sequences of geometric patterns with oblivious mobile robots. *Distributed Computing*, 28:131–145, 2015.

[18] X. Défago, M. G. Potop-Butucaru, J. Clément, M. Messika, and R. Raipin-Parvédy. Fault and byzantine tolerant self-stabilizing mobile robots gathering – feasibility study. arXiv: 1602.05546, 2016.

[19] X. Défago, M. G. Potop-Butucaru, and S. Tixeuil. Fault-tolerant mobile robots. In *Distributed Computing by Mobile Entities, Lecture Notes in Computer Science*, volume 11340, pages 234–251, 2019.

[20] P. Flocchini. Gathering. In *Distributed Computing by Mobile Entities, Lecture Notes in Computer Science*, volume 11340, pages 63–82, 2019.

[21] P. Flocchini, G. Prencipe, and N. Santoro. *Distributed Computing by Oblivious Mobile Robots, Synthesis Lectures on Distributed Computing Theory 10*. Morgan & Claypool Publishers, 2012.

[22] P. Flocchini, G. Prencipe, N. Santoro, and P. Widmayer. Hard tasks for weak robots: the role of common knowledge in pattern formation by autonomous mobile robots. In *Proc. 10th International Symposium on Algorithms and Computation(ISAAC1999), Lecture Notes in Computer Science*, volume 1741, pages 93–102, 1999.

[23] T. Izumi, S. Soussi, Y. Katayama, N. Inuzuka, X. Defago, K. Wada, and M. Yamashita. The gathering problem for two oblivious robots with unreliable compasses. *SIAM J. Computing*, 41:26–46, 2012.

[24] S. Kamei, A. Lamani, F. Ooshita, S. Tixeuil, and K. Wada. Gathering on rings for myopic asynchronous robots with lights. In *Proc. 23rd International Conference on Principles of Distributed Systems*, pages 27:1–27:17, 2019.

[25] B. Katreniak. Convergence with limited visibility by asynchronous mobile robots. In *Proc. 18th International Colloquium on Structural Information and Communication Complexity (SIROCCO2011), Lecture Notes in Computer Science*, volume 6786, pages 125–137, 2011.

[26] Y. Kawauchi, M. Inaba, and T. Fukuda. A principle of decision making of cellular robotic system (cebot). In *Proc. IEEE Conference on Robotics and Automation*, pages 833–838, 1993.

[27] S. Murata, H. Kurokawa, and S. Kokaji. Self-assembling machine. In *Proc. IEEE Conference on Robotics and Automation*, pages 441–448, 1994.

[28] L. E. Parker. Designing control laws for cooperative agent teams. In *Proc. IEEE Conference on Robotics and Automation*, pages 582–587, 1993.

[29] L. E. Parker and C. Touzet. Multi-robot learning in a cooperative observation task. *Distributed Autonomous Robotic Systems*, 4:391–401, 2000.

[30] G. Prencipe. Pattern formation. In *Distributed Computing by Mobile Entities, Lecture Notes in Computer Science*, volume 11340, pages 37–62, 2019.

[31] K. Sugihara and I. Suzuki. Distributed algorithms for formation of geometric patterns with many mobile robots. *J. Robotic Systems*, 13:127–139, 1996.

[32] I. Suzuki and M. Yamashita. Distributed anonymous mobile robots – formation and agreement problems. *SIAM J. Computing*, 28:1347–1363, 1999.

[33] G. Viglietta. Uniform circle formation. In *Distributed Computing by Mobile Entities, Lecture Notes in Computer Science*, volume 11340, pages 83–108, 2019.

[34] I. A. Wagner and A. M. Bruckstein. From ants to a(ge)nts. *Ann. Math. Artificial Intelligence*, 31:1–5, 1996.

[35] M. Yamashita and I. Suzuki. Characterizing geometric patterns formable by oblivious anonymous mobile robots. *Theoretical Computer Science*, 411:2433–2453, 2010.

[36] Y. Yamauchi, T. Uehara, S. Kijima, and M. Yamashita. Plane formation by synchronous mobile robots in the three-dimensional euclidean space. *J. ACM*, 64:1–4, 2017.

Appendix A. Proof of Lemma 4

For any finite set S, $m_S = |S|$ is the size of S, C_S is the smallest enclosing circle of S, whose center is $\boldsymbol{o}_S$ and radius is d_S, k_S is the order of the rotation group G_S of S around $\boldsymbol{o}_S$, and CH_S is the convex hull of S. We introduce a notation for the convenience of description. Let C be a circle, and $\boldsymbol{x}$ and $\boldsymbol{y}$ be points on C. By $C(\boldsymbol{x}, \boldsymbol{y})$ (resp. $C[\boldsymbol{x}, \boldsymbol{y}]$), we denote the arc of C from $\boldsymbol{x}$ to $\boldsymbol{y}$ counter-clockwise, excluding (resp. including) both ends $\boldsymbol{x}$ and $\boldsymbol{y}$.

Consider any finite set A which satisfies the following conditions:

1. A is not linear, i.e., CH_A is not a line segment,
2. $k_A \geq 2$, and
3. $\boldsymbol{o}_A \notin A$.

Since $k_A \geq 2$ and A is not linear, $m_A \geq 3$. For any point $\boldsymbol{a} \in A$, let $B = (A \cup \{\boldsymbol{o}_A\}) \setminus \{\boldsymbol{a}\}$. We show that $k_B = 1$. Proof is by contradiction. We assume $k_B \geq 2$ to derive a contradiction.

(I) We first show that $\boldsymbol{o}_A = \boldsymbol{o}_B$. Proof is by contradiction. We assume $\boldsymbol{o}_B \neq \boldsymbol{o}_A$ to derive a contradiction.

If $k_A \geq 4$, then $C_A = C_B$ and hence $\boldsymbol{o}_A = \boldsymbol{o}_B$. Thus, $2 \leq k_A \leq 3$, and the number of points in A on C_A ($\boldsymbol{a}$ must be one of them) is at most 3. Without loss of generality, we assume that $\boldsymbol{o}_A = (0, 0)$ and $\boldsymbol{o}_B = (-1, 0)$. Then the x-coordinate of $\boldsymbol{a}$ is positive, since $d_B \leq d_A$.

