The Unbroken Telephone Game: Keeping Swarms Connected

Extended Abstract

Vivek Shankar Varadharajan
Ècole polytechnique de Montrèal
Montréal, Québec
vivek-shankar.varadharajan@polymtl.ca

Bram Adams
Ècole polytechnique de Montrèal
Montréal, Québec
bram.adams@polymtl.ca

Giovanni Beltrame
Ècole polytechnique de Montrèal
Montréal, Québec
giovanni.beltrame@polymtl.ca

ABSTRACT

Connectivity maintenance plays a key role in achieving a desired global behaviour among a swarm of robots. Yet, lack of computation resources, low communication bandwidth, robot failures, and unstable links are tough challenges for connectivity maintenance in realistic environments. In this paper, we propose a novel decentralized connectivity-preserving algorithm that can be deployed on top of other behaviours to enforce connectivity constraints. The algorithm takes a set of targets to be reached while keeping a minimum number of redundant links between robots, with the goal of guaranteeing bandwidth and reliability. We empirically study the performance of the algorithm, analyzing its time to convergence and robustness to failure.

KEYWORDS

Swarm robotics; Connectivity Maintenance; Fault-Tolerance

ACM Reference Format:
Vivek Shankar Varadharajan, Bram Adams, and Giovanni Beltrame. 2019. The Unbroken Telephone Game: Keeping Swarms Connected. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019*, IFAAMAS, 3 pages.

1 INTRODUCTION

Swarm robotics is a field of engineering that deals with relatively simple physical agents to achieve a global behaviour that emerges as a result of local interactions [15]. Swarm robotics has been widely investigated in the last decade [1] for a number of different applications, mainly due to its inherent benefits: robustness, scalablity, and flexibility. With a large swarm, in general, the loss of a single agent does not jeopardize the overall mission and a failed agent could be replaced with another. Hence, robotic swarms are deemed cost effective solutions when dealing with large, spatially distributed tasks like exploration [8], search and rescue [16], and area coverage [5].

In many of these applications the robots need communication between each other to coordinate. For the information to propagate, the swarm needs to be *connected*, i.e., there has to be a communication path between all robots of the swarm. The problem of *maintaining connectivity* is widely discussed in literature, with a number of different recent approaches [3, 7, 10, 11]. Some of these approaches design control strategies to enforce algebraic connectivity [2] among a group of connected nodes [3, 4, 9, 14], while others address the connectivity problem by enforcing virtual forces on a pre-existent structure [6, 17, 19].

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada.

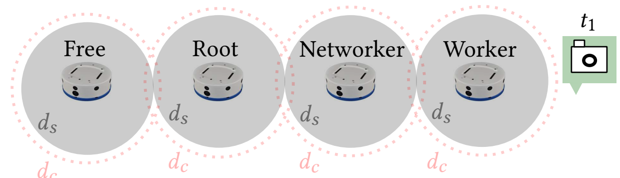

Figure 1: Illustration of chain formation algorithm.

During a real-world mission, robots can fail for a number of reasons (environmental factors, wear and tear, etc.) and break connectivity, compromising the mission objectives. Consider the Fukushima accident in 2011: robots were used to inspect the collapsed nuclear power plant (with a video feed), and they were subject to extremely high failure rates due to radiation. In addition, maintaining connectivity is likely not to be the only requirement for mission success. Panerati et al. [11] consider a robustness factor into the designed control law to tackle robot failures while enforcing connectivity. Yet, convergence is slow due to the computation of the Fiedler vector (an algebraic connectiviy measure) using a power iteration method, which requires multiple information exchanges throughout the entire swarm. In this paper, we propose a decentralized, failure-tolerant connectivity maintenance approach that can be added to existing control algorithms. Our approach is lightweight in both communication and computation requirements, freeing resources to achieve the mission goals. In practice, we progressively and dynamically use the robots in the swarm to form a communication backbone from a root to a set of targets. We set the number of links between the robots as a configurable factor.

