Improvement of network fragility for multi-robot robustness

Toru Murayama * **Lorenzo Sabattini** **

* *National Institute of Technology, Wakayama College, Wakayama, Japan (e-mail: murayama@wakayama-nct.ac.jp).*
** *University of Modena and Reggio Emilia, Reggio Emilia, Italy (e-mail: lorenzo.sabattini@unimore.it)*

Abstract: This paper proposes a novel concept of robustness for a multi-robot network. Although connectivity is a helpful indicator for robustness against failures of the robots, preserving a high connectivity limits the configuration space of the system. To relax this problem, we propose to preserve a large connected component instead of the high connectivity of the entire network, and present articulation node importance and graph fragility as indicators of the robustness. An estimation method for the node importance and a control law using the importance value are also introduced to improve the network robustness.

Keywords: Multi-robot network, Decentralized systems, Fault-tolerance, Connectivity, Distributed control,

1. INTRODUCTION

A multi-robot system is a system in which a collection of autonomous mobile robots shows some collective behaviours, that emerge from inter-robot communications. Such communication is typically achieved equipping the robots with appropriate communication means, such as some self-configuring wireless technology, thus defining a mobile ad-hoc network. Since information for the collective behaviours flows through the network, the network should be connected during their task performances in spite of robots' individual autonomous actions. For this reason, many studies proposed decentralized or distributed algorithms for preserve the network connectivity, such as Zavlanos and Pappas (2007); Schuresko and Cortés (2009); Kan et al. (2012); Sabattini et al. (2013); Cai et al. (2017). The connectivity is also a good index of network robustness against some failures since it indicates the minimum number of nodes/edges removed from a graph to make the remaining graph disconnected. Moreover, some studies proposed approaches to improve or robustify the network robustness against the failures, such as Ghedini et al. (2015, 2017); Panerati et al. (2018). In these studies, the number of 2-hop neighbors connected with only one path is employed as an indicator of node vulnerability, and a control law to move towards the barycenter of the 2-hop neighbors is proposed for improving the network robustness.

Although a multi-robot network characterized by high connectivity is robust against failures of robots, this characteristics narrows the configuration space of the system because the distance of each pair of the robots adjacent through a link is bound to the communication range. In case the system is controlled to preserve n node connectivity, each robotic node needs to keep n links at least, and this means each robot is not allowed to move away from all positions of the n robots. This represents a disadvantage in tasks where the multi-robot system is required to spread out to a given area, like a coverage task for sensing or monitoring a wide region. A trade-off is then achieved between the connectivity preservation and the desired configuration: as such, the multi-robot system may have to choose between the robustness against failures and the complete task achievement.

In order to handle the problem mentioned above, we aim to preserve a large connected component instead of preserving the connectedness of the entire network. In other words, we propose a method in which we try to preserve the overall configuration, but a small number of robots may be abandoned in the case of failures. We define the concept of fragility of a graph, introducing a novel concept of node importance to evaluate the size of the connected component that will remain after a robot failure, and propose a control law for improving the network topology when the network is too fragile. A distributed algorithm to estimate the node importance is also introduced for the control law to be executable in a distributed manner. From the proposition of this study, the system achieves a relatively large configuration space and preserves the large connected component even if one of the robots fails.

2. PRELIMINARY AND CONCEPT

In this section, we introduce some definitions and expressions as a preliminary of this study, we introduce the articulation node importance to evaluate an impact of the node failure, and we define the fragility of the network. The control objective of each the robots is also defined using the fragility.

2.1 Preliminary

Considering a connected simple graph $\mathcal{G}$, we introduce the adjacency matrix A of a graph $\mathcal{G}$. We also define a node set $\mathcal{V}$ of a graph $\mathcal{G}$ and a cardinal number $V = |\mathcal{V}|$ of a node set $\mathcal{V}$ (i.e. the number of nodes in a graph $\mathcal{G}$). A neighbor node set of node i is described as $\mathcal{N}_i$. An articulation node is defined as a node whose removal makes graph $\mathcal{G}$ disconnected. Removing an articulation node i brings two or more connected components as a result. We define node sets of such connected components by $\mathcal{C}_m,\ m = 1, \ldots, r$, where r denotes the number of the connected components. The number r of the connected components is less than or equal to the number $|\mathcal{N}_i|$ of the neighbors. Also define a cardinal number $C_m = |\mathcal{C}_m|$ of a connected component $\mathcal{C}_m$. Clearly we can see $\bigcup_m \mathcal{C}_m = \mathcal{V} \setminus \{i\}$ and $\sum_m C_m = V - 1$ from the definition.