(A) First consider the case in which $k_A = 2$. We start with showing that $(2 \leq)$ $k_B \leq 3$. There are exactly two points $\boldsymbol{a}$ and $-\boldsymbol{a}$ on C_A, where $-\boldsymbol{a}$ is the opposite point of $\boldsymbol{a}$ about $\boldsymbol{o}_A$. Thus $-\boldsymbol{a}$ is on C_A and in A. Let $\boldsymbol{c} = (c_x, c_y)$ and $\boldsymbol{c}' = (c_x, -c_y)$ be the intersections of C_A and C_B, where $c_y \geq 0$ ($\boldsymbol{c} = \boldsymbol{c}'$, i.e., $c_y = 0$ may occur). Since no points in B are on $C_B(\boldsymbol{c}, \boldsymbol{c}')$, if $c_x > -1$ held, C_B would not be the smallest enclosing circle. Thus $c_x \leq -1$ holds.

We examine where a point in B occurs on C_B. Let $\boldsymbol{h}$ and $-\boldsymbol{h}$ be the intersections of C_B and the y-axis, i.e., $\boldsymbol{h} = (0, \sqrt{d_B^2 - 1})$ and $-\boldsymbol{h} = (0, -\sqrt{d_B^2 - 1})$. Since $\boldsymbol{o}_A \in B$, $d_B \geq 1$, and there indeed exist $\boldsymbol{h}$ and $-\boldsymbol{h}$ (which may be the same). First, there is a point in B on $C_B[\boldsymbol{h}, -\boldsymbol{h}]$, since otherwise, C_B is not the smallest enclosing circle of B. It cannot occur on $C_B(\boldsymbol{c}, \boldsymbol{c}')$, since $C_B(\boldsymbol{c}, \boldsymbol{c}')$ is located outside C_A. Furthermore, it does not occur either on $C_B(\boldsymbol{h}, \boldsymbol{c})$ or on $C_B(\boldsymbol{c}', -\boldsymbol{h})$. If a point $\boldsymbol{x} \in B$ occurred there, $d_B = dist(\boldsymbol{x}, \boldsymbol{o}_B) < dist(-\boldsymbol{x}, \boldsymbol{o}_B)$ and $-\boldsymbol{x} \in B$ would hold. Note that $\boldsymbol{x}$ and thus $-\boldsymbol{x}$ is not on C_A, and hence $-\boldsymbol{x}$ is not $\boldsymbol{a}$. Thus if $\boldsymbol{x} \in B$ is a point on $C_B[\boldsymbol{h}, -\boldsymbol{h}]$, it is either $\boldsymbol{c}$ or $\boldsymbol{c}'$.

If both of $\boldsymbol{c}$ and $\boldsymbol{c}'$ were in B, since they were also on C_A, $k_A = 2$, and $c_x \leq -1$, $\boldsymbol{o}_A = \boldsymbol{o}_B$ would hold. Thus exactly one point in B is on $C_B[\boldsymbol{h}, -\boldsymbol{h}]$, which immediately implies $k_B \leq 3$.

(A1) Consider the case in which $k_A = k_B = 2$. Suppose that $\boldsymbol{a} \neq (d_A, 0)$. For any point $\boldsymbol{p}_0 \in B$, define a sequence of points

$$\mathcal{X} : \boldsymbol{p}_0, \boldsymbol{q}_0, \boldsymbol{p}_1, \boldsymbol{q}_1, \ldots$$

as follows: For any $i \geq 0$, $\boldsymbol{q}_i$ is the opposite point of $\boldsymbol{p}_i$ about $\boldsymbol{o}_B$, and for any $i \geq 1$, $\boldsymbol{p}_i$ is the opposite point of $\boldsymbol{q}_{i-1}$ about $\boldsymbol{o}_A$. If $\boldsymbol{p}_i \neq \boldsymbol{a}$ and $\boldsymbol{q}_i \neq \boldsymbol{o}_A$ for all $i \geq 0$, then $\boldsymbol{p}_i, \boldsymbol{q}_i \in A \cap B$, since $k_A = k_B = 2$.

Consider an instance of $\mathcal{X}$ for $\boldsymbol{p}_0 = \boldsymbol{o}_A$. First $\boldsymbol{q}_i \neq \boldsymbol{o}_A$ by definition. Next $\boldsymbol{p}_i \neq \boldsymbol{a}$, since all $\boldsymbol{p}_i$ and $\boldsymbol{q}_i$ occur on the x-axis, and $\boldsymbol{a}$ is not on the x-axis. Thus $\mathcal{X}$ consists of an infinite number of distinct points. It is a contradiction, since there are only m_A points in A (and B).

Suppose otherwise that $\boldsymbol{a} = (d_A, 0)$. There is a point $\boldsymbol{p} = (p_x, p_y) \in A$ such that $p_x \geq 0$, $p_y \neq 0$, and it is not on C_A, since A is not linear and $k_A = 2$. Since $p_y \neq 0$, $\boldsymbol{p}$ is neither $\boldsymbol{o}_A$ nor $\boldsymbol{a}$, and thus $\boldsymbol{p} \in B$, as well. Consider another instance of $\mathcal{X}$ for $\boldsymbol{p}_0 = \boldsymbol{p}$. Let $\boldsymbol{p}_i = (p_x^i, p_y^i)$ and $\boldsymbol{q}_i = (q_x^i, q_y^i)$ for $i \geq 0$. By a simple induction, for all $i \geq 0$, $p_x^{i+1} > p_x^i \geq 0$, $p_y^i \neq 0$, $q_x^{i+1} < q_x^i < -1$, $q_y^i \neq 0$, and $\boldsymbol{p}_i, \boldsymbol{q}_i \in A \cap B$. It is a contradiction, since $\boldsymbol{p}_i \neq \boldsymbol{a}$ and $\boldsymbol{q}_i \neq \boldsymbol{o}_A$ for all $i \geq 0$; $\mathcal{X}$ consists of an infinite number of distinct points.

(A2) Consider the case in which $k_A = 2$ and $k_B = 3$. We again consider the sequence $\mathcal{X} : \boldsymbol{p}_0, \boldsymbol{q}_0, \boldsymbol{p}_1, \boldsymbol{q}_1, \ldots$ after making two modifications. Since $k_B = 3$, we have two candidates to determine $\boldsymbol{q}_i$ from $\boldsymbol{p}_i$. Let C be the circle with center $\boldsymbol{o}_B$ such that it contains $\boldsymbol{p}_i$. Let $\boldsymbol{x}$ and $\boldsymbol{x}'$ be two points on C such that they form an equilateral triangle with $\boldsymbol{p}_i$. Since $k_B = 3$, they both belong to B. We assume that $\boldsymbol{x}$ has a smaller x-coordinate than $\boldsymbol{x}'$ (in case of a tie, we assume that $\boldsymbol{x}$ has a smaller y-coordinate than $\boldsymbol{x}'$). Then we choose $\boldsymbol{x}$ as $\boldsymbol{q}_i$.