2 APPROACH

In this work, the robots in the swarm are self organizing, with a global behaviour emerging as a result of local interactions among robots. We assume that the robots are randomly deployed, not necessarily in a fully connected topology. The desired global behavior of the swarm is to construct a tree from a central reference robot (aka root) to the robots visiting one or more target (workers), using minimal number of inter-connecting robots (networkers), as illustrated in Figure 1. Robots are organized in a parent-child relationship starting from the root and going towards the target. For this purpose, we build a communication chain for the robots visiting each target t_i from the target set T. We specify a target t_i by its position, orientation, and required number of links. At first, we select the root and workers. A free robot can either switch to be a networker or a worker depending on the immediate need of the swarm. Once the workers and the root are selected, the worker extends the communication chain starting from the root. When a worker determines it has reached a threshold distance d_s from the root, the robot chooses a free robot as a networker to act as relay to the root. Subsequently, when a networker reaches a suitable distance, it selects a new free robot to serve as a networker, and so on, forming a chain until the worker reaches the desired target. To tackle intermediate robot failures and disconnections, robots exchange all the robot IDs and last known locations in a chain. In case of an intermediate robot failure, the robots move to close the gap using its last known location. One way to model the interaction between the robots in a chain from a worker to the root is by using the notion of virtual springs [21]. Each robot exerts a virtual force (F^s_{ij}) on its neighbors to stay within a safe communication distance (d_s). The exerted force is:

$$F^s_{ij} = k(d_{ij} - d_s) \quad (1)$$

where k is the spring stiffness constant, d_{ij} is the distance between the robot i and robot j. d_s is the length of the spring, which defines the safe communication distance between the agents.

Worker robots are assumed to be the robots visiting a target at a distant location. The control input of the worker robot u^w_i is formulated as a sum of virtual forces:

$$u^w_i = \begin{cases} u^{sd}_i + f(d^p_i)(u^t_i + u^o_i), & \text{if } d^p_i < d_c, \forall p \in P_i \\ u^p_i, & \text{otherwise} \end{cases} \quad (2)$$

$$u^{sd}_i = \sum_{\forall j \in P_i} u^{sd}_{ij} \quad (3)$$

where $f(d^p_i) = 1$ if $d^p_i < d_s$ or 0 otherwise, and $u^{sd}_{ij} \propto F^s_{ij}$ as in Equ. 1 and Equ. 3 define the sum of this force over all neighbors in the direction of the root. u^t_i correspond to the control input attracting a robot towards a target and the resultant force is computed using the Equ. 1, with distance to target replacing inter-robot distance. u^o_i defines the control velocity that results from a repulsive potential created by obstacles, so as to avoid the obstacles, as described in [18]. d_c is the critical communication distance above which communication becomes unreliable and eventually results in a broken link. Let P_i be the set of parents of robot i connecting i to the root robot, either directly or through other robots acting as communication relay. If the distance between the parent and the child (d^p_i) increases above a critical communication distance d_c, the robot performs an emergency maneuver towards the parent using the virtual force created by u^p_i, resulting in a chain retraction towards the root.

Networker robots maintain a certain distance from their neighbors to establish a communication relay. The control law maintaining the integrity of the networker position in the chain is:

$$u^w_i = \begin{cases} u^p_i, & \text{if } d^p_i >= d_c, \forall p \in P_i \\ u^{sd}_i + f(d^{pc}_i)(u^o_i), & \text{otherwise} \end{cases} \quad (4)$$

$$u^{sd}_i = u^{sd}_p + u^{sd}_c \quad (5)$$

where u^p_i is the force that attracts a networker in the direction of the root, if the distance is over the critical distance d_c, as in Equ. 4, retracts the chain towards the root; u^{sd}_i is the control law that ensures the networker is positioned between parent and child; and d^c_i is the distance between robot i and child c.

$$f(d^{pc}_i) = \begin{cases} 1, & \text{if } d^p_i < d_s \text{ or } d^c_i < d_s \\ 0, & otherwise \end{cases} \quad (6)$$

Free robots form a cluster around the root using the virtual force created by the Lennard-Jones (LJ) potential [20], waiting to be selected to serve as a relay by a networker or a worker.