In order to evaluate a graph $\mathcal{G}$ after a node i has been removed, we consider a perturbed graph $\mathcal{G}(\varepsilon)$ in which link weights connected to the node i are perturbed. We define a perturbed adjacency matrix $A(\varepsilon)$ multiplying a small positive value ε by all entries a_{ij} in the i-th row and a_{ji} in the i-th column of an (ordinal) adjacency matrix A with respect to a graph $\mathcal{G}$. And also we define a perturbed Laplacian matrix $L(\varepsilon)$ from the perturbed adjacency matrix $A(\varepsilon)$. The second smallest eigenvalue of a perturbed Laplacian matrix $L(\varepsilon)$ is expressed as $\lambda(\varepsilon)$, and the eigenvector $v(\varepsilon)$ corresponding to the second smallest eigenvalue $\lambda(\varepsilon)$ is called as perturbed Fiedler vector in this study. Note that a perturbed Fiedler vector $v(\varepsilon)$ is orthogonal to the vector $\mathbf{1}$ which is the eigenvector of a Laplacian matrix with respect to the smallest eigenvalue 0.

2.2 Importance of articulation node

Our concern is the size of remaining connected components $\mathcal{C}_m$ after removing a node i from the given graph $\mathcal{G}$. In fact, having a (remaining) larger connected component is useful to continue some tasks of a multi-robot team, while scattering into smaller connected components should be avoided, as discussed in the introduction section.

From the above discussion, we define the impact of an articulation node i removal as

$$a_i = \frac{\sum_m C_m - C_M}{C_M}, \tag{1}$$

where $C_M = \max_m C_m$ is the size of largest connected component. This value a_i represents the fraction of the number of nodes which will be lost from the largest components. Along these lines, we define $a_i = 0$ if the node i is not an articulation one, as a case when no node will be lost. For instance, the value $a_i = 1$ means the largest connected component is almost a half of the former graph $\mathcal{G}$ (strictly $C_M = (V-1)/2$), and therefore $a_i > 1$ means removal of the node i causes 3 or more connected components whose sizes are less than a half of the former graph. Clearly the value $a_i + 1$ implies a fraction $V - 1$ of the former graph, excluding the node i, and the size C_M of the largest connected component, from the definition. Hereafter we refer to the value a_i as the importance of node i.

The size C_m of the connected components is also useful for calculation of the betweenness centrality, defined by

$$b_i = \sum_{j,k \in \mathcal{V} \setminus \{i\}} \frac{s_{jk}(i)}{s_{jk}}, \tag{2}$$

where s_{jk} is the total number of paths between node j and k, and $s_{jk}(i)$ is the number of paths between node j and k through the node i. Assuming the node i is articulation one, we get

$$b_i = \sum_{m \neq l} C_m C_l, \tag{3}$$

therefore we can calculate the centrality from the sizes of the components.

However, the betweenness centrality does not match our purpose since we are particularly interested in the size of the largest connected component and the betweenness centrality depends on the topology other than the largest connected component. For instance, assume two example cases,

(1) Case 1: the removal of the node i generates $r = 2$ connected components $C_1 = V - 1 - n$ and $C_2 = n$ where $(V-1)/2 \geq n$
(2) Case 2: the removal of the node i generates $r = n+1$ connected components $C_1 = V - 1 - n$ and $C_2 = \cdots = C_{n+1} = 1$.

Despite the invariance of the size of the largest component, the betweenness centrality differs in each cases, as summarized in Table 1. Moreover, the betweenness centrality takes a same value even though the size of the largest connected component differs (e.g. $V = 7$, in case 1 with $n = 3$ and in case 2 with $n = 2$). That is the reason why we employ the importance a_i as an indicator of each nodes in this study.