We consider two instances of $\mathcal{X}$. $\mathcal{X}_1$ starts with $\boldsymbol{p}_0 = \boldsymbol{o}_A$ and $\boldsymbol{q}_0 = (-3/2, \sqrt{3}/2)$, and $\mathcal{X}_2$ with $\boldsymbol{p}_0 = \boldsymbol{o}_A$ and $\boldsymbol{q}_0 = (-3/2, -\sqrt{3}/2)$. Then, $\boldsymbol{p}_i = (p_x^i, p_y^i)$ (resp. $\boldsymbol{q}_i = (q_x^i, q_y^i)$) in $\mathcal{X}_1$, if and only if $\boldsymbol{p}_i = (p_x^i, -p_y^i)$ (resp. $\boldsymbol{q}_i = (q_x^i, -q_y^i)$) in $\mathcal{X}_2$.

Consider two straight lines $\ell_1 : y = \sqrt{3}(x+1)$ and $\ell_2 : y = -\sqrt{3}(x+1)$. By a similar induction like (A1), we can show that for all $i \geq 1$, $p_x^{i+1} > p_x^i > 0$, $p_y^i \neq 0$, $q_x^{i+1} < q_x^i < -1$, $q_y^i \neq 0$, since $\boldsymbol{p}_i$ (resp. $\boldsymbol{q}_i$) is located to the right (resp. left) side of ℓ_1 and ℓ_2. Since $\boldsymbol{q}_i$ do not reach $\boldsymbol{o}_A$, $\mathcal{X}_1$ and $\mathcal{X}_2$ both create an infinite number of distinct points, if they do not reach $\boldsymbol{a}$.

By definition, $\boldsymbol{p}_i = (p_x^i, p_y^i)$ (resp. $\boldsymbol{q}_i = (q_x^i, q_y^i)$) in $\mathcal{X}_1$, if and only if $\boldsymbol{p}_i = (p_x^i, -p_y^i)$ (resp. $\boldsymbol{q}_i = (q_x^i, -q_y^i)$) in $\mathcal{X}_2$, and $p_y^i, q_y^i \neq 0$, for all $i \geq 1$. Thus at least one of them creates an infinite number of distinct points, no matter where $\boldsymbol{a}$ is. It is a contradiction.

(B) Next consider the case in which $k_A = 3$. First we show that $k_B \leq 2$. Let $\boldsymbol{p}, \boldsymbol{x}$, and $\boldsymbol{x}'$ be the three points on C_A, which form an equilateral triangle. Recall that $\boldsymbol{c}$ and $\boldsymbol{c}'$ are intersections of C_A and C_B. By a similar argument in (A), one of $\boldsymbol{x}$ or $\boldsymbol{x}'$ is either $\boldsymbol{c}$ or $\boldsymbol{c}'$. Without loss of generality, we assume that $\boldsymbol{x} = \boldsymbol{c}$. Since $\boldsymbol{x}' \in B$, it occurs inside C_B (and on C_A). Thus it occurs on $C_A[\boldsymbol{c}, \boldsymbol{c}']$. Since $\angle \boldsymbol{x}\boldsymbol{o}_A\boldsymbol{x}' = 2\pi/3$, $\angle \boldsymbol{c}\boldsymbol{o}_B\boldsymbol{c}' > 2\pi/3$, and there is no point in B on $C_B[\boldsymbol{c}, \boldsymbol{c}']$, $k_B \leq 2$ holds. All what we need to consider is thus the case in which $k_A = 3$ and $k_B = 2$.

We again consider the sequence $\mathcal{X} : \boldsymbol{p}_0, \boldsymbol{q}_0, \boldsymbol{p}_1, \boldsymbol{q}_1, \ldots$ after making two modifications. Since $k_A = 3$ this time, we have two candidates to determine $\boldsymbol{p}_i$ from $\boldsymbol{q}_{i-1}$. Let C be the circle with center $\boldsymbol{o}_A$ such that it contains $\boldsymbol{q}_{i-1}$. Let $\boldsymbol{x}$ and $\boldsymbol{x}'$ be two points on C such that they form an equilateral triangle with $\boldsymbol{q}_{i-1}$. Since $k_A = 3$, they both belong to A. We assume that $\boldsymbol{x}$ has a larger x-coordinate than $\boldsymbol{x}'$ (in case of tie, we assume that $\boldsymbol{x}$ has a larger y-coordinate than $\boldsymbol{x}'$). Then we choose $\boldsymbol{x}$ as $\boldsymbol{p}_i$.

We consider two instances of $\mathcal{X}$. $\mathcal{X}_1$ starts with $\boldsymbol{p}_0 = \boldsymbol{o}_A$, $\boldsymbol{q}_0 = (-2, 0)$, and $\boldsymbol{p}_1 = (1, \sqrt{3})$, and $\mathcal{X}_2$ with $\boldsymbol{p}_0 = \boldsymbol{o}_A$, $\boldsymbol{q}_0 = (-2, 0)$, and $\boldsymbol{p}_1 = (1, -\sqrt{3})$. Then $\boldsymbol{p}_i = (p_x^i, p_y^i)$ (resp. $\boldsymbol{q}_i = (q_x^i, q_y^i)$) in $\mathcal{X}_1$, if and only if $\boldsymbol{p}_i = (p_x^i, -p_y^i)$ (resp. $\boldsymbol{q}_i = (q_x^i, -q_y^i)$) in $\mathcal{X}_2$, and $p_y^i, q_y^i \neq 0$ for all $i \geq 1$. By a similar argument in (A2), they both create an infinite number of distinct points, if they do not reach $\boldsymbol{a}$. Thus at least one of them creates an infinite number of distinct points, no matter where $\boldsymbol{a}$ is. It is a contradiction.

(II) Thus, $\boldsymbol{o}_A = \boldsymbol{o}_B$, if $k_B \geq 2$. It is however a contradiction. Without loss of generality, we may assume that $\boldsymbol{o}_A = \boldsymbol{o}_B = (0, 0)$. Since $k_A, k_B \geq 2$, $\sum_{\boldsymbol{x} \in A} \boldsymbol{x} = \sum_{\boldsymbol{y} \in B} \boldsymbol{y} = (0, 0)$. It is a contradiction, since $\sum_{\boldsymbol{y} \in B} \boldsymbol{y} = (\sum_{\boldsymbol{x} \in A} \boldsymbol{x}) - \boldsymbol{a} + (0, 0)$ and $\boldsymbol{a} \neq \boldsymbol{o}_A = (0, 0)$.