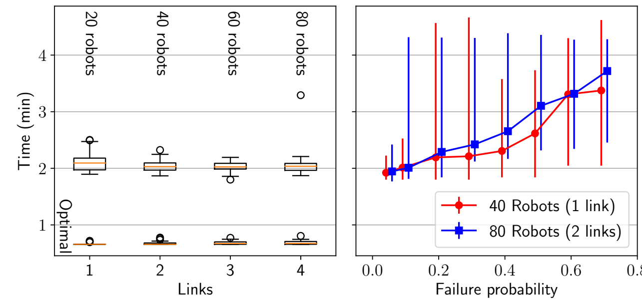

Figure 2: Time to build chains with 20, 40, 60 and 80 robots without failures (left); time to build chains with 40 and 80 robots and with varying failure probability (right).

3 EXPERIMENTS

We conducted an experimental evaluation using a realistic physics-based multi-robot simulator, ARGoS [13], and Buzz [12] to implement the robot controllers and the robots were using a double integrator model to exert the virtual forces. We performed two sets of large scale simulations, studying the time to build the chain: with and without the introduction of failures. In particular, Figure 2 reports the time needed by the swarm to reach the targets while maintaining a connected network with the desired number of links, over 35 trials, without failure (left) and with failure (right). The figure also reports the optimal time required by robots to navigate to chain assigned locations (assuming they all know the optimal locations in advance). We can observe from Figure 2 (left) that the robots forming different number of links consumed almost equivalent amount of time to build the chain, over different trials, from which the effect of a distributed algorithm is evident.

4 CONCLUSIONS

We propose a distributed approach to enforce connectivity constraints, capable of working alongside an existing algorithm, with low computational and communication requirements. The algorithm progressively builds a communication backbone for a set of robots visiting a distant target. Our approach is self-organizing and inherently robust to single agent failure. We tackle agent failures by propagating a minimal amount of information through the communication backbone. We studied the performance of the proposed algorithm through an initial set of simulation experiments that empirically demonstrate the properties of the proposed algorithm in terms of time to convergence, robustness to failure, and scalability to up to hundreds of agents.

REFERENCES

[1] Manuele Brambilla, Eliseo Ferrante, Mauro Birattari, and Marco Dorigo. 2013. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence* 7, 1 (01 Mar 2013), 1–41.

[2] Miroslav Fiedler. 1973. Algebraic connectivity of graphs. *Czechoslovak mathematical journal* 23, 2 (1973), 298–305.

[3] Andrea Gasparri, Lorenzo Sabattini, and Giovanni Ulivi. 2017. Bounded control law for global connectivity maintenance in cooperative multirobot systems. *IEEE Transactions on Robotics* 33, 3 (2017), 700–717.

[4] Cinara Ghedini, Carlos HC Ribeiro, and Lorenzo Sabattini. 2016. Improving the fault tolerance of multi-robot networks through a combined control law strategy. In *Resilient Networks Design and Modeling (RNDM), 2016 8th International Workshop on*. IEEE, 209–215.

[5] Luca Giuggioli, Idan Arye, Alexandro Heiblum Robles, and Gal A. Kaminka. 2018. *From Ants to Birds: A Novel Bio-Inspired Approach to Online Area Coverage*. Springer International Publishing, Cham, 31–43.

[6] Dominik Krupke, Maximilian Ernestus, Michael Hemmer, and Sándor P Fekete. 2015. Distributed cohesive control for robot swarms: Maintaining good connectivity in the presence of exterior forces. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 413–420.

[7] Nathalie Majcherczyk, Adhavan Jayabalan, Giovanni Beltrame, and Carlo Pinciroli. 2018. Decentralized Connectivity-Preserving Deployment of Large-Scale Robot Swarms. *arXiv preprint arXiv:1806.00150* (2018).