Table 1. Importance and centrality in the considered examples

	importance a_i	betweenness centrality b_i
Case 1	$n/(V-1-n)$	$n(V-1-n)$
Case 2	$n/(V-1-n)$	$n(V-1-n) + n(n-1)/2$

2.3 Concept of fragility

Using the node importance a_i, we define the measurement of graph fragility by

$$F(\mathcal{G}) = \max_{i \in \mathcal{V}} a_i, \tag{4}$$

which represents the fraction of nodes in the largest component after a robot loss, i.e., when any one of the robots fails. Then our objective in this study is to reduce the network fragility $F(\mathcal{G})$ with given threshold T, that is, to move the robots to satisfy $F(\mathcal{G}) < T$. In the latter sections, we introduce strategies to achieve this objective.

3. IMPORTANCE ESTIMATION

In this section, we describe an estimation method for the node importance a_i. First we introduce some properties of the perturbed Fiedler vector, which will be instrumental for the proposed estimation method. The estimation procedure is, in fact, described introducing algorithms for estimating the Fiedler vector and algorithms for classification. The computation of the node importance is defined considering both the following cases: (1) if all the entries

$v_{j\in\mathcal{V}}$ are available, and (2) if only the local entries $v_{j\in\mathcal{N}_i}$ are available.

3.1 Properties of the perturbed Fiedler vector

We introduce some properties of the perturbed Fiedler vector $v(\varepsilon)$, as proven in Murayama (2018), for the importance estimation method detailed in the following subsection.

Theorem 1. As $\varepsilon \to +0$, the perturbed Fiedler vector $v(\varepsilon)$ approaches vector v which satisfies

$$\dot{\lambda} v_i = \sum_{n\in\mathcal{N}_i} (v_i - v_n) = \sum_{j\in\mathcal{V}} \frac{d_{(j)}}{C_{(j)}} (v_i - v_j), \tag{5}$$

$$\dot{\lambda} v_j = \frac{d_{(j)}}{C_{(j)}} (v_j - v_i), \ {}^\forall j \in \mathcal{V} \setminus \{i\}, \tag{6}$$

$$v_j = v_k, \ {}^\forall j, k \in \mathcal{C}_m, \tag{7}$$

where $\dot{\lambda} = \lim_{\varepsilon\to+0} \lambda(\varepsilon)/\varepsilon$, and v_i is the i-th entry of the vector v. $C_{(j)}$ denotes the cardinal number of the connected component $\mathcal{C}_m$ such that $j \in \mathcal{C}_m$, and $d_{(j)}$ denotes the number of nodes which are adjacent to the node i in the connected component $\mathcal{C}_m$ including the node j, that is, $d_{(j)} = |\{n \in \mathcal{N}_i \cap \mathcal{C}_m\}|$. ◇

From this theoretical fact, we can see that, if $v_j \neq v_k$, then node j and node k are not in the same connected component. As a special case, we can find the below fact.

Corollary 2. Suppose the elimination of the node i divides the graph $\mathcal{G}$ into exactly two connected components ($r = 2$). Then, the equation $v_j = v_k$ is satisfied if and only if both nodes j and k are in the same connected component $\mathcal{C}_m$. ◇

This fact indicates that, if the number of neighbors $\mathcal{N}_i$ is 2, then the size C_m of both connected components can be computed from the entries of the perturbed Fiedler vector. In other cases, i.e. when $r \geq 3$, there may exist indexes $j \neq k$ satisfying both $v_j = v_k$ and $\mathcal{C}_{(j)} \neq \mathcal{C}_{(k)}$ due to $d_{(j)}/C_{(j)} = d_{(k)}/C_{(k)}$: therefore, the sizes C_m of connected components may not be computed from the values v_j of the perturbed Fiedler vector. To avoid this trouble, it is possible to use the perturbed Fiedler vector of a weighted Laplacian matrix instead of a standard Laplacian matrix.

3.2 Estimation algorithm

The overview of the main steps of the proposed estimation procedure is summarized below.

(1) Compute the perturbed Fiedler vector $v_i(\varepsilon)$ corresponding to node i using a sufficiently small perturbation parameter $\varepsilon > 0$.
(2) Classify the entries v_j of the perturbed Fiedler vector $v(\varepsilon)$.
(3) Calculate the importance a_i according to the entries and the classes.