Appendix B. Proof of Lemma 5

Suppose that a robot r_i starts a Look-Compute-Move phase when the configuration is $P = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_n\}$ (in Z_0). Let Z_i and γ_i be the x-y local coordinate system of r_i, and the coordinate transformation from Z_i to Z_0, respectively. Then r_i observes $Q^{(i)} = \{\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_n\}$ in Z_i in Look phase, where $\boldsymbol{p}_j = \gamma_i(\boldsymbol{q}_j)$. By the definition of Z_i, $(0,0) \in Q^{(i)}$, and there is an index j such that $\gamma_i((0,0)) = \boldsymbol{p}_j$. ($P$ may not contain $(0,0)$.)

In Compute phase, r_i computes $\psi_{(n,2)}(Q^{(i)})$ (not $\psi_{(n,2)}(P)$), which is the target point of r_i in Z_i, where $\psi_{(n,2)}(Q^{(i)})$ is either $\boldsymbol{o}_{Q^{(i)}}$, $\boldsymbol{M}_{Q^{(i)}}$, or $\boldsymbol{a}_{Q^{(i)}}$. Immediately, $k_{Q^{(i)}}$, $\boldsymbol{o}_{Q^{(i)}}$, and $\boldsymbol{M}_{Q^{(i)}}$ are computable from $Q^{(i)}$, and $k_{Q^{(i)}} = k_P$, $\gamma_i(\boldsymbol{o}_{Q^{(i)}}) = \boldsymbol{o}_P$, and $\gamma_i(\boldsymbol{M}_{Q^{(i)}}) = \boldsymbol{M}_P$.

To compute $\boldsymbol{a}_{Q^{(i)}}$ for Step 1(b), r_i computes $View_{Q^{(i)}}$. Indeed, it is possible, since r_i can construct $\Xi_{\boldsymbol{q}}$ for all $\boldsymbol{q} \in Q^{(i)} \setminus \{\boldsymbol{o}_{Q^{(i)}}\}$. Then it computes $\succ_{Q^{(i)}}$ and $\boldsymbol{a}_{Q^{(i)}}$. By the definition of $\Xi_{\boldsymbol{q}}$, $View_P = View_{Q^{(i)}}$, which implies $\succ_{Q^{(i)}} = \succ_P$ and $\gamma_i(\boldsymbol{a}_{Q^{(i)}}) = \boldsymbol{a}_P$. Hence when $\psi_{(n,2)}(Q^{(i)})$ is $\boldsymbol{o}_{Q^{(i)}}, \boldsymbol{M}_{Q^{(i)}}$, or $\boldsymbol{a}_{Q^{(i)}}$ in Z_i, r_i moves to $\boldsymbol{o}_P$, $\boldsymbol{M}_P$, or $\boldsymbol{a}_P$ in Z_0, respectively. Using this relation, we analyze $\psi_{(n,2)}$.

Consider any execution $\mathcal{E} : P_0, P_1, \ldots$ of $\psi_{(n,2)}$, starting from any initial configuration P_0 (in Z_0). We show that in $\mathcal{E}$, each robot converges to one of at most two convergence points, provided that at most two robots crash. It is sufficient to show that $\mathcal{E}$ eventually reaches a configuration P_t of type L, since $\mathrm{LN}_{(n,2)}$ is invoked at t and solves FC(2)-PO, without reaching a configuration not in type L.

Our proof scenario is as follows: Using an exhaustive search, we figure out how types of configurations change as the execution evolves. The crucial observations for $\psi_{(n,2)}$, which we shall make, are (i) the transition diagram among the types has a single sink L, (ii) every loop eventually terminates if the execution does not converge to a point or two, and (iii) some transition eventually occurs from each type. Based on these, we conclude that the execution eventually reaches a type L configuration, if it does not converge to one or two convergence points.

The search is exhaustive and is not very difficult, if we gaze the effects of the fairness of scheduler and faulty robots. Consider any configuration P_t not in type L. If every robot r activated at t is either faulty or $\psi_{(n,2)}$ instructs it not to move, $P_{t+1} = P_t$ holds.

A robot r is said to be *ready* at P_t, if it moves once it is activated. We first confirm that at least three ready robots exist, in the following observations. (This confirmation is easy, and we omit to explicitly mention it.) Then at least one non-faulty ready robot r exists in P_t. Since the scheduler is fair, r is activated eventually at some time $t' > t$. If $P_{t'} = P_t$ and r is activated at t', at least r moves, and the execution reaches a configuration $P_{t'+1}$. (Note that $P_{t'+1} = P_{t'}$ may hold, although some robots move, e.g., when two robots exchange their positions.) Thus, if there are three ready robots in P_t, in every execution starting from P_t, there is a time $t' > t$ such that there are robots that

Figure B.4: The transition diagram among types that $\psi_{(n,2)}$ specifies.

move at t'. We therefore assume that some robots always move every time t without loss of generality, provided that there are at least three ready robots.

We first make a series of observations (I)-(VII). Figure B.4 shows the transition diagram among the types, which summarizes the observations (I)–(V). Let $k_t = k_{P_t}$, $m_t = m_{P_t}$, $CH_t = CH(P_t)$, and $\boldsymbol{o}_t$ be the center of the smallest enclosing circle C_t of P_t.

(I) When P_t is type Z. Suppose that P_t is type Z. Then P_{t+1} can be any type in L, T, I, S, and Z. (We regard a type G configuration as type L.) Suppose that P_{t+1} is type Z. We show that this self-loop from Z to Z eventually terminates.

(A) Suppose that $k_t = 1$. Let $\boldsymbol{a}_t$ be the largest point in $\overline{P_t}$ with respect to $\succ_{P_t}$, and let γ_i be the coordinate transformation from Z_i to Z_0. When a robot r_i is activated, in Look phase, it identifies a configuration $Q_t^{(i)}$ such that $\gamma_i(Q_t^{(i)}) = P_t$, and computes $\psi_{(n,2)}(Q_t^{(i)}) = \boldsymbol{a}_t^{(i)}$ in Compute phase. Since $Q_t^{(i)}$ is type Z, and $k_{Q_t^{(i)}} = 1$, $\boldsymbol{a}_t^{(i)}$ is the largest point in $\overline{Q_t^{(i)}}$ with respect to $\succ_{Q_t^{(i)}}$ $(= \ \succ_{P_t})$. Thus $\gamma_i(\boldsymbol{a}_t^{(i)}) = \boldsymbol{a}_t$.

