Resilient Consensus via Voronoi Communication Graphs

Kelsey Saulnier¹, Lifeng Zhou², George Pappas¹, Vijay Kumar¹

Abstract—Consensus algorithms form the foundation for many distributed algorithms by enabling multiple robots to converge to consistent estimates of global variables using only local communication. However, standard consensus protocols can be easily led astray by non-cooperative team members. As such, the study of resilient forms of consensus is necessary for designing resilient distributed algorithms. W-MSR consensus is one such resilient consensus algorithm that allows for resilient consensus with only local knowledge of the communication graph and no a priori model for the data being shared. However, the verification that a given communication graph meets the strict graph connectivity requirement makes W-MSR difficult to use in practice. In this paper, we show that a commonly used communication graph structure in robotics literature, the communication graph built based on the Voronoi tessellation, automatically results in a sufficiently connected graph to reject a single non-cooperative team member. Further, we show how this graph can be enhanced to reject two non-cooperative team members and provide a roadmap for modifications for further resilience. This contribution will allow for the easy application of resilient consensus to algorithms that already rely on Voronoi-based communication such as distributed coverage and exploration algorithms.

Index Terms—Distributed Robot Systems, Multi-Robot Systems, Networked Robots, Resilience

I. INTRODUCTION

As multi-robot systems become larger, more complex, and operate over larger areas, there is a greater need for computation and coordination solutions that are distributed instead of centralized. One challenging objective in distributed systems is to ensure that the system is in agreement with the solution that is being computed so it is no surprise that consensus algorithms are often heavily featured in distributed computing and coordination. Consensus algorithms are used to allow agents to arrive at an agreement on estimates of variables in a distributed fashion and they appear in wide-ranging applications from distributed filtering [1], distributed field estimation [2], [3], data aggregation [4], formation control [5], flocking [6], and more [7].

Consensus algorithms are sometimes used in cases where a leader is designated to lead the team to a desired value, such as in formation control or guided flocking where specific robots have special information which is used to guide the group [8]. However, this sensitivity to leadership is problematic in the case where no leader is desired since a single malfunctioning, non-cooperative, or malicious agent can guide the entire network of agents’ behavior. Several types of methods have been introduced to guard against such agents such as reputation or trust-based methods [9], where each node gains a score based on its behavior, and fault detection and isolation methods such as [10], [11]. Most methods require each node to have significant knowledge of the communication graph structure. LeBlanc et al. introduced the Weighted-Mean Subsequence Reduced (W-MSR) algorithm to address the problem of non-cooperative team members disrupting the consensus value, without the need for global information about the communication network or computationally complex algorithms [12]. This powerful tool is limited by a strict network connectivity requirement called $(r, s)$-robustness, required to guarantee convergence of the consensus algorithm [13] to a safe value. However, verifying that a given network is sufficiently robust has been shown to be NP-Complete [14], which makes this algorithm generally unsuitable for use in mobile robot teams where the communication network is continuously changing, or even for large static networks if computation time is limited. Prior work shows some exceptions such as [15] where an easily computed bound is used to allow mobile robots to use W-MSR consensus to agree on a flocking direction in the presence of malicious team members. However, this easily computed bound often results in highly connected communication graphs which can be unsuitable for applications where robots must be spread around the environment. Some specialized formation rules have also been studied, such as formations on lattices [16]–[18] and periodic formations [19], with the former being limited to a single non-cooperative agent, and the later being limited to circular periodic patterns.

In this work, we expand the knowledge of graphs that can be used with the W-MSR algorithm. In particular, we will show that the network created by connecting the neighbors in a Voronoi tessellation is another special network that can be used with W-MSR consensus to reject a single non-cooperative member, and we will further provide a method for enhancing that communication graph to allow for two or more non-cooperative team members. The Voronoi tessellation is frequently used to divide work in multi-robot coverage and exploration tasks [2], [20]–[24], so we believe this network and its resilience properties will be very useful for the application of resilient consensus to large mobile robot teams, by eliminating the need to explicitly compute the robustness of the communication graph.

We gratefully acknowledge the support of ARL, DCIST CRA W911NF-17-2-0181, NSF Grants CCR-2112665, and ONR grant N00014-20-S-B001.

¹The authors are with the GRASP Laboratory at the University of Pennsylvania, Philadelphia, PA, 19104, USA (email: saulnier, pappasg, kumar@seas.upenn.edu).

²The author is with the Department of Electrical and Computer Engineering, Drexel University, Philadelphia, PA 19104, USA (email: lz457@drexel.edu).

The authors are with the GRASP Laboratory at the University of Pennsylvania, Philadelphia, PA 19104, USA.
II. BACKGROUND

Let there be a set of robots $\mathcal{V}$, such that $|\mathcal{V}| = N$, and an undirected communication graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ such that if $(v_i, v_j) \in \mathcal{E}$ the robots at $v_i$ and $v_j$ are neighbors indicating that they communicate with each other, denoted $v_i \in \mathcal{N}_i$ and $v_j \in \mathcal{N}_j$. Weighted linear consensus is defined as

$$c^{(i)}[k + 1] = w_{ii}c^{(i)}[k] + \sum_{j \in \mathcal{N}_i} w_{ij}c^{(j)}[k],$$  \hspace{1cm} (1)

where $\sum_{j} w_{ij} = 1$ which, when performed on a connected, undirected, communication graph $\mathcal{G}$ results in the robots achieving asymptotic consensus on a weighted average of the initial values

$$\lim_{k \rightarrow \infty} c^{(i)}[k] = \sum_{i=1}^{n} \alpha_i c^{(i)}[0],$$

for weights $\alpha_i$ such that $\sum_{i} \alpha_i = 1$ at an exponential rate determined by the algebraic connectivity of the graph [25].

A feature of standard linear consensus is that a single robot can lead the network to reach a consensus value arbitrarily far from the the initial conditions by simply not cooperating with the consensus update [26]. If such a robot exists in the network, the remaining robots will converge to the value that the non-cooperative robot is sharing with its neighbors. This can cause arbitrarily bad behavior in the system if the controlling robot is malicious or if the non-cooperative behavior is not part of the design.

The W-MSR algorithm is a modified version of the linear consensus which is designed to reject the influence of a number of non-cooperative robots in the network with only local information about the communication graph and with no model of the data the robots are reaching the consensus on. It is defined for a parameter $F$ as

$$c^{(i)}[k + 1] = w_{ii}c^{(i)}[k] + \sum_{j \in \mathcal{N}_{i-F}[k]} w_{ij}c^{(j)}[k],$$  \hspace{1cm} (2)

where the set $\mathcal{N}_{i-F}[k] \subset \mathcal{N}_i$ is constructed by ordering the set of consensus values provided by neighbors of robot $i$ and removing the neighbors corresponding to the $F$ largest values larger than $c^{(i)}[k]$, and the $F$ smallest values smaller than $c^{(i)}[k]$. This results in at most $2F$ neighbors being removed from the set at each round of W-MSR. Note that, for example, only $F$ neighbors are removed from the set if $\forall \{c^{(j)} : j \in \mathcal{N}_i\}, c^{(j)}[k] < c^{(i)}[k]$, since in that case all $F$ largest values are smaller than $c^{(i)}[k]$ and so they are not removed. W-MSR with parameter $F = 0$ is simply the standard weighted linear consensus defined in [1].