First, compute the perturbed Fiedler vector $v(\varepsilon)$ exploiting any algorithm available in the literature: it is worth noting that several algorithms can be found in the literature for distributed computation of the eigenvectors of the Laplacian matrix, so we can exploit any of them. Here we categorize the algorithms into two types:

- Algorithms of the first type (e.g. Gusrialdi and Qu (2017); Zareh et al. (2018)) can compute the entire eigenvector $v(\varepsilon)$ in a distributed fashion, and therefore all the entries $v_{j\in\mathcal{V}}$ are available for the node i.
- With algorithms of the second type (e.g. Bertrand and Moonen (2013)), the node i computes only its own entry v_i and then only the local entries $v_{j\in\mathcal{N}_i}$ are available for the node i.

In case all the entries $v_{j\in\mathcal{V}}$ of the perturbed Fiedler vector are available, the node i classifies the entries v_j (except for v_i) into each class $\mathcal{C}_m$. Since the number of classes is unknown by the node i, some classification algorithms not using class number (like Pelleg and Moore (2000)) will work well, and we also have proposed an algorithm for the connected component classification in Murayama (2018). If $v_j \neq 0$ and the eigenvalue $\lambda(\varepsilon)$ is available in the eigenpair estimator, the class size formula

$$C_{(j)} = \frac{d_{(j)}}{\dot{\lambda}} \frac{v_j - v_i}{v_j}, \tag{8}$$

with $\dot{\lambda} \simeq \lambda(\varepsilon)/\varepsilon$ is useful to evaluate the classification error, as shown in our previous study. The differentiated eigenvalue $\dot{\lambda}$ can be calculated from

$$\dot{\lambda} = \frac{\sum_{n\in\mathcal{N}_i} (v_i - v_n)}{v_i}, \tag{9}$$

if $v_i \neq 0$, derived from (5). The cardinal number $|\mathcal{C}_m|$ of each class directly indicates the number C_m of nodes in the connected component, therefore the importance value a_i is directly computed from the definition (1).

In case only the local entries $v_{j\in\mathcal{N}_i}$ are available, the node i classifies them into the neighbor classes $\mathcal{D}_m$. Once the classes $\mathcal{D}_m$ have been classified, the importance value a_i is computed in the same manner as in the previous case. Even if $v_i = 0$, a ratio between two connected components $C_{(j)}$ and $C_{(k)}$ where $j \in \mathcal{D}_m$, $k \in \mathcal{D}_l$, and $\mathcal{D}_m \neq \mathcal{D}_l$ can be computed as

$$\frac{C_{(k)}}{C_{(j)}} = \frac{d_{(k)} v_j (v_k - v_i)}{d_{(j)} v_k (v_j - v_i)}, \tag{10}$$

with $v_j \neq 0$ and $v_k \neq 0$, derived from (6). The label of the largest connected component $M = \arg\max C_m$ is found from the magnitude relationships, then the importance a_i is computed from

$$a_i = \sum_m \frac{C_m}{C_M} - 1. \tag{11}$$

In case there are some entries such that $v_j = 0$, the computations of (8) or (10) do not work well. However, some racks of information about C_m may be resumed from the fact $\sum_m C_m = V - 1$, because it is enough for the calculation of the importance a_i to find the size C_M of the largest component.

4. NETWORK IMPROVEMENT

In this section we propose a method to improve the network robustness according to the importance value. Here we design the control law for letting articulation nodes with highest importance to change their configuration to non-articulation nodes.

We consider robotic nodes with the following single integrator dynamics

$$\dot{p}_i = u_i, \tag{12}$$

where p_i and u_i denote the position and the control input of the robot i respectively. The previous studies in Ghedini et al. (2017) proposed a control law for network improvement, described as

$$u_i = \phi u_i^c + \psi u_i^r, \tag{13}$$

where u_i^c is the connectivity maintenance control law, u_i^c is the control law for robustness improvement, and coefficients ϕ, ψ are design parameters that represent control gains.