By the definition of $\succ_{P_t}$, $\mu_{P_t}(\boldsymbol{a}_t) \geq \mu_{P_t}(\boldsymbol{p})$ for all $\boldsymbol{p} \in \overline{P_t}$. Thus, $\mu_{P_{t+1}}(\boldsymbol{a}_t) > \mu_{P_t}(\boldsymbol{a}_t) \geq \mu_{P_t}(\boldsymbol{p}) \geq \mu_{P_{t+1}}(\boldsymbol{p})$, for all $\boldsymbol{p} \in \overline{P_t} \setminus \{\boldsymbol{a}_t\}$. Therefore, if P_{t+1} is type Z, $k_{t+1} = 1$ and $\boldsymbol{a}_{t+1} = \boldsymbol{a}_t$.

Suppose that $P_{t'}$ is type Z with $k_{t'} = 1$ for all $t' > t$. Then a contradiction is derived, since $\boldsymbol{a}_{t'} = \boldsymbol{a}_t$ and $\mu_{P_{t'}}(\boldsymbol{a}_t)$ increases unboundedly. Thus the self-loop of Z eventually terminates.

(B) Suppose that $k_t \geq 2$. We show that, if $P_{t'}$ is type Z for all $t' > t$, then there is a time t' such that $k_{t'} = 1$ or $m_{t'} < m_t$ holds, which implies that the self-loop of Z eventually terminates by (A).

Suppose that $k_{t+1} \geq 2$. Then $m_{t+1} \leq m_t + 1$. If $m_{t+1} < m_t$, there is nothing to show. Thus there are two cases to be considered.

First consider the case of $m_{t+1} = m_t$. If $\overline{P_{t+1}} = \overline{P_t}$, then $\boldsymbol{o}_{t+1} = \boldsymbol{o}_t$, and hence $\mu_{t+1}(\boldsymbol{o}_{t+1}) = \mu_{t+1}(\boldsymbol{o}_t) > \mu_t(\boldsymbol{o}_t)$. Thus, there is a time $t' > t+1$ such that $m_{t'} < m_{t+1} = m_t$.

Otherwise if $\overline{P_{t+1}} \neq \overline{P_t}$, $\boldsymbol{o}_t \notin \overline{P_t}$, and there is a $\boldsymbol{q} \in \overline{P_t}$ such that $\overline{P_{t+1}} = (\overline{P_t} \setminus \{\boldsymbol{q}\}) \cup \{\boldsymbol{o}_t\}$. By Lemma 4, $k_{t+1} = 1$.

Next consider the case of $m_{t+1} = m_t + 1$. Then $\boldsymbol{o}_t \notin \overline{P_t}$, and $\overline{P_{t+1}} = \overline{P_t} \cup \{\boldsymbol{o}_t\}$. Since $\boldsymbol{o}_{t+1} = \boldsymbol{o}_t$ and $\mu_{t+1}(\boldsymbol{o}_{t+1}) = \mu_{t+1}(\boldsymbol{o}_t) > \mu_t(\boldsymbol{o}_t)$, there is a time $t' > t+1$ such that $m_{t'} < m_{t+1}$. Without loss of generality, we assume $m_{t+1} = m_{t+2} = \cdots = m_{t'-1}$, i.e., $\overline{P_{t+1}} = \overline{P_{t+2}} = \cdots = \overline{P_{t'-1}}$ and hence $\boldsymbol{o}_{t+1} = \boldsymbol{o}_{t+2} = \cdots = \boldsymbol{o}_{t'-1}$. If $m_{t'} < m_t$, there is nothing to show.

Suppose that $m_{t'} = m_t$. By assumption, at each time $t'' = t, t+1, \ldots, t'-1$, some robots not at $\boldsymbol{o}_t$ move to $\boldsymbol{o}_t$. Let A be the set of robots activated in this period. If all robots in A are activated at t, $P_{t'}$ yields. By Lemma 4, $k_{t'} = 1$.

Thus the self-loop of Z eventually terminates.

(II) When P_t is type L. Then $\mathrm{LN}_{(n,2)}$ is invoked to solve FC(2)-PO. Recall that P_t is type L, so is P_{t+1}. Thus L is a sink in the transition diagram.

(III) When P_t is type T. Let $\overline{P_t} = \{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}\}$.

(A) Suppose that triangle $\boldsymbol{abc}$ is equilateral. Since $\psi_{(n,2)}(P_t) = \boldsymbol{o}_t$,[15] every non-faulty robot, once activated, moves to $\boldsymbol{o}_t$. Thus the type of P_{t+1} is L, T, or I. If the type of P_{t+1} is T, then CH_{t+1} is not equilateral, and $h_{t+1} = (2\sqrt{3}+3)h_t/9$, where $h_t = h(CH_t)$ is the perimeter of CH_t.

(B) Suppose that triangle $\boldsymbol{abc}$ is not equilateral. Since $\psi_{(n,2)}(P_t) = \boldsymbol{M}_t$, and all Z_i are right-handed, the type of P_{t+1} is L, T, or S. where $\boldsymbol{M}_t = \boldsymbol{M}_{P_t}$. If the type of P_{t+1} is T, then $h_{t+1} < h_t$.

We show that CH_t (and thus P_t) converges to a point, if the self-loop of T does not terminate. Suppose that P_t is type T for all $t > t_0$ for some time t_0. After time t_0, CH_t is equilateral at a finite number of times, since if CH_t is equilateral then $h_{t+1} = (2\sqrt{3}+3)h_t/9$. Thus, there is a time t_1 such that CH_t is not equilateral for all $t > t_1$. Since $h_{t+2} \leq 5h_t/6$ by a simple calculation, we conclude that P_t converges to a point.

(IV) When P_t is type I. Since $\psi_{(n,2)}(P_t) = \boldsymbol{o}_t$, the type of P_{t+1} is either L, T, or I.

[15] Formally, the equality "$\psi_{(n,2)}(P_t) = \boldsymbol{o}_t$" does not hold, since P_t may not contain $(0,0)$, despite that the domain of target function $\psi_{(n,2)}$ is $\mathcal{P}$. Here and later when notation $\psi_{(n,2)}(P_t)$ appears, recall the discussion at the beginning of the proof. On robot r_i, $\psi_{(n,2)}$ is not applied to P_t, but to $Q_t^{(i)}$, where $\gamma_i(Q_t^{(i)}) = P_t$. In this case, $\psi_{(n,2)}(Q_t^{(i)}) = \boldsymbol{o}_{Q_t^{(i)}}$. Since $\gamma_i(\boldsymbol{o}_{Q_t^{(i)}}) = \boldsymbol{o}_t$, the target point of r_i is $\boldsymbol{o}_t$ in Z_0. Formally, $\gamma_i(\psi_{(n,2)}(\gamma_i^{-1}(P_t))) = \boldsymbol{o}_t$. This is what "$\psi_{(n,2)}(P_t) = \boldsymbol{o}_t$" means. The same convention applies to $\mathrm{LN}_{(n,2)}$ later in Appendix C.