W-MSR consensus will allow robots in a sufficiently robust communication graph $\mathcal{G}$ to reach consensus in the presence of non-cooperative robots. Non-cooperative robots are any robots in the consensus network which are sharing values with their neighbors which were not computed using the consensus algorithm. We assume that non-cooperative robots share the same value with all their neighbors at each consensus step, but are otherwise unrestricted in their behavior. If the network is sufficiently robust, the cooperative robots are guaranteed to reach a safe consensus value, defined as a value in between the maximum and minimum initial values of the cooperative robots,

$$\lim_{k \rightarrow \infty} c^{(i)}[k] = c_s, \hspace{.5cm} \min c[0] \leq c_s \leq \max c[0].$$  \hspace{1cm} (3)

The convergence of W-MSR is contingent on each robot in the communication graph $\mathcal{G}$ having a sufficient quantity and diversity of communication, quantified by a graph property called $(r, s)$-robustness.

Definition 1 ($(r, s)$-robust): A communication network graph $\mathcal{G}(\mathcal{E}, \mathcal{V})$ is $(r, s)$-robust if and only if for every pair of non-empty disjoint subsets $S_1, S_2 \subset \mathcal{V}$ at least one of the following holds

1) $|X_{S}^{X_{S_{1}}} = |S_{1}|$
2) $|X_{S}^{X_{S_{2}}} = |S_{2}|$
3) $|X_{S}^{X_{S_{1}}} + |X_{S}^{X_{S_{2}}}| \geq s$

where $X_{S}^{X_{S_{k}}} = \{v_i \in \mathcal{S}_k : |\mathcal{N}_i \setminus \mathcal{S}_k| \geq r\}$ is the number of nodes in $\mathcal{S}_k$ with at least $r$ neighbors outside $\mathcal{S}_k$.

The required level of robustness to guarantee convergence under W-MSR is determined by the threat model. If the total number of non-cooperative robots in the network $N_{nc}$ is bounded by parameter $F$, this is called the $F$-global threat model. To guarantee convergence in this case the communication graph must be at least $(F+1, F+1)$-robust. Alternatively under the $F$-local threat model there are no more than $F$ non-cooperative robots in the neighbor set of every cooperative robot. This can result in a $N_{nc} >> F$. In the $F$-local case the graph must be at least $(2F+1, 1)$-robust. The non-cooperative robots can fail to reach consensus or be corrupted by the non-cooperative robots if the graph does not have a sufficiently high level of $(r, s)$-robustness. This failure to converge can happen when using W-MSR with parameter $F > 0$ even in the case where $N_{nc} = 0$ if the graph is not sufficiently robust, since robots can have so few neighbors that $\mathcal{N}_{i-F} = \emptyset$ at all times.

The $(r, s)$-robustness is coNP-Complete to verify for a given $r$, $s$, and communication network graph $\mathcal{G}$ [14] and therefore it is useful to consider cases where sufficient robustness can be guaranteed without requiring explicit verification. This is particularly important in mobile robot systems, where communication graphs may vary over time, and in systems with large numbers of robots to deploy where even one time verification can take significant computation time.

In this paper, we add to the state of the art by showing that the graph created by connecting neighbors in a Voronoi tessellation, also called the Delaunay triangulation, is $(2, 2)$-robust and therefore can support W-MSR with $F = 1$, to reject a single non-cooperative robot under the $F$-global threat model. Furthermore, we prove that the Voronoi graph can be augmented by connecting two hop neighbors to achieve $(3, 3)$-robustness, required for handling an $F$-global threat with $F = 2$, or $F$-local threat with $F = 1$. Finally, we show numerical evidence that this augmentation scheme can be continued to create networks with increasing robustness guarantees.

III. RESILIENT CONSENSUS FOR VORONOI GRAPHS

In this section, we prove that the communication graph created by connecting each robot to the robots in neighboring
Voronoi cells create a graph, $G_\Delta$, that is resilient to a single non-cooperative robot under the $F$-global threat model. An example of this graph is shown in Figure 1. This result is related to previous work by Saldaña et al. where the authors argue that all triangular graphs are $(2,2)$-robust. There are technical challenges that prevent us from directly using their result, so we provide a proof specific to Voronoi communication graphs. Further, we prove that this graph can be enhanced by connecting two-hop neighbors to create a graph that is resilient to 2 non-cooperative robots under the $F$-global threat model or 1 non-cooperative robot under the $F$-local threat model. Lastly, we show numerical evidence that the graph is resilient to 2 non-cooperative robots under the $F$-local threat model or 1 non-cooperative robot under the $F$-global threat model.

**Theorem 1:** The communication graph, $G_\Delta(V,E)$, formed by the Delaunay triangulation with $N > 2$ robots is $(2,2)$-robust.

In the following proof, and other proofs in this paper we will use the notation $G[S]$ to indicate the induced subgraph of $G(V,E)$ formed by the vertex set $S$. This is the subgraph containing vertices $S \subseteq V$ and all edges in $E$ which have both endpoints in $S$. We use $\bullet$ to indicate an ordered path of vertices and $\{\bullet\}$ to indicate an unordered set of vertices.

**Proof:** Given any pair of non-empty disjoint subsets $S_1, S_2 \subseteq V$, let $\{S_a,i\}$ be the set of connected components of the induced subgraph $G_\Delta[S_a]$, where $S_a \in \{S_1, S_2\}$. There are then the following cases:

1) $\exists S_a$ such that $|S_a,i| = 1$, $\forall i$. All connected components of $G_\Delta[S_a]$ are size 1. Since all nodes have degree $\geq 2$ (triangle faces), this means that each subset of size 1 has at least 2 edges leaving the set and therefore $|X^2_{S_a}| = |S_a|$, satisfying Condition 1 or 2 in Definition 1.

2) $\exists l_1, l_2, \ldots, l_t, s.t.,$ $S_1,l_1 > 1$ and $S_2,l_2 > 1$, i.e., $G_\Delta[S_1]$ and $G_\Delta[S_2]$ each contain a connected component with multiple vertices. First consider $S_2,l_2$. We know there exists a path $(v_i, v_j, \ldots, v_y)$ from $S_1$ to $S_2,l_2$ such that $v_i \in S_1, v_j \in S_1, v_i \notin S_1$, and $v_y \in S_2,l_2$. A diagram of this path can be seen at the top of Figure 2. We know $v_j$ has a neighbor $v_l \in S_1$ (since $S_2,l_2$ is a connected component of at least two vertices, worst case $v_l$ can be the other node in this connected component).

