A Scalable, Decentralised Large-scale Network of Mobile Robots for Multi-Target Tracking

Pham Duy Hung[1], Tran Quang Vinh[1], and Trung Dung Ngo[2]

[1] University of Engineering and Technology, Hanoi, Vietnam
[2] University of Brunei Darussalam, Brunei Darussalam
The More Than One Robotics Laboratory
www.morelab.org

Abstract. A decentralised large-scale network of mobile robots for multi-target tracking is addressed in this paper. The decentralised control is originally built up by behavioural control but upgraded with connectivity maintenance and hierarchical connectivity removal. The multi-target tracking algorithm guarantees that the mobile robots can reach the targets at the very high success rate while at least an interconnectivity network connecting all the mobile robots exists. The Monte-Carlo simulation results illustrate scalability properties of the large-scale network of mobile robots in terms of number of robots and types of scenarios.

Keywords: Decentralised Control, Multi-Target Tracking, Scalability, Connectivity Maintenance, Network Preservation, Hierarchical Connectivity Removal, Multi-robot Systems

1 Introduction

A decentralised large-scale network of mobile robots can be used for various applications in diverse environments, e.g., surveillance and reconnaissance, patrolling and monitoring, coverage, multiple target tracking in very wide and hazardous areas. To deploy such a system into an environment, the mobile robots have to manage their connectivities with immediate neighbours for not getting lost from the network. The mobile robots through interconnectivity of the network can make consensus decisions for their cooperative and coordinative operations of which multi-target tracking is one of case studies. Multi-target tracking is a special case of coverage but the mobile robots must track and reach specific targets in diversity of environments. To ensure that all the tracked targets are explored, identified, and reported back to the users, the mobile robots have to establish a communication network using inter-networking connectivities for data exchange. Hence, a decentralised control for connectivity maintenance and network preservation is the key factor for large-scale networks of mobile robots for multi-target tracking applications.

Decentralised control of multi-robot systems for connectivity maintenance has been grown up with two mainstreams, to our best of knowledge: artificial potential field and graph theory. The artificial potential field, which was originally coined out by [3] for control of mobile robot and manipulator, has been extended to widely apply for development of the decentralised control of multi-robot systems for connectivity maintenance. Synthesis of the attractive and repulsive forces generated by the artificial potential fields drives the mobile robots towards the goals without colliding with obstacles. Artificial potential forces used for multi-robot systems are typically divided into the three categories: linear function [4],[5], quadratic form [6],[7],[8], and exponential expression [9],[10], just to name a few. Graph theory is another approach representing connectivities of agents interacting and communicating in networked systems. Cooperative and coordinative operations of networked systems relied on connectives of agents can be mathematically managed by the graph theory. For examples, stable flocking of mobile agents in both fixed [11], and dynamic topology [12] are proven by algebraic graph theory. Connectivities of agents used to design co-ordination and formation control in multi-agent systems[13],[14] [15] are governed by the graph Laplacian. In [16], connectivities of networked agents is modelled by the weighted graph.

The authors in [17] introduced a distributed algorithm of a homogenous robot swarm with limited sensing for tracing moving objects. An extension of the basic behavioral set with following and circulating behaviors is realised to track and move around a desired object. In [18], Boyoon Jung, et al. released a behaviour-based control for tracking targets using multiple mobile robots. This controller consists of three functionality layers of motor actuation, monitoring, and target tracking: basic behaviors based motor actuation layer directs the robots towards targets; the monitoring layer observes the internal status of the controller during operation; and the target tracking layer detects and estimates the target positions. Similarly, in [19], authors described a control algorithm for distributed robotic macro sensors based on the virtual spring mesh to track targets in both discrete and diffuse nature. Parker in [20] presented a behavior-based method of developing distributed algorithms for cooperative robot observation of multiple moving targets. The distributed control was synthesised by force vectors generated by relative localisation between the robots, and the robots and the moving targets with specific weights. In [21], the authors presented dynamic target tracking and observing in a mobile sensor network in two cases: tracking a moving target with complex environment by adaptive flocking control algorithm of which robotic sensor nodes cooperatively learn the network's parameters to decide the network size in the decentralised fashion, multiple dynamic target tracking by a Seed Growing Graph Partition (SGGP) algorithm proposed to solve the problem of splitting/merging the sensor agents from the network. In [22], a team of robots is used to catch the evader or defend an area. The authors proposed a target tracking algorithm for a robot team using fuzzy cost function control in the framework of game theory. A scalable and fault-tolerant framework for distributed multi-robot patrol is presented in [23]. Domagoj et.al [24] proposed the cooperative multi-target tracking using hybrid modelling and geometric optimisation.

To our best of knowledge, the existing decentralised controls have primarily focused on connectivity maintenance/preservation of mobile robots without considering applications of the robotic systems. As a result, the decentralised network of the mobile robots is not able to expand to reach the all the farthest targets. On the other hand, most of previous works of target tracking or multiple target tracking by mobile robots has not seriously considered scalability of the network as they have only demonstrated a few case studies with a lot of predefined conditions of systems and environments.

To overcome shortcomings of existing connectivity maintenance methods for multiple-target tracking, we propose the new method to design the decentralised control for large-scale network of mobile robots for multiple target tracking. The decentralised control is developed by the origin of behavioural control but upgraded to become the hierarchically structured control of behavioural control, connectivity maintenance, and hierarchical connectivity removal. The mobile robots with new control are capable of keeping or removing connectivities with their nearest neighbours for expansion of the network coverage while retaining at least one interconnectivity through all the robots. We investigate scalability of the proposed decentralised control for large-scale network of mobile by the Monte-Carlo simulation method.