Suppose that P_{t+1} is type I. We show that this self-loop from I to I eventually terminates.

If P_{t+1} is type I, $\boldsymbol{o}_{t+1} = \boldsymbol{o}_t$, and hence $\mu_{P_{t+1}}(\boldsymbol{o}_{t+1}) = \mu_{P_{t+1}}(\boldsymbol{o}_t) > \mu_{P_t}(\boldsymbol{o}_t)$. Thus this self-loop of I eventually terminates (and the execution will reach a type L or T configuration).

If P_{t+1} is type T, then like (III), $h_{t+1} = (2\sqrt{3}+3)h_t/9$.

(V) When P_t is type S. Since $\psi_{(n,2)}(P_t) = \boldsymbol{M}_t$, the type of P_{t+1} is either L, T, or S.

If P_{t+1} is type S, $\boldsymbol{M}_{t+1} = \boldsymbol{M}_t$, and hence $\mu_{P_{t+1}}(\boldsymbol{M}_{t+1}) = \mu_{P_{t+1}}(\boldsymbol{M}_t) > \mu_{P_t}(\boldsymbol{M}_t)$. Thus this self-loop from S to S eventually terminates (and the execution will reach a type L or T configuration).

(VI) TI$^+$T-loop. Suppose that P_t is type T. If triangle $\boldsymbol{abc}$ is equilateral, the execution may reach a configuration $P_{t'}$ of type T via several type I configurations as observed in (III) and (IV). Now $\overline{P_{t'}} = \{\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{o}_t\}$, where $\boldsymbol{x}, \boldsymbol{y} \in \{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}\}$ and $\boldsymbol{x} \neq \boldsymbol{y}$. Thus $h_{t'} = (2\sqrt{3}+3)h_t/9$.

(VII) TS$^+$T-loop. Suppose again that P_t is type T. If triangle $\boldsymbol{abc}$ is not equilateral, the execution may reach a configuration $P_{t'}$ of type T via several type S configurations as observed in (III) and (V). Now $\overline{P_{t'}} = \{\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{M}_t\}$, where $\boldsymbol{x}, \boldsymbol{y} \in \{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}\}$ and $\boldsymbol{x} \neq \boldsymbol{y}$.

We go on the proof. The observations (I)–(V) summarized in Figure B.4 show that any execution $\mathcal{E}$ starting from a configuration of any type eventually reaches a type L configuration, if neither TI$^+$T-loop nor TS$^+$T-loop repeats infinitely many times.

Let us consider what happens if TI$^+$T-loop or TS$^+$T-loop repeats infinitely many times. Since $\psi_{(n,2)}(P) \in CH(P)$ for all $P \in \mathcal{P}$, $h_{t+1} \leq h_t$ for all t. If TI$^+$T-loop occurs infinitely many times in $\mathcal{E}$, h_t converges to 0 by observation (VI), and hence P_t converges to a point (although $\mathcal{E}$ may not contain a configuration of type L or G).

We assume that $\mathcal{E}$ contains a finite number of occurrences of TI$^+$T-loop. Then there is a t_0 such that the postfix $\mathcal{E}'$ of $\mathcal{E}$: $P_{t_0}, P_{t_0+1}, \ldots$ of $\mathcal{E}$ does not contain an occurrence of TI$^+$T-loop. Since $\mathcal{E}'$ does not contain a type L configuration, it is a repetition of TS$^+$T-loop. Suppose that at time t a TS$^+$T-loop starts and at time $t'(> t)$ the second TS$^+$T-loop ends (counting after t). Then it is easy to observe that $h_{t'} \leq 5h_t/6$ and thus $\mathcal{E}$' (and hence $\mathcal{E}$) converges to a point as observed in (III).

Thus $\mathcal{E}$ eventually reaches a configuration of type L, if it does not converges to a point.

Appendix C. Proof of Lemma 6

Our proof scenario is similar to the proof of Lemma 5. By an exhaustive search, we draw the transition diagram among the types, and show that if the

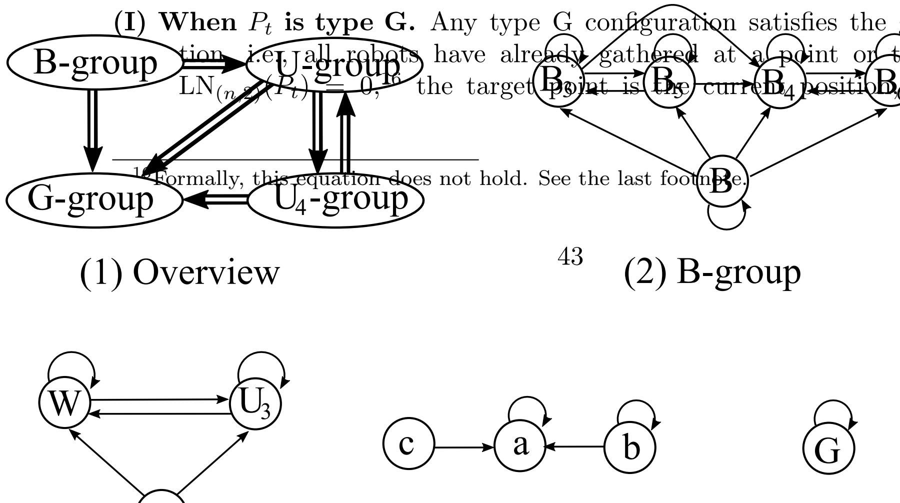


Figure C.5: The transition diagram among the types that $\mathrm{LN}_{(n,2)}$ specifies. Part (1): The overview of the transition diagram. Part (2): B-group contains types B, B_3, B_4, B_5, and B_6. Part (3): U-group contains types U, U_3, and W. Part (4): U_4-group contains three cases (a), (b), and (c) in type U_4. Part (5): G-group contains type G. In each of Parts (2)-(5), an arrow represents a transition between two types (or two cases in U_4-group). In Part (1), a double arrow from a group to another group represents that there is a transition from a type in the former group to a type in the latter one. A loop between U-group and U_4-group in Part (1) eventually terminates by (XI) and (XV).

execution does not converge to at most two points, it eventually reaches a type G configuration.

We borrow symbols and notations from the proof of Lemma 5.