The connectivity control law u_i^c is defined as the gradient of an energy function, as

$$u_i^c = -\frac{\partial V(\lambda_W)}{\partial p_i} = -\frac{\partial V(\lambda_W)}{\partial \lambda_W}\frac{\partial \lambda_W}{\partial p_i}, \tag{14}$$

where $V(\cdot) \geq 0$ is an energy function for the connectivity maintenance, and λ_W denotes the second smallest eigenvalue of a weighted Laplacian matrix L_W induced by a weighted adjacency matrix W. Examples of the energy function and the elements w_{ij} of the weighted adjacency matrix introduced in Ghedini et al. (2017) are given as

$$V(\lambda_W) = \begin{cases} \coth(\lambda_W - \epsilon), & \text{if } \lambda_W > \epsilon, \\ 0, & \text{otherwise,} \end{cases} \tag{15}$$

$$w_{ij} = \begin{cases} \exp(-\frac{\|p_i - p_j\|^2}{2\sigma^2}), & \text{if } \|p_i - p_j\| \leq R \text{ and } i \neq j, \\ 0, & \text{otherwise,} \end{cases} \tag{16}$$

where $\epsilon > 0$ is a designed lower-bound of the weighted connectivity, $R > 0$ is the upper-bound of communication range, and $\sigma > 0$ is a scaling factor.

In the previous studies, the robustness improving control law u_i^r is designed to move a node towards the barycenter of its weakly connected 2-hop neighbors' positions. Even though the control objective of this study differs from the previous ones, here we employ a similar approach, such that each robot heads to the barycenter of some neighbors: the approach tends to get a position consensus, and therefore the network robustness is expected to improve.

The control law for the robustness improvement in this study is defined by

$$u_i^r = \begin{cases} \frac{\sum_{n \in \mathcal{N}_i} \alpha(a_i, a_n)(p_n - p_i)}{\sum_{n \in \mathcal{N}_i} \alpha(a_i, a_n)}, & \text{if } \sum_{n \in \mathcal{N}_i} \alpha(a_i, a_n) \neq 0, \\ 0, & \text{otherwise,} \end{cases} \tag{17}$$

where $\alpha(\cdot)$ is a weight function to reflect the threshold T of the fragility, for instance

$$\alpha(a_i, a_n) = a_i[a_i \geq T] + a_n[a_n \geq T], \tag{18}$$

where $[X]$ denotes the Iverson bracket such that $[X] = 1$ if X is true and $[X] = 0$ otherwise (detailed in Graham et al. (1994)). This control law implies that the robot i moves towards the barycenter of its high important neighbors' positions p_n, considering the weights $\alpha(a_i, a_n)$. By executing this control law for all robots, each neighbor converges near the high important robots $a_i \geq T$: as a consequence, it is expected that the fragility $F(\mathcal{G})$ is improved by creating new links connecting these important nodes to their neighbors, thus leading the importance $a_i = 0$. Because each robot neighboring the high importance ones moves according to a consensus-like algorithm with a weighted graph induced by the weight $\alpha(\cdot)$, the robots continue to gather until the high importance robots become low importance ones.

From the control law (13), each robot moves to improve the node importance while preserving the weighted connectivity, whenever the network of the system has high fragility. As the result, the system has the larger configuration space than that of the system with the high connectivity, and will preserve a large connected component even if one of the robots fails.

5. EXPERIMENT

In this section, we show some results of our robustness concept. Assume that each the robots is equipped with a sensor which can detect something on a disk $\mathcal{D}(p_i, r)$, centered in the robot position p_i and with radius r. Since the network may not be connected due to the failure of robots, the coverage area of the robotic team is defined by $\mathcal{A} = \bigcup_{i \in \mathcal{C}_M} \mathcal{D}(p_i, r)$ where $\mathcal{C}_M$ denotes the largest connected component. Thus, the task of the system is to keep the coverage area $\mathcal{A}$ as large as possible, in the risk of robot failure.

Some results of an experimental example with the proposed control law are shown in Figs. 1 and 2. The experiment were performed on the Robotarium, a swarm robotics testbed by Georgia Tech, described in Pickem et al. (2017). In this example, $V = 10$ robots moved in order to make the network fragility $F(\mathcal{G})$ go below the threshold $T = 0.25$. The robots controlled the network topology from the initial one (blue lines) to the final one (red lines) as shown in Fig. 1. We can see from Fig. 1 that the robots converged for the improvement of the high importance robots, while the low importance robots remained the articulation ones. This fact can also be seen in Fig. 2, which shows the time variation of the fragility $F(\mathcal{G})$ and each importance a_i. The high importance ($a_i \geq T$) robots were improved over time, while the low importance robots remained unchanged, therefore the fragility was improved, achieving $F(\mathcal{G}) < T$ as the result.