The bottom panel of Figure 2 shows this path. Since $G$ is a triangulation, all the interior faces must be triangles. Thus there exists an ordering of the neighbors of $v_j$ such that $T_{j,n,m}$, $n = 1, \ldots, |N_{v_j}| - 1$, $m = 2, \ldots, |N_{v_j}|$ are triangles (notation $T_{j,n,m}$ is a triangle face with vertices $v_j, v_n, v_m$). Note if $v_j$ is not on the convex hull, there is another triangle $T_{j,n,m}$, but it is not needed in the proof). This implies that the neighbors of $v_j$ form a connected subset which is a simple path or cycle containing all neighbors. Thus, since $v_j$ and $v_l$ are both neighbors of $v_j$, there exists a path $(v_i, v_j, v_m, \ldots, v_j)$ $\in N_{v_j}$ such that $v_n \in S_1, v_m \in S_1$. Therefore $v_m \in \{v_j, v_n\} \in S_1$, proving that $|X^2_{S_1}| \geq 2$. Following the same logic with $S_1,l_1$ produces the result $|X^2_{S_a}| \geq 1$, therefore $|X^2_{S_a}| + |X^2_{S_a}| \geq 2$, satisfying Condition 3 in Definition 1 for $r = s = 2$.

Since for every pair of disjoint subsets of $V$ one of the two cases holds, the graph satisfies the conditions in Definition 1 with $r = 2, s = 2$ for all pairs of disjoint subsets and is $(2,2)$-robust.

The graph $G_\Delta$ can be used with W-MSR with parameter $F = 1$ to reject a single $F$-global non-cooperative robot. We will now provide a method to increase the $(r,s)$-robustness and reject a larger set of non-cooperative robots. To this end we define an enhanced graph, $G_{\Delta2}$, which is the graph $G_\Delta$ with additional edges to directly connect 2-hop neighbors. A comparison of $G_\Delta$ and $G_{\Delta2}$ for several interesting formations can be seen in Figure 3.

**Theorem 2:** If the communication graph $G_\Delta(E_\Delta, V_\Delta)$ formed by connecting the Voronoi neighbors with $N > 4$ robots is extended to $G_{\Delta2}(E_{\Delta2}, V_{\Delta2})$ with $E_\Delta \subseteq E_{\Delta2}$ such that if $(v, w) \in E_\Delta$ and $(v, n) \in E_{\Delta2}$ then $(v, z) \in E_{\Delta2}$ (connecting nodes in $G_\Delta$ to their neighbor’s neighbors), then the resulting graph is $(3,3)$-robust.

The proof of this theorem is included in the Appendix.

**Corollary 2.1:** The graph $G_{\Delta2}$ when $F = 1$ is also resilient to $F$-local non-cooperative robots, where $F$-local indicates fewer than $F$ non-cooperative neighbors in the neighborhood of every robot.

![Fig. 1: Robot positions are shown in green, the current Voronoi partition in grey, and the communication graph $G$ calculated using Delaunay triangulation is shown in blue.](image1.png)

![Fig. 2: When $G[S_1]$ and $G[S_2]$ each have at least one connected component with 2 or more vertices, there exists a path from a vertex $v_i \in S_1$ (pink) to a vertex $v_g \in S_2$ (blue) such that $|N_{v_i} \cap N_{v_g}| \geq 2$.](image2.png)
A. Distributed Parameter Estimation

In this example, a set of $N$ robots are distributed over an area and are tasked with reaching consensus on a parameter $\alpha$ for which each robot has an estimate $\alpha_i$. In the network there are some number, $N_{nc}$, of non-cooperative robots. In this simple example, the robots are stationary during consensus which will allow for an investigation of the graphs $G_\Delta$ and $G_{\Delta^2}$.

Figure 3 shows a group of $N = 100$ robots randomly distributed in a coverage area. In this case, there are $N_{nc} = 2$ non-cooperative robots which share values outside the safe convergence region to their neighbors shown as the red dotted lines on the plots. It can be clearly seen that the plain consensus does not provide a safe value or consensus among the cooperative agents as they each receive different amounts of influence from the two non-cooperative robots. When using W-MSR with $F = 1$, $F$ is too small to handle the $N_{nc} = 2$ F-global threat. Despite this, the cooperative robots are able to reach a consensus on a safe value on $G_\Delta$. This is likely due to the fact that no cooperative robot has multiple non-cooperative neighbors. On $G_{\Delta^2}$, however, the cooperative agents are converging to the value of one of the non-cooperative robots. This shows an interesting interplay between connectivity and resilience. The greater connectivity of $G_{\Delta^2}$ provides greater resilience, but if $F < N_{nc}$, it can also increase the number of robots that have more than $F$ non-cooperative neighbors, causing W-MSR to fail. When using an appropriate parameter $F = N_{nc} = 2$, $G_{\Delta^2}$ is able to reject both non-cooperate agents. This can be seen on the far right of Figure 4. In contrast, $G_\Delta$ does not provide enough connectivity to successfully run W-MSR with $F = 2$, which is evidenced by a single cooperative robot with a constant consensus value. That robot has only 2 neighbors which its will always discard with parameter $F = 2$ and therefore it never deviates from its initial value.

We also examine the influence the non-cooperative robot is able to exert on the final consensus value of the non-cooperative agents when using resilient consensus. Figure 5 shows a network of $N = 20$ agents arranged in a circular configuration with $N_{nc} = 1$. This example shows the effect the non-cooperative agent can have on the cooperative agents in a worst-case scenario. In this case, using $G_\Delta$, the non-cooperative agent is neighbors with all the other agents, and the cooperative agents each have only 2 cooperative neighbors. We know from the use of $G_\Delta$ that the network is $(2, 2)$-robust, so the value the non-cooperative agent shares will be discarded under the W-MSR algorithm. However, the cooperative agents

https://youtu.be/zZ5RLdVlUEY
B. Polygon Rendezvous

In this example, a set of $N$ robots are distributed in the environment and must come to an agreement on a rendezvous position. Each robot has an assigned offset and angle from the agreed rendezvous point, with the locations corresponding to the corners of an $N$-sided regular polygon. At each time step, the robots run consensus using their current Voronoi tessellation to determine their neighbors according to $G_\Delta$ or $G_{\Delta 2}$. Robots seek to find consensus on the rendezvous center location coordinates $(c_x, c_y)$ individually:

$$
c_x^{(i)}[k + 1] = \frac{1}{|N_{i,F}| + 1} \left( c_x^{(i)}[k] + \sum_{j \in N_{i,F}} c_x^{(j)}[k] \right)
$$

and the same for $c_y$, where $N_{i,F}$ is the set of neighbors after discarding for W-MSR (with the plain consensus being $F = 0$ indicating no reduction of the neighbor set). The robots then compute their current goal location from their estimate of the rendezvous center as

$$
\begin{bmatrix}
g_x^{(i)}[k] \\
g_y^{(i)}[k]
\end{bmatrix} = \begin{bmatrix}
c_x^{(i)}[k] + r \cos \left( \frac{2\pi}{N} \right)
\end{bmatrix}
$$

for a preassigned radius $r$ and robot number $i$. This distributes the robots to preassigned locations relative to the rendezvous point. Each robot then moves towards the rendezvous point with a basic proportional controller with a max velocity:

$$p^{(i)}[k + 1] = p^{(i)}[k] + \tau \min \left( \| g^{(i)}[k] - p^{(i)}[k] \|, \upsilon_{\text{max}} \right)
$$