Consider any execution $\mathcal{E} : P_0, P_1, \dots$ starting from any initial configuration P_0 of type L. Note that P_0 (in Z_0) is type L, but may not contain $(0, 0)$. A configuration Q_0 identified by a robot at time 0, however, is type L and contains $(0, 0)$. Then Q_0, after identified with a set of real numbers including 0, applies to $\mathrm{LN}_{(n,2)}$. By the definition of $\mathrm{LN}_{(n,2)}$, $CH_{t+1} \subseteq CH_t$, which implies that P_t is type L for all $t \geq 0$. Like the proof of Lemma 5, we make a series of observations (I)–(XV). Figure C.5 shows the transition diagram among the types, which summarizes the observations (I)–(V), (VII), and (X)–(XIV).

(I) When P_t is type G. Any type G configuration satisfies the goal condition, i.e., all robots have already gathered at a point or two. Since $\mathrm{LN}_{(n,2)}(P_t) = 0$,[16] the target point is the current position, hence the

[16] Formally, this equation does not hold. See the last footnote.

execution stays type G forever, once it reaches a configuration of type G.

(II) When P_t is type B. Suppose that P_t is type B. Then P_{t+1} can be any type.

If P_{t+1} is type B, each activated robot, as long as it is not faulty, moves either to b_1 or b_{m_t} at time t. Unless $m_{t+1} < m_t$ (i.e., if $m_{t+1} = m_t$), $\mu_{P_{t+1}}(b_1) + \mu_{P_{t+1}}(b_{m_t}) > \mu_{P_t}(b_1) + \mu_{P_t}(b_{m_t})$. Thus this self-loop from B to B cannot repeat more than n times, and the execution eventually reaches a configuration whose type is not B.

(III) When P_t is type B_3. Suppose that P_t is type B_3. Then the type of P_{t+1} is either G, B_3, B_4, B_5, U_3, U_4, or U.

Let $L_t = L_{P_t}$, which is the length of CH_t. If P_{t+1} is type B_3, then $L_{t+1} = L_t/2$. Thus the execution converges to a point, if this self-loop from B_3 to B_3 repeats infinitely many times.

(IV) When P_t is type B_4. Suppose that P_t is type B_4. Then the type of P_{t+1} is either G, B_4, B_6, U_3, U_4, or U.

Let $\lambda_t = \lambda_{P_t}$. If P_{t+1} is type B_4, $\lambda_{t+1} = \lambda_t/2$. Thus the execution converges to two points, if this self-loop from B_4 to B_4 repeats infinitely many times.

(V) When P_t is type B_5. Suppose that P_t is type B_5. Then the type of P_{t+1} is either G, B_3, B_4, B_5, U_3, U_4, or U.

If P_{t+1} is type B_5, each activated robot, as long as it is not faulty, moves either to b_2 or b_4 at time t. Unless $m_{t+1} < m_t$ (i.e., if $m_{t+1} = m_t$), $\mu_{P_{t+1}}(b_2) + \mu_{P_{t+1}}(b_4) > \mu_{P_t}(b_2) + \mu_{P_t}(b_4)$. Thus this self-loop from B_5 to B_5 cannot repeat more than n times, and the execution eventually reaches a configuration whose type is not B_5.

(VI) $B_3B_5^+B_3$ loop. Suppose that P_t is type B_3. Then as observed in (III), P_{t+1} can be type B_5. Then as observed in (V), after several repetition of the self-loop of B_5, the execution can reach a configuration $P_{t'}$ of type B_3 for the first time after t. Let $\overline{P_t} = \{a, b, c\}$, where $b - a = c - b$. Then $\overline{P_{t+1}} = \{a, M_{ab}, b, M_{bc}, c\}$, and $\overline{P_{t'}} = \{M_{ab}, b, M_{bc}\}$. Thus $L_{t'} = L_t/2$. If this $B_3B_5^+B_3$ loop repeats infinitely many times, the execution converges to a point.

(VII) When P_t is type B_6. Suppose that P_t is type B_6. Then the type of P_{t+1} is either G, B_4, B_6, U_3, W, U_4, or U.

If P_{t+1} is type B_6, each activated robot, as long as it is not faulty, moves either to b_2 or b_5 at time t. Unless $m_{t+1} < m_t$ (i.e., if $m_{t+1} = m_t$), $\mu_{P_{t+1}}(b_2) + \mu_{P_{t+1}}(b_5) > \mu_{P_t}(b_2) + \mu_{P_t}(b_5)$. Thus this self-loop from B_6 to B_6 cannot repeat more than n times, and the execution eventually reaches a configuration whose type is not B_6.

(VIII) $\mathrm{B}_4\mathrm{B}_6^+\mathrm{B}_4$ loop. Suppose that P_t is type B_4. Then as observed in (IV), P_{t+1} can be type B_6. Then as observed in (VII), after several repetition of the self-loop of B_6, the execution can reach a configuration $P_{t'}$ of type B_4 for the first time after t. Let $\overline{P_t} = \{a, b, c, d\}$, where $a < b < c < d$ and $b - a = d - c$. Then $\overline{P_{t+1}} = \{a, M_{ab}, b, c, M_{cd}, d\}$, and $\overline{P_{t'}} = \{x, M_{ab}, M_{cd}, y\}$, where $x \in \{a, b\}$ and $y \in \{c, d\}$. (We ignore the order among x, M_{ab}, M_{cd}, y in $\overline{P_{t'}}$.) Thus $\lambda_{t'} = \lambda_t/2$. If this $\mathrm{B}_4\mathrm{B}_6^+\mathrm{B}_4$ loop repeats infinitely many times, the execution converges to two points.

(IX) Summary of (II)–(VIII). Suppose that $k_0 = k_{P_0} = 2$. Unless the execution converges to a point or two, by repeating $\mathrm{B}_3\mathrm{B}_5^+\mathrm{B}_3$ loop or $\mathrm{B}_4\mathrm{B}_6^+\mathrm{B}_4$ loop infinitely many times, it eventually reaches a type G configuration or a configuration P_t such that $k_t = 1$.

(X) When P_t is type U. Suppose that P_t is type U. Then the type of P_{t+1} is either G, U_3, W, U_4, or U.

If $b_1 \succ_{P_t} b_{m_t}$, then $\mu_{P_t}(b_1) \geq \mu_{P_t}(b_{m_t})$. Since $\mathrm{LN}_{(n,2)}(P_t) = b_1$, unless $m_{t+1} < m_t$, $\mu_{P_{t+1}}(b_1) > \mu_{P_t}(b_1) \geq \mu_{P_t}(b_{m_t}) \geq \mu_{P_{t+1}}(b_{m_t})$. Thus $b_1 \succ_{P_{t+1}} b_{m_t}$ at $t+1$, and P_{t+1} is type U. This self-loop from U to U cannot repeat more than n times.