The relation between the coverage area $\mathcal{A}$ and the threshold T when a single robot fails is shown in Fig. 3. The result is derived from numerical simulations with the sensing radius $r = 0.5$, and the other conditions are identical to the experiment described in the preceding paragraph. Note that the threshold $T > 4/5$ maintains the initial graph shown in Fig. 1, and the threshold $T \leq 1/8$ brings a 2-connected graph that we can get from the traditional robustification method. From the result, we can find that the minimum coverage area with the threshold $T \leq 3/6$ is larger. Suppose the critical robot fails with probability p and each the other robot fails with probability $(1-p)/(V-1)$, then the expected coverage area $\mathbb{E}(|\mathcal{A}|)$ is described as Fig. 4. It shows the expected area with the threshold $T \in (2/7, 3/6]$ is the maximum when the probability p is around 0.55. This fact indicates that the robustness concept we discussed in this paper is meaningful in the sense of the coverage.

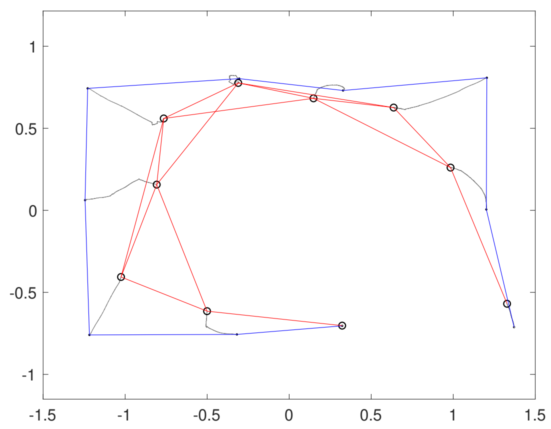

Fig. 1. Trajectories of each robot (black lines) and network topology (initial: blue lines; final: red lines)

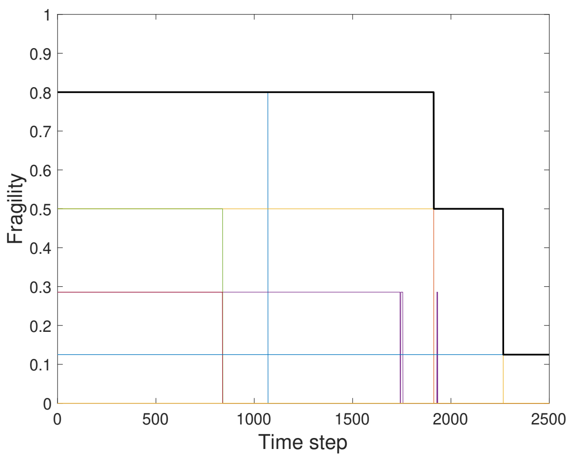

Fig. 2. Network fragility (black line) and importance of each robot

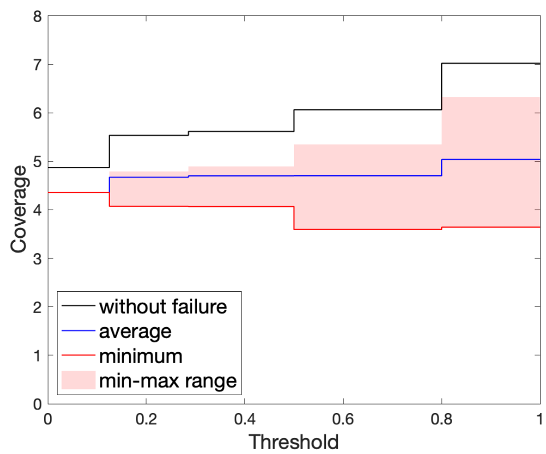

Fig. 3. Coverage area without robot failure (black line), average (blue line) and minimum (red line) in case single robot fails

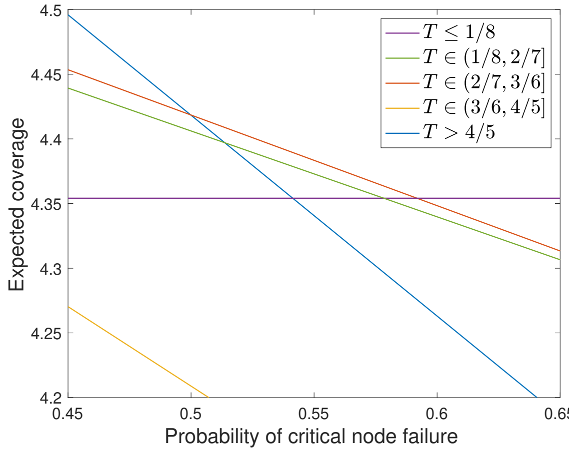

Fig. 4. Expected coverage area vs. robot failure probability with several thresholds