for some time step $\tau$, robot positions $p^{(i)} = [p^{(i)}_x, p^{(i)}_y]$, and goal positions $g^{(i)} = [g_x^{(i)}, g_y^{(i)}]$. The first polygon rendezvous experiment explores the effects of W-MSR when there are no non-cooperative robots in the team. The results can be seen in Figure 6. It shows that the consensus converges in all cases except the case where W-MSR with $F = 2$ is being run on the Voronoi graph $G_\Delta$ which does not have enough connectivity to support $F = 2$. The second polygon rendezvous experiment introduces a single non-cooperative robot into the team which shares a constantly changing rendezvous location with its neighbors in an attempt to lead them away from the safe green area defined by the cooperative initial conditions. The results can be seen in Figure 7. In this case, plain consensus follows the non-cooperative robot away from the safe convergence zone. This is the basic leader-follower behavior that consensus is occasionally used for, however, in this case, it is not the desired behavior because the leader is non-cooperative. For the cases where W-MSR is being run with $F = 1$, the non-cooperative robot’s rendezvous information is rejected and the cooperative robots are able to reach a consensus on a safe value. The same is true for W-MSR with $F = 2$ on $G_{\Delta 2}$, though as before it cannot converge on $G_{\Delta 2}$ due to lack of connectivity. Finally, the same experiment was run with 2 non-cooperative robots attempting to lead the team away from the safe consensus area, the results of which can be seen in Figure 8. In this case, the robots follow the non-cooperative robots in the plain consensus and the W-MSR consensus
with $F = 1$. In the plain consensus, the robots end up not reaching consensus on $G_\Delta$ though most cooperative robots closely follow the orange non-cooperative robot, which is the one with the most neighbors in $G_\Delta$. With $G_{\Delta_2}$, the robots have much higher connectivity and reach values closer to the average of the non-cooperative center locations. For W-MSR with $F = 1$, the algorithm is able to reject one of the non-cooperative robots but not both, resulting in the cooperative robots following one of the non-cooperative robots away from the safe zone. Finally, when running W-MSR with $F = 2$, the robots are able to reach a consensus on a safe value on the enhanced graph $G_{\Delta_2}$, but not on the lower connectivity triangulation graph $G_\Delta$.

### C. Map Consensus

The final example we will show is a case of map consensus. In this scenario, a team of robots is looking to achieve consensus on an occupancy map of a simple hallway environment. A non-cooperative robot is attempting to influence the team by sharing values indicating that part of the hallway is inaccessible. Each cell’s occupancy is represented as a number $c_i \in [0, 1]$ where 0 indicates free space, 1 occupied space, and the numbers between are used to express uncertainty. In this example, the robots sense their environment noise-free in a square area around their location. The robots communicate according to the communication graph $G_\Delta$, and utilize a simple controller which moves them towards the nearest unknown cell that can be reached. If the robots perceive that there are no more reachable unknown cells, they consider the exploration complete and stop their motion. Unlike previous examples, robots in this example begin with no initial value for many of the cells in the environment. Robots keep track of which cells they do not know and run consensus only after they receive information about a cell from a neighbor. In the W-MSR case, this requires at least one neighbor to exist in the reduced neighbor set $\mathcal{N}_i - F$. Robots with no information use a neutral value of 0.5 as their value when deciding which neighbors to disregard. Figure 9a shows the hallway environment populated by cooperative robots (green) and one non-cooperative robot (red X). The non-cooperative robot is communicating with its neighbors that cells circled in red are occupied. Figures 9b and 9d show how the understanding of the environment has progressed after 5 consensus steps from the perspective of an example robot marked in the figures with a blue star. Two effects are clearly seen here. There are more cells in the W-MSR case which are still unknown to the example.
Definition 1 and beginning with $G_r$ corresponding Voronoi graph randomly scaled to produce rectangles ranging from almost locations within a rectangular environment which has been form distribution, and distributing them uniformly at random created by choosing $N$ In the first numerical study, randomly generated graphs were $G$ or equivalently connecting the 2-hop neighbors of $r$-bors in robustness gained by continuing to connect more distant neigh-
ors within a rectangular environment. It takes 36 steps for all robots in the W-MSR algorithm, i.e., W-MSR to run the W-MSR algorithm, i.e., W-MSR will provide no information if only one neighbor has information or if only two neighbors have information and it is conflicting. The robots running plain consensus exercise no such caution and it can be seen that the example robot believes there are walls where the non-cooperative agent is asserting there are walls. By 13 steps in the plain consensus case, and 26 steps in the W-MSR case, the example robot believes that it has a full understanding of the environment. It takes 36 steps for all robots in the W-MSR case to reach a consensus on the map. Figure 8a and 8b show the final belief of the hallways configuration for the example robot for plain consensus and W-MSR consensus respectively. It is clear that the non-cooperative robot is able to prevent exploration of the top section of the hallway when the robots are using plain consensus, but is unable to do so when the robots are running the resilient W-MSR. The cost for resilience is that W-MSR takes significantly longer to converge since new information takes longer to propagate through the graph and because it requires more than one robot to measure each cell in the environment.

V. RESILIENT CONSENSUS FOR ENHANCED VORONOI GRAPHS

Numerical studies were conducted to explore the $(r, s)$-robustness gained by continuing to connect more distant neighbors in $G_r$. To this end we define $G_{r,k}$ as the graph created from $G_r$ by connecting $k$-hop neighbors for $k = 2, ..., K$, or equivalently connecting the 2-hop neighbors of $G_{r,k-1}$. In the first numerical study, randomly generated graphs were created by choosing $N$ robots, $N \in [10, 19]$, from a uniform distribution, and distributing them uniformly at random locations within a rectangular environment which has been randomly scaled to produce rectangles ranging from almost square to long and thin. From the generated robot positions, the corresponding Voronoi graph $G_r$ was constructed. Following Definition 1 and beginning with $r = s = \lfloor N/2 \rfloor$, every pair of disjoint subsets of vertices $\{S_1, S_2\}$ was checked to verify $(r, s)$-robustness of the graph. In the case where a pair of sets did not meet the conditions in Definition 1 for the current $r = s$, $r$ and $s$ were decremented until the conditions were met. This resulted in an algorithm that found the maximum $(r, s)$-robustness such that $r = s$ for each graph. Since $(F+1, F+1)$-robustness can reject $F$-global malicious robots, the largest $r = s$ is sufficient to understand to what extent the graph is robust to this threat model. Graphs with $N \geq 20$ were not tested due to the exponentially increasing run time as the number of pairs of disjoint subsets of vertices scales with $O(3^N)$. Table I shows the results of 100 random graphs generated for each graph type. 100 graphs were generated for each of the graph types ($G_{r,s}, G_{r,k},$ etc.). Table I shows the percentage of each graph type which achieved $(r, s)$-robustness for each $r = s$. Note that graphs which are $(r, s)$-robust are also $(r-l, s-l)$-robust.