The rest of this paper is organised as follows: the graph-theoretic model and the hierarchical connectivity removal used to develop the decentralised control are illustrated in section 2. The algorithm of multi-target tracking is elaborated in section 3. The Monte-Carlo simulations and statistical results are shown in section 4. We conclude this paper with essential scalability properties of this decentralised large-scale network of mobile robots.

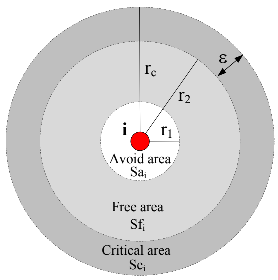
2 Decentralized Control

2.1 Graph-theoretic Model

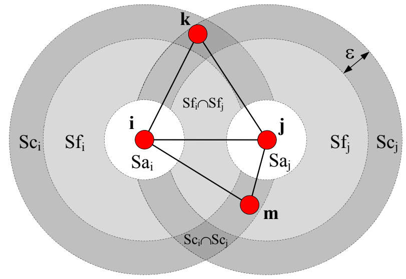
A large-scale network of N mobile robots and E connectivities made among them is described as an undirected graph $G(N, E)$. A connectivity between two mobile robots, i and j, is represented by an edge of the connectivity graph, $e_{ij} \in E$. A mobile robot i can perceive and communicate with its neighbouring robots N_i, if the relative distance between them is within the disk-based sensing and communication range r_c. That is, the connectivity e_{ij} between the robots i and j exists if $e_{ji} \leq r_c : j \in N_i$

Connectivity graph can be described by the mean of the adjacency matrix $\mathbf{A} \in \mathbf{R}^{N*N}$ Each element e_{ij} of the adjacency matrix $\mathbf{A}$ is defined as the weight of the edge between the robot i and the robot j, which is a positive value if $j \in N_i$, or zero otherwise. In the undirected graph, $\mathbf{A}$ is a symmetric matrix with each element e_{ij} represented below:

$$e_{ij} = e_{ji} = \begin{cases} 1 & \|\mathbf{r}_{ij}\| \leq r_c \\ 0 & \|\mathbf{r}_{ij}\| > r_c \end{cases} \tag{1}$$



(a) Connectivity range



(b) Critical robots

Fig. 1. Connectivity range and critical robots: (a) the robot j freely move within the Sf_i, move away from Sa_i, and is considered as a candidate of critical robots for the robot i if the robot j is within Sc_i; (b) the robot i has critical robots j, k, m; the robot j has critical robots i and k; the robot i is not the critical robot of the robot j.

Connectivity property of the mobile robots is identified through the second smallest eigenvalue λ_2 of the Laplacian matrix $\mathbf{A}$. The mobile robots are well connected if $\lambda_2 \geq 0$. Connectivity strength is proportional to the value of λ_2.

The network of the mobile robots start moving from an initial location, then navigate towards the multiple targets. During the movement, the mobile robots estimate the graph adjacency to check the connectivity property that is propagated to its neighbours.

We release the following definitions used to develop the decentralised control with network preservation.

Definition 1. (Sub Adjacency Matrix) *The robot i has a set of neighbours N_i. We define Sub Adjacency Matrix, **subA**, of the robot i as the adjacency matrix of N_i.*

The sensing and communication range of each mobile robot is divided into three areas: *obstacle avoidance* area **Sa** with radii range r_1; *free area* **Sf** inside annulus circle between two radii r_2 and r_1, and *critical area* **Sc** inside annulus circle between two radii r_c and r_2 as in Figure 2.1.

Definition 2. (Candidates as Critical Neighbours) *The j^{th} robot becomes a candidate for critical robot of the robot i if it is within the robot i's critical area.*

$$\mathbf{C}_i = j \in Sc_i \tag{2}$$

where ε is a constant, called as critical error.

Definition 3. (Critical Neighbours and Critical Connectivities) *The robot j is a candidate of critical robots of the robot i, $j \in C_i$. The robot i becomes a critical robot of the robot j, and the connectivity between the robot i and the robot j is considered as the critical connectivity, and vice versa if there does not exist any other robots inside the intersection area between their free areas $\mathbf{Sf}_j \cap \mathbf{Sf}_i$.*

$$\mathbf{Cn}_i = \{j \in C_i, : \mathbf{Sf}_j \cap \mathbf{Sf}_i = \emptyset\} \tag{3}$$

2.2 Hierarchical Connectivity Removal

In general, we have to take all the neighbours of the robot i into consideration for designing its decentralised control. However, the robot i's critical neighbours act like "anchors" to present the target reaching of the robot i. To allow the robot i to move towards the assigned target while preserving the network connectivities, we have to deal with three connectivity topologies established by the robot i and its critical neighbours Cn_i:

Triangle topology: There exists two critical neighbours j and k connected together in Cn_i, i.e., checked by $subA_i = 1$. Both the critical connectivities, e_{ij} and e_{ik}, cause the robot i impossible to reach the targets out of the coverage by the neighbours j and k. As illustrated in Figure 2(a), the triangle topology of three robots n_1, n_2, n_3 prevents the robot n_1 to reach the target T_1.

K-connected topology: A robot i has two critical neighbours j and k that are not connected in Cn_i, i.e., $subA_i = 0$. If the robots j, k have the neighbouring robots in $N \cap N_i$ connected directly or indirectly through groups of intermediate robots, denoted R_j, R_k, and the intersection of the two groups is not a empty set, $R_j \cap R_k \neq \emptyset$, the