6. CONCLUSION

In this paper, we propose a novel concept about network fragility for the robustness of a multi-robot network against a robot failure. The concept of the articulation node importance and the graph fragility are introduced, then an estimation algorithm for the node importance and a control law to improve the graph fragility are proposed. The appropriateness of each the concepts and the methods are theoretically shown, and preliminary valuation is provided.

REFERENCES

Bertrand, A. and Moonen, M. (2013). Distributed computation of the Fiedler vector with application to topology inference in ad hoc networks. *Signal Processing*, 93(5), 1106–1117. doi:10.1016/j.sigpro.2012.12.002.

Cai, D., Wu, S., and Deng, J. (2017). Distributed Global Connectivity Maintenance and Control of Multi-Robot Networks. *IEEE Access*. doi: 10.1109/ACCESS.2017.2708422.

Ghedini, C., Ribeiro, C., and Sabattini, L. (2017). Toward fault-tolerant multi-robot networks. *Networks*, 70(4), 388–400. doi:10.1002/net.21784.

Ghedini, C., Secchi, C., Ribeiro, C.H., and Sabattini, L. (2015). Improving robustness in multi-robot networks. *IFAC-PapersOnLine*. doi:10.1016/j.ifacol.2015.12.011.

Graham, R.L., Knuth, D.E., and Patashnik, O. (1994). *Concrete Mathematics: A Foundation for Computer Science*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition.

Gusrialdi, A. and Qu, Z. (2017). Distributed Estimation of All the Eigenvalues and Eigenvectors of Matrices Associated With Strongly Connected Digraphs. *IEEE Control Systems Letters*, 1(2). doi: 10.1109/LCSYS.2017.2717799.

Kan, Z., Dani, A.P., Shea, J.M., and Dixon, W.E. (2012). Network connectivity preserving formation stabilization and obstacle avoidance via a decentralized controller. *IEEE Transactions on Automatic Control*, 57(7), 1827–1832. doi:10.1109/TAC.2011.2178883.

Murayama, T. (2018). Articulation Node Importance Estimation and Its Correctness for Robustness of

Multi-robot Network. *IFAC-PapersOnLine*. doi: 10.1016/j.ifacol.2018.11.536.

Panerati, J., Minelli, M., Ghedini, C., Meyer, L., Kaufmann, M., Sabattini, L., and Beltrame, G. (2018). Robust connectivity maintenance for fallible robots. doi: 10.1007/s10514-018-9812-8.

Pelleg, D. and Moore, A. (2000). X-means: Extending k-means with efficient estimation of the number of clusters. In *Proceedings of the 17th International Conf. on Machine Learning*, 727–734. Morgan Kaufmann.

Pickem, D., Glotfelter, P., Wang, L., Mote, M., Ames, A., Feron, E., and Egerstedt, M. (2017). The robotarium: A remotely accessible swarm robotics research testbed. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 1699–1706. doi: 10.1109/ICRA.2017.7989200.

Sabattini, L., Secchi, C., Chopra, N., and Gasparri, A. (2013). Distributed control of multirobot systems with global connectivity maintenance. *IEEE Transactions on Robotics*, 29(5). doi:10.1109/TRO.2013.2267971.

Schuresko, M. and Cortés, J. (2009). Distributed motion constraints for algebraic connectivity of robotic networks. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 56(1-2), 99–126. doi: 10.1007/s10846-009-9328-8.

Zareh, M., Sabattini, L., and Secchi, C. (2018). Distributed Laplacian Eigenvalue and Eigenvector Estimation in Multi-robot Systems. In *Distributed Autonomous Robotic Systems*, 191–204.

Zavlanos, M.M. and Pappas, G.J. (2007). Distributed connectivity control of mobile networks. *Proceedings of the IEEE Conference on Decision and Control*, 24(6), 3591–3596. doi:10.1109/CDC.2007.4434525.