![Diagram showing consensus to a rendezvous location with $N_{oc} = 2$. Non-cooperative robots (dotted lines, blue 0 and orange 1 circles) initially share a safe center point for consensus, then move their point farther from the safe area with each step. For plain consensus and for W-MSR with $F = 1$, the robots follow the non-cooperative robots and since they are communicating different values, the cooperative robots do not reach consensus. In the W-MSR $F = 1$ case, the robots successfully reject one of the non-cooperative robots but $F = 1 < N_{oc}$ is not sufficient to reject both. For W-MSR with $F = 2$, the consensus is reached only on $G_{r,2}$. In the case $F = 2$ with communication graph $G_r$, the consensus is not reached because there is not enough connectivity resulting in a non-circular final formation.]

| $r = s$ | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | KN |
|--------|---|---|---|---|---|---|---|---|---|----|
| $G_{r}$ | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 11% |
| $G_{r,2}$ | 100% | 81% | 18% | 1% | 100% | 82% | 64% | 38% | 14% | 26% |
| $G_{r,3}$ | 100% | 82% | 64% | 38% | 14% | 2% | 100% | 75% | 62% | 47% |
| $G_{r,4}$ | 100% | 75% | 62% | 47% | 33% | 12% | 100% | 100% | 100% | 99% |

TABLE I: The percentage of randomly generated graphs which have particular $(r, s)$-robustness assuming $r = s$. KN indicates the percentage of the randomly generated graphs that are complete, i.e., have all possible edges.
this resilience can be improved by connecting 2-hop neighbors. The connections are based on the distance between vertices in the Voronoi diagram, allowing the network to maintain a high level of connectivity and robustness.

### TABLE II: $(r, s)$-robustness for the “two lines” formation on 10 vertices and 19 vertices.

| N  | $r = s$ | # edges |
|----|---------|---------|
| 11 | 2       | 19      |
| 19 | 3       | 35      |
| 26 | 5       | 66      |
| 34 | 4       | 45      |
| 43 | 6       | 52      |
| 52 | 5       | 55      |
| 61 | 6       | 55      |
| 70 | 6       | 55      |
| 79 | 6       | 55      |

The connectivity increases since by $G_{\Delta4}$, most of the generated graphs are complete.

To explore the limits of the graph extension method, it is more helpful to look at a low connectivity case. The “two lines” example in Figure 3d provides just such a case. In this example, $G_{\Delta}$ is the same graph that would be constructed if starting with a single triangle face on three vertices, each additional vertex is added to maximize the average shortest path from the current vertices to the new vertex. In this case, we find a linear relationship between $(r, s)$-robustness and the level of hops which are connected from the base graph $G_{\Delta}$, with the single caveat that there must be sufficient vertices, e.g., a five-vertex graph cannot be $(4, 4)$-robust no matter how many hops away vertices are connected, this requires at least seven vertices.

Table II shows how the $(r, s)$-robustness increases as more distant neighbors in $G_{\Delta}$ are connected. The formation was tested with $N = 11$ (the smallest number for $(6, 6)$-robust) and $N = 19$ to show how the number of edges scales as more vertices are added. The $(r, s)$-robustness an be seen to increase linearly with $K$ for $G_{\Delta K}$, providing further evidence for the proposed method of increasing robustness. Also of note is that for the two lines formation the number of edges for a given $K$ scales roughly linearly with the number of robots. Other formations will have different scaling properties, but this indicates that at least for some formations $G_{\Delta K}$ provides a communication graph that scales well with the number of robots and the desired $(r, s)$-robustness.

### VI. Conclusions

In this paper, we prove that the communication graph built by connecting Voronoi neighbors is resilient to a single F-global non-cooperative team member. We further prove that this resilience can be improved by connecting 2-hop neighbors in order to reject 2 non-cooperative robots in the network or 1 non-cooperative robot in the neighborhood of every cooperative robot. We also provide numerical evidence that this resilience can be further improved by continuing to connect more distant neighbors in the Voronoi communication graph. This opens the door to the use of resilient consensus with many coverage and exploration algorithms that already assume the use of Voronoi tessellations and/or Voronoi communication networks and also provides a new method for constructing resilient networks for other applications. We show several examples of the use of Voronoi communication graphs and their extended versions in domains where direct verification of robustness would be prohibitively expensive such as in large stationary networks, and continuously varying networks. Finally, we show a complex example of consensus on an occupancy grid map, where resilient consensus allows the robots to fully explore the environment despite a malicious team member.

The results in this paper suggest several directions for future research. One interesting direction is the application of resilient consensus to tasks such as distributed filtering and distributed target tracking. Tasks for which information is
privileged based on sensing location, such as in target tracking, will require new ways of redundantly partitioning work to ensure that enough information is available to run resilient consensus. A resilient graph will not be enough if only a single robot has information to share. This challenge also appears in the mapping example in this paper, where the naive solution of allowing each robot to seek information that it doesn’t have results in significant clustering as large groups of robots move together to explore the remaining uncertain cells. Resilient distributed filtering, such a resilient form of the distributed Kalman filter \( \Pi \), also relies on consensus, so future research might examine the effect of using W-MSR consensus with such algorithms. Finally, since this work relies on computing Voronoi neighbors, it would be useful to study the interaction between the results presented here and existing distributed Voronoi partition algorithms and the distributed construction of resilient Voronoi communication graphs.

APPENDIX

PROOF: \( (3,3) \)-ROBUSTNESS OF \( G_{\Delta 2} \)

Proof: First we will cover some notation that is used throughout the proof. First, it is often necessary to indicate edges and neighbors that are in the subgraph \( G_{\Delta} \), the Delaunay triangulation that was used to create \( G_{\Delta 2} \). We refer to these edges as \( \Delta \)-edges, and neighbors as \( \Delta \)-neighbors, with a \( \Delta \)-path being a path using only \( \Delta \)-edges. Further, we use the notation \( N_j \) to indicate the set of \( v_j \)'s \( \Delta \)-neighbors and \( N_j \) to refer to the set of all \( v_j \)'s neighbors.

Second, we will frequently make use of the property that the \( \Delta \)-neighbors of a vertex \( v_j \) form a simple path in \( G_{\Delta} \). This is also used in the proof for Theorem 1. We also occasionally use the definition of triangulation as a graph with all triangular faces except possibly the outer face which is a cycle.

For this proof we will break the cases for \( S_1 \) and \( S_2 \) into two main cases. The first case is when the connected components of either \( G_{\Delta 2}[S_1] \) or \( G_{\Delta 2}[S_2] \) contain no edges from \( E_{\Delta} \), or consist of exactly two vertices with a connection by an edge in \( E_{\Delta} \). This case is handled separately because it encapsulates all cases where Conditions 1 and 2 from Definition 1 are required. The second case is when for both \( \forall G_{\Delta 2}[S_1] \) and \( \forall G_{\Delta 2}[S_2] \) there is at least one connected component which contains a \( \Delta \)-path of more than 3 vertices, or a connected component with two \( \Delta \)-connected vertices and at least 1 in-set extended edge. In this second case, we will instead show that Condition 3 of Definition 1 holds.