Otherwise, if $b_{m_t} \succ b_1$, similarly, P_{t+1} is type U, and this self-loop from U to U cannot repeat more than n times.

Thus the execution eventually reaches a configuration of type G, U_3, W, or U_4.

(XI) When P_t is type U_3. Suppose that P_t is type U_3. Then $k_{t+1} = 1$ unless the type of P_{t+1} is G. The type of P_{t+1} is either G, U_3, or W (but not U_4).

Suppose that the type of P_{t+1} is U_3. Then all the robots at exactly one of b_1, b_2, and b_3 have moved to $\mathrm{LN}_{(n,2)}(P_t)$ at t, which implies that $\lambda_{t+1} \leq 2\lambda_t/3$. Thus the execution eventually converges to two points, if this self-loop from U_3 to U_3 repeats infinitely many times.

(XII) When P_t is type W. Suppose that P_t is type W. By the definition of $\mathrm{LN}_{(n,2)}$, $k_{t+1} = 1$, and the type of P_{t+1} is either G, U_3, or W. Furthermore, if P_{t+1} is type W, then $\lambda_{t+1} = \lambda_t$ holds.

Suppose that $b_3 \leq M_{b_1 b_4}$. Since $\mu_{P_{t+1}}(b_2) > \mu_{P_t}(b_2)$, this self-loop from W to W can repeat at most n times, and the execution eventually reaches a configuration of type G or U_3.

Suppose otherwise that $b_2 \geq M_{b_1 b_4}$. Then $\mu_{P_{t+1}}(b_3) > \mu_{P_t}(b_3)$, this self-loop from W to W can repeat at most n times, and the execution eventually reaches a configuration of type G or U_3.

(XIII) $\mathrm{U}_3\mathrm{W}^+\mathrm{U}_3$ loop. Suppose that the type of P_t is U_3, and that of P_{t+1} is W. As observed in (XII), the execution can reach a configuration $P_{t'}$ of type U_3 for the first time after t. Then by (XI) and (XII), $\lambda_{t'} \leq 2\lambda_t/3$.

Thus the execution converges to two points, if this $U_3W^+U_3$ loop repeats infinitely many times.

(XIV) When P_t is type U_4. Suppose that P_t is type U_4 and satisfies $\mu_{P_t}(b_1) \geq \mu_{P_t}(b_4)$. We consider three cases corresponding to the three cases (a)-(c) in the definition of $LN_{(n,2)}$.

- **(a) When $\mu_{P_t}(b_1) \geq \mu_{P_t}(b_3)$:** Since $LN_{(n,2)}(P_t) = b_1$, $\mu_{P_{t+1}}(b_1) > \mu_{P_t}(b_1) \geq \mu_{P_t}(b_4) \geq \mu_{P_{t+1}}(b_4)$. Thus $k_{t+1} = 1$, $m_{t+1} \leq 4$, and the type of P_{t+1} is either G, U_3, or U_4.

  Moreover, if P_{t+1} is type U_4, then it satisfies the condition (a). Then this self-loop from U_4(a) to U_4(a) cannot repeat more than n times, and the execution eventually reaches a configuration of type G or U_3.

- **(b) When $(\mu_{P_t}(b_1) < \mu_{P_t}(b_3)) \wedge (\mu_{P_t}(b_3) \geq 3)$:** By the definition of $LN_{(n,2)}$, only robots at b_3 can move (to b_1), and the other robots cannot move (even if they are activated), since $LN_{(n,2)}(P_t) = 0$.

  First confirm that at least one robot r at b_3 is non-faulty, and eventually r is activated to change the configuration, since there are at most two faulty robots. Thus $k_{t+1} = 1$, $m_t \leq 4$, and the type of P_{t+1} is either U_3 or U_4.

  If P_{t+1} is type U_4, then it satisfies condition (a) or (b). If P_{t+1} satisfies condition (b), this self-loop from U_4(b) to U_4(b) cannot repeat more than n times, since $\mu_{P_{t+1}}(b_1) > \mu_{P_t}(b_1) \geq \mu_{P_t}(b_4) \geq \mu_{P_{t+1}}(b_4)$, and the execution eventually reaches a configuration of type G or U_3.

  On the other hand, if P_{t+1} satisfies condition (a), then as observed in case (a), the execution eventually reaches a configuration of type G or U_3.

  Thus the execution eventually reaches a configuration of type G or U_3, regardless of whether or not P_{t+1} satisfies condition (b).

- **(c) When $(\mu_{P_t}(b_1) < \mu_{P_t}(b_3)) \wedge (\mu_{P_t}(b_3) < 3)$:** Condition (c) holds, if and only if $1 \leq \mu_{P_t}(b_4) = \mu_{P_t}(b_1) < \mu_{P_t}(b_3) = 2$ (and $\mu_{P_t}(b_2) \geq 1$) hold. Since $\mu_{P_t}(b_2) + \mu_{P_t}(b_3) \geq 3$, there is at least one non-faulty robot r at b_2 or b_3, and r eventually moves to b_1, since $LN_{(n,2)}(P_t) = b_1$, if $(b_2 = 0) \vee (b_3 = 0)$. Since $\mu_{P_{t+1}}(b_1) \neq \mu_{P_{t+1}}(b_4)$, $k_{t+1} = 1$, and $m_{t+1} \leq 4$. Then the type of P_{t+1} is G, U_3, or U_4.

  Moreover if P_{t+1} is type U_4, then it must satisfy condition (a), since $\mu_{t+1}(b_1) \geq 2 \geq \mu_t(b_3) \geq \mu_{t+1}(b_3)$.

Let us summarize: When P_t is type U_4, eventually the execution reaches a configuration of type G or U_3.

(XV) Summary of (X)–(XIV). Suppose that $k_0 = k_{P_0} = 1$. Unless the execution converges to two points by repeating U_3 self-loop or $U_3W^+U_3$ loop infinitely many times, it eventually reaches a type G configuration. That is, the loop between U-group and U_4-group in Figure C.5(1) eventually terminates, by (XI) and the summary of (XIV).

Now, we go on the proof. The observations summarized in Figure C.5 show that any execution $\mathcal{E}$ starting from a configuration of any type eventually reaches a type G configuration, if neither $B_3B_5^+B_3$-loop, $B_4B_6^+B_4$-loop, nor $U_3W^+U_3$-loop repeats infinitely many times. We can conclude the correctness of $LN_{(n,2)}$ by observations (IX) and (XV) which show that any one of these loops cannot repeat infinitely many times without reaching a type G configuration.