[8] Sandeep Manjanna, Alberto Quattrini Li, Ryan N Smith, Ioannis Rekleitis, and Gregory Dudek. 2018. Heterogeneous Multi-Robot System for Exploration and Strategic Water Sampling. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 1–8.

[9] Marco Minelli, Marcel Kaufmann, Jacopo Panerati, Cinara Ghedini, Giovanni Beltrame, and Lorenzo Sabattini. 2018. Stop, think, and roll: Online gain optimization for resilient multi-robot topologies. *arXiv preprint arXiv:1809.07123* (2018).

[10] Jacopo Panerati, Luca Gianoli, Carlo Pinciroli, Abdo Shabah, Gabriela Nicolescu, and Giovanni Beltrame. 2018. From swarms to stars: Task coverage in robot swarms with connectivity constraints. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 7674–7681.

[11] Jacopo Panerati, Marco Minelli, Cinara Ghedini, Lucas Meyer, Marcel Kaufmann, Lorenzo Sabattini, and Giovanni Beltrame. 2018. Robust connectivity maintenance for fallible robots. *Autonomous Robots* (2018), 1–19.

[12] Carlo Pinciroli and Giovanni Beltrame. 2016. Buzz: An Extensible Programming Language for Heterogeneous Swarm Robotics. In *International Conference on Intelligent Robots and Systems*. IEEE, 3794–3800.

[13] Carlo Pinciroli, Vito Trianni, Rehan O'Grady, Giovanni Pini, Arne Brutschy, Manuele Brambilla, Nithin Mathews, Eliseo Ferrante, Gianni Di Caro, Frederick Ducatelle, Mauro Birattari, Luca Maria Gambardella, and Marco Dorigo. 2012. ARGoS: a Modular, Parallel, Multi-Engine Simulator for Multi-Robot Systems. *Swarm Intelligence* 6, 4 (2012), 271–295.

[14] Lorenzo Sabattini, Nikhil Chopra, and Cristian Secchi. 2013. Decentralized connectivity maintenance for cooperative control of mobile robotic systems. *The International Journal of Robotics Research* 32, 12 (2013), 1411–1423.

[15] Erol Şahin. 2005. Swarm Robotics: From Sources of Inspiration to Domains of Application. In *Swarm Robotics*, Erol Şahin and William M. Spears (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 10–20.

[16] Carlos Sampedro, Alejandro Rodriguez-Ramos, Hriday Bavle, Adrian Carrio, Paloma de la Puente, and Pascual Campoy. 2018. A Fully-Autonomous Aerial Robot for Search and Rescue Applications in Indoor Environments using Learning-Based Techniques. *Journal of Intelligent & Robotic Systems* (03 Jul 2018).

[17] Michael Schuresko and Jorge Cortes. 2012. Distributed tree rearrangements for reachability and robust connectivity. *SIAM Journal on Control and Optimization* 50, 5 (2012), 2588–2620.

[18] Mohammadali Shahriari, Ivan Svogor, David St-Onge, and Giovanni Beltrame. 2018. Lightweight Collision Avoidance for Resource-Constrained Robots. In *Intelligent Robots and Systems (IROS), 2018 IEEE/RSJ International Conference on*. IEEE.

[19] Touraj Soleymani, Emanuele Garone, and Marco Dorigo. 2015. Distributed constrained connectivity control for proximity networks based on a receding horizon scheme. In *American Control Conference (ACC), 2015*. IEEE, 1369–1374.

[20] Loup Verlet. 1967. Computer" experiments" on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules. *Physical review* 159, 1 (1967), 98.

[21] Jakub Wiech, Victor A. Eremeyev, and Ivan Giorgio. 2018. Virtual spring damper method for nonholonomic robotic swarm self-organization and leader following. *Continuum Mechanics and Thermodynamics* 30, 5 (01 Sep 2018), 1091–1102.