1) The connected components of either \( G_{\Delta 2}[S_1] \) or \( G_{\Delta 2}[S_2] \) contain no edges from \( E_{\Delta} \), or consist of exactly two vertices with a connection by an edge in \( E_{\Delta} \). Call the set for which this is true \( S_a \in \{ S_1, S_2 \} \). In this case we will show that every vertex in \( S_a \) is in \( \lambda_3^2 \) which then satisfies Condition 1 or 2 from Definition 1.

a) Consider any connected component of \( G_{\Delta 2}[S_a] \) which contains no \( \Delta \)-edges. Denote the set of vertices in this connected component as \( S_{a,E} \subset S_a \). Consider vertex \( v_i \in S_{a,E} \). Since \( v_i \in S_{a,E} \), we know it has no \( \Delta \)-edges to any other vertices in \( S_a \). Since \( G_{\Delta} \) is formed via Delaunay triangulation, we know all the faces are triangles. Thus \( v_i \) is the vertex of at least one triangle and the other two vertices of this triangle, denoted \( v_j, v_k \), must be in \( S_a \), the complement of \( S_a \). If \( v_i \) is the vertex of more than one triangle, then \( v_i \in \lambda_3^2 \). If \( v_i \) is the vertex of only a single triangle face, then since we also know that \( |N_{v_2}| \geq 5 \), the graph contains at least three triangle faces. By our assumption, none of the remaining triangles can include \( v_i, v_j, v_k \) must be part of a second triangle with a shared neighbor \( v_z \). The third triangle could share edge \(( v_j, v_z ) \) or \(( v_k, v_z ) \). Note that adding a triangle face which includes the edge \(( v_j, v_z ) \) again does not result in a Delaunay triangulation because a single edge cannot be shared by more than two triangle faces. Denote the third vertex of this triangle as \( v_z ' \). We know \( v_z, v_z ' \in N_{v_i} \) via extended, 2-hop edges since each shares at least one \( \Delta \)-neighbor with \( v_i \). We know that \( v_z, v_z ' \) cannot both be in \( S_{a,E} \) since \( v_z, v_z ' \in N_{\Delta 2} \) (and vice versa) and \( S_{a,E} \) is defined as having no in-set \( \Delta \)-edges. Therefore at most one \( v_z \) or \( v_z ' \) is in \( S_a \). That leaves at least three nodes \( v_j, v_k, v_z \) or \( v_z, v_z ' \) in \( N_{\Delta 2} \) or \( S_a \) such that \( v_i \in \lambda_3^2 \). Thus \( \forall i \in S_{a,E} \) and \( \forall S_{a,E} \in S_a, \ i \in \lambda_3^2 \). To sum up, if \( \exists S_a \in \{ S_1, S_2 \} \) such that \( G_{\Delta 2}[S_a] \) contains only connected components that either have no \( \Delta \) edges, or are exactly two vertices connected with a \( \Delta \) edge, then all vertices in that set are in \( \lambda_3^2 \).

2) The remaining case is that in both \( G_{\Delta 2}[S_1] \) and \( G_{\Delta 2}[S_2] \)
there is at least one connected component which contains a \( \Delta \)-path of more than three vertices, or a connected component with two \( \Delta \)-connected vertices and at least one extended edge. We will show that in this case \( S_1 \) has at least two vertices in \( X^3_{S_1} \) and since we have the same assumption made about both sets we can then apply the same logic to \( S_2 \) which allows us to conclude that \( |X^3_{S_1}| + |X^3_{S_2}| \geq 4 \). Since \( 4 \geq 3 \), this more than satisfies Condition 3 for \( r = 3 \) and \( s = 3 \).

Consider a vertex \( v_j \in S_1 \) such that \( |S_1 \cap N_{S_2}| \neq \emptyset \) and a vertex \( g \in S_2 \) which belongs to a connected component of \( G_\Delta[S_2] \) with 3 or more vertices, or of only 2 vertices at least one of which has an in-set extended edge. We know there is a \( \Delta \)-path between these two vertices since \( G_\Delta \) is connected. Since this path starts in \( S_1 \) and ends in \( S_2 \), we know there must be a pair of vertices on the path, \( v_i, v_j \), for which \( v_i \in S_1 \) and \( v_j \in S_2 \). Thus there is a \( \Delta \)-path \( (v_s, v_k, v_l, v_j, v_g, v_j, v_i) \) for which \( v_s, v_k, v_l, v_j, v_i \in S_1 \) (with possibly \( v_g = v_j \)) belong to a connected component of \( G_\Delta[S_1] \) with at least two vertices. This path can also always be defined such that \( (v_j, v_g, v_i) \in S_1 \) as shown in Figure 11a. To see this, consider \( (v_j, v_g, v_i) \notin S_1 \) so there \( \exists v_m \in S_1 \) on the path between \( v_j \) and \( v_g \). If \( N_{m,\Delta} \cap S_1 = \emptyset \), then we know all \( N_{m,\Delta} \in S_1 \). Since the \( \Delta \)-neighbors of every vertex in \( G_\Delta \) form a \( \Delta \)-path, we then know that there is a \( \Delta \)-path in \( S_1 \) which can circumvent \( v_m \) to create a path \( (v_j, \ldots, v_g) \notin S_1 \). Alternatively if \( |N_{m,\Delta} \cap S_1| \geq 1 \), then \( v_m \) is part of a connected component of \( G_\Delta[S_1] \) with 2 or more vertices and we can redefine \( v_s = v_m \) and obtain a path where \( v_m \) is not between \( v_j \) and \( v_g \). These cases are shown in Figure 11b.

Therefore there exists a \( \Delta \)-path \( (v_s, \ldots, v_k, v_l, v_j, v_g, v_i) \) for which \( (v_s, \ldots, v_k, v_l, v_j) \in S_1 \), \( (v_j, v_g, v_i) \in S_1 \), and due to the definitions of the connected components \( v_s \) and \( v_g \) are chosen from \( \{v_j, v_g, v_i\} \geq 2 \) and \( \{v_j, v_g\} \geq 2 \) or \( \{v_j, v_g\} = 2 \) with an extended edge from \( v_j \) to another vertex in \( S_2 \). Note: The purpose of this path is to provide a reasonable location for finding two vertices in \( X^3_{S_2} \), since near the transition \( v_i, v_j \) there are at least 2 vertices in \( S_1 \) and at least 3 nearby vertices in \( S_1 \). There are two cases to consider.

a) There are at least three vertices in the path \( (v_j, \ldots, v_g) \).

This is the case when \( v_g \) belongs to a connected component of \( G_\Delta[S_2] \) of size three or more, or when \( v_g \) belongs to a connected component of only two vertices (with a required extended edge), but some vertices in \( (j, \ldots, g) \notin S_2 \), i.e., there are some vertices in that are in neither set along the \( \Delta \)-path. In this case, we will denote the 2nd and 3rd vertices on the path \( l, q \in S_1 \), and consider the sub-path \( (k, i, j, l, q) \). There are three cases to consider.

i) \( |N_{i,\Delta} \cap S_1| \geq 3 \): In this case \( v_i \) has at least 3 \( \Delta \)-neighbors in \( S_1 \) so \( v_i \in X^3_{S_1} \). Also, \( v_k \) has at least extended edges to each of those 3 neighbors, so \( v_k \in X^3_{S_1} \). Therefore \( |X^3_{S_1}| \geq 2 \).

ii) \( |N_{i,\Delta} \cap S_1| = 2 \): One of these neighbors is \( v_j \), lets call the other \( v_i' \), with \( v_i' \neq v_j \). Right away we can see that \( v_i \in X^3_{S_1} \), with neighbors \( \{v_j, v_i', v_j\} \) (or if \( v_j = v_i \), \( \{v_j, v_i, v_j\} \)). We know \( N_{i,\Delta} \) form a \( \Delta \)-path, so \( v_i \) must share a \( \Delta \) neighbor \( v_j \) with \( v_i \). The following cases are shown in Figure 12.

A) if the only option is \( v_i = v_i \) (i.e., \( N_{i,\Delta} \cap N_{i,\Delta} = \emptyset \), shown in Figure 12a) then there is a path in \( i \)'s \( \Delta \)-neighbors from \( v_j \) to \( v_k \) that does not include \( v_j \) (To see this note that \( v_j \) has only a single \( \Delta \)-neighbor in \( N_{i,\Delta} \) and so must form a path endpoint). There are transition vertices in this path \( v_m \in S_1 \cap N_{i,\Delta} \) and \( v_m \in S_1 \cap N_{i,\Delta} \cap N_{m,\Delta} \). \( v_m \) \( \neq v_i \) since \( v_m = v_i \) would have \( |N_{i,\Delta} \cap S_1| \geq 3 \), the previous case so \( |X^3_{S_1}| \geq 2 \).

B) \( v_i \neq v_i, v_i \in S_1 \) (Figure 12b). \( v_i \in X^3_{S_2} \) with neighbors \( \{v_j, v_i, v_j\} \) so including \( v_i \), one has \( |X^3_{S_1}| \geq 2 \).

C) \( v_i \neq v_i, v_i \in S_1 \). First consider the case that \( v_j \) has a neighbor in \( S_1 \setminus \{v_i\} \) (Figure 12c), then that neighbor and \( v_i \) are in \( X^3_{S_1} \). If we assume that neighbor doesn’t exist, consider that \( N_{i,\Delta} \) also must form a path (Figure 12d). Therefore there must be a \( \Delta \)-path in \( N_{i,\Delta} \) from \( v_j \) to \( v_j \) but since we have assumed that \( v_j \) as no \( \Delta \)-neighbors in \( S_1 \), and \( v_j \) only has two \( \Delta \)-neighbors in \( S_1 \), \( v_j' \) is the only possible \( \Delta \)-neighbor for \( j \) in \( N_{j,\Delta} \) so \( j \in N_{j,\Delta} \). The path of \( v_i \)'s neighbors must go from \( v_i \) to \( v_j \) without passing \( v_j \). This path contains \( v_i \in S_1 \), and \( v_j' \in S_1 \) there must be a path in \( N_{i,\Delta} \) from \( v_k \) to \( v_j' \) with transition nodes \( v_m \in S_1 \).
and \( v'_{m} \in S_1 \cap N_{i,\Delta} \cap N_{m',\Delta} \). We know \( v'_{m} = v'_{j} \) since \( v_{i} \) can only have the two \( \Delta \)-neighbors in \( S_1 \). Furthermore, consider that \( N_{j,\Delta} \) must also from a \( \Delta \)-path and since \( v_{i} \) is \( v'_{j} \)'s only \( \Delta \)-neighbor in \( S_1 \), all nodes in \( N_{j,\Delta} \setminus \{ v_{i} \} \in S_1 \). \( v_{i} \) cannot have any more neighbors in \( S_1 \), so there must be a path in \( S_1 \cap N_{j,\Delta} \) from \( v'_{j} \) to \( v_{i} \).

We can then see that \( v'_{m} \in X_{S_1}^3 \) with neighbors \( v'_{j}, v_{j} \) plus a node on this path to \( v_{b} \). Therefore in this case \( |X_{S_1}^3| \geq 2 \) with nodes \( v'_{m} \) and \( v_{i} \).

iii) \( |N_{i,\Delta} \cap S_1| = 1 \): (Figure 13) We will show this case is included in the previous cases via a relabeling of vertices. Though in this case, \( v_{i} \) has only a single neighbor in \( S_1 \), \( N_{j,\Delta} \) must form a \( \Delta \)-path so there must be a path in \( N_{j,\Delta} \) between \( v_{i} \) and \( v_{j} \). Since \( v_{i} \in S_1 \), \( v_{j} \in S_1 \) there must be at least one transition between sets \( v_{m} \in S_1 \cap N_{j,\Delta} \) and \( v_{n} \in S_1 \cap N_{j,\Delta} \cap N_{m,\Delta} \). Choose \( v_{m}, v_{n} \) to indicate the transition such that \( (i,...,m) \in S_1 \), and consider this new path \( (s,...,k,i,j,l,q,...,g) \). A relabeling of \( v_{i} = v_{m} \) and the vertex proceeding \( v_{m} \) in \( (i,...,m) \) as \( v_{k} \), there is a path \( \{s,...,k,i,j,l,q,...,g\} \) where \( |X_{S_1}^3| \geq 2 \) which using the previous two cases implies that \( |X_{S_1}^3| \geq 2 \).

![Fig. 12: Cases for ii) where i has two neighbors.](image)

b) There are not three vertices in the path \((v_j,...,v_g)\). In this case there must be two vertices such that \( v_j \in S_2 \) and \( v_g \in S_2 \) are a pair of \( \Delta \)-connected vertices which have a neighbor in \( S_2 \) via an extended edge. In this case we call the vertex connected with the extended edge as \( v_{r} \in N_{j,\Delta} \cap S_2 \) and assume without loss of generality that it is connected to \( v_{g} \) (there is a path in \( S_1 \cap N_{j,\Delta} \) from \( v_{i} \) to \( v_{g} \), so if the extended edge is connected to \( v_{i} \), the case is identical after a relabeling of vertices). Since \( v_{i} \) and \( v_{g} \) are connected with an extended edge, we know by the construction of \( G_{\Delta 2} \) that they must share a \( \Delta \)-neighbor, \( v_{m} \). If \( v_{m} \in S_1 \), this case is covered by the previous case with \( v_{g} = v_{m} \). So assume \( v_{m} \in S_1 \), \( v_{g} \in X_{S_1}^3 \) with neighbors \( \{j,g,r\} \). Figure 14 is for visualizing these steps. \( N_{m,\Delta} \) must form a \( \Delta \)-path and \( v_{g} \) does not connect via a \( \Delta \)-connection to \( v_{r} \), so there must be at least one intermediate node \( v_{n} \in N_{g,\Delta} \cap N_{m,\Delta} \). If \( v_{n} \in S_1 \), then there are three vertices in \( S_1 \) in a row, and that is covered in the previous case. Otherwise, if \( v_{n} \in S_1 \), \( v_{n} \in X_{S_1}^3 \) with neighbors \( \{v_{j}, v_{g}, v_{r}\} \), thus \( |X_{S_1}^3| \geq 2 \) with nodes \( v_{n}, v_{m} \).

We conclude that \( |X_{S_1}^3| \geq 2 \) in the case where both \( G_{\Delta}[S_1] \) and \( G_{\Delta}[S_2] \) have at least one connected component with 3 or more vertices, or a connected component with 2 vertices at least one of which has an in-set extended edge in \( G_{\Delta 2} \). Using the same logic we can conclude the same for \( X_{S_2}^3 \). Thus we have shown that \( |X_{S_1}^3| + |X_{S_2}^3| \geq 4 \geq 3 \) and so Condition 3 is satisfied for \( r = 3 \), \( s = 3 \).

Finally, since we have shown that for all possible choices of disjoint subsets \( S_1 \) and \( S_2 \) one of the three Conditions of Definition 1 holds for \( r = 3 \) and \( s = 3 \), we conclude that the extended graph \( G_{\Delta 2} \) is \((3,3)\)-robust.

![Fig. 13: The case where \( v_i \) only has one \( \Delta \)-neighbor in \( S_1 \) is included in the cases where \( v_i \) has two or more \( \Delta \)-neighbors in \( S_1 \) via relabeling of nodes.](image)

![Fig. 14: The case where there are only 2 vertices in the path \((v_j,...,v_g)\) and an extended edge to \( v_r \in S_2 \).](image)

**REFERENCES**

[1] R. Offati-Saber, “Distributed kalman filtering for sensor networks,” in *IEEE Conference on Decision and Control*, 2007, pp. 5492–5498.

[2] M. Schwager, M. Vittas, S. Powers, D. Rus, and C. Tomlin, “Robust adaptive coverage control for robotic sensor networks,” *IEEE Transactions on Control of Network Systems*, vol. 4, no. 3, pp. 462–476, 2015.

[3] D. Jang, J. Yoo, C. Y. Son, D. Kim, and H. J. Kim, “Multi-robot active sensing and environmental model learning with distributed gaussian process,” *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5905–5912, 2020.

[4] M. Kumar, L. Schieweckt, and M. Brockmeyer, “Efficient data aggregation middleware for wireless sensor networks,” in *2004 IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE Cat. No.04EX975)*, 2004, pp. 579–581.
[5] W. Ren and N. Sorensen, “Distributed coordination architecture for multi-robot formation control,” Robotics and Autonomous Systems, vol. 56, no. 4, pp. 324–333, 2008.

[6] A. E. Turgut, Ç. Çelikkanat, F. Gökçe, and E. Şahin, “Self-organized flocking in mobile robot swarms,” Swarm Intelligence, vol. 2, no. 2, pp. 97–120, 2008.

[7] W. Ren, R. W. Beard, and E. M. Atkins, “Information consensus in multivehicle cooperative control,” IEEE Cont. Syst., vol. 27, no. 2, pp. 71–82, April 2007.

[8] X. Su, X. Wang, and Z. Lin, “Flocking of multi-agents with a virtual leader,” IEEE transactions on automatic control, vol. 54, no. 2, pp. 293–307, 2009.

[9] X. Liu and J. S. Baras, “Using trust in distributed consensus with adversaries in sensor and other networks,” in FUSION 2014 - 17th International Conference on Information Fusion, 2014.

[10] F. Pasqualetti, A. Bicchi, and F. Bullo, “Distributed intrusion detection for secure consensus computations,” Proceedings of the IEEE Conference on Decision and Control, pp. 5594–5599, 2007.

[11] ———, “Consensus computation in unreliable networks: A system theoretic approach,” IEEE Trans. Automat. Cont., vol. 57, no. 1, pp. 90–104, 2012.

[12] H. J. LeBlanc, H. Zhang, X. Koutsoukos, and S. Sundaram, “Resilient asymptotic consensus in robust networks,” IEEE J. Select. Areas Commun., vol. 31, no. 4, pp. 766–781, April 2013.

[13] H. J. LeBlanc, H. Zhang, S. Sundaram, and X. Koutsoukos, “Consensus of multi-agent networks in the presence of adversaries using only local information,” in Proc. of the 1st international conference on High Confidence Networked Systems - HiCoNS ’12. New York, New York, USA: ACM Press, 2012.

[14] H. Zhang, E. Fata, and S. Sundaram, “Robustness of complex networks: Reaching consensus despite adversaries,” CoRR, vol. abs/1203.6119, 2012.

[15] K. Saulnier, D. Saldana, A. Prorok, G. J. Pappas, and V. Kumar, “Resilient Flocking for Mobile Robot Teams,” IEEE Robotics and Automation Letters, vol. 2, no. 2, pp. 1039–1046, 2017.

[16] D. Saldana, A. Prorok, M. F. M. Campos, and V. Kumar, “Triangular Networks for Resilient Formations,” in Springer Proceedings in Advanced Robotics, R. Groß, A. Kolling, S. Berman, E. Frazzoli, A. Martinoli, F. Matsuno, and M. Gauci, Eds. Springer International Publishing, 2018, vol. 6, pp. 147–159.

[17] X. Yu, D. Shishika, D. Saldana, and M. A. Hsieh, “Modular robot formation and routing for resilient consensus,” in Proc. American Control Conference (ACC), 2020, pp. 2464–2471.

[18] L. Guerrero-Bonilla, D. Saldana, and V. Kumar, “Dense r-robust formations on lattices,” in 2020 IEEE International Conference on Robotics and Automation (ICRA), 2020, pp. 6633–6639.

[19] D. Saldana, A. Prorok, S. Sundaram, M. F. M. Campos, and V. Kumar, “Resilient consensus for time-varying networks of dynamic agents,” in American Control Conference (ACC), 2017, pp. 252–258.

[20] A. Breitenmoser, M. Schwager, J.-C. Mettger, R. Siegwart, and D. Rus, “Voronoi coverage of non-convex environments with a group of networked robots,” in 2010 IEEE international conference on robotics and automation. IEEE, 2010, pp. 4982–4989.

[21] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, “Coverage control for mobile sensing networks,” IEEE Transactions on Robotics and Automation, vol. 20, no. 2, pp. 243–255, 2004.

[22] W. Luo and K. Sycara, “Adaptive sampling and online learning in multi-robot sensor coverage with mixture of gaussian processes,” in 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018, pp. 6359–6364.

[23] S. Kemna, J. G. Rogers, C. Nieto-Granda, S. Young, and G. S. Sukhatme, “Multi-robot coordination through dynamic voronoi partitioning for informative adaptive sampling in communication-constrained environments,” in 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2017, pp. 2124–2130.

[24] J. Hu, H. Niu, J. Carrasco, B. Lennox, and F. Arvin, “Voronoi-based multi-robot autonomous exploration in unknown environments via deep reinforcement learning,” IEEE Transactions on Vehicular Technology, vol. 69, no. 12, pp. 14413–14423, 2020.

[25] R. Olfati-Saber, J. A. Fax, and R. M. Murray, “Consensus and cooperation in networked multi-agent systems,” Proceedings of the IEEE, vol. 95, no. 1, pp. 215–233, 2007.

[26] V. Gupta, C. Langbort, and R. M. Murray, “On the robustness of distributed algorithms,” in Proceedings of the 45th IEEE Conference on Decision and Control. IEEE, 2006, pp. 3473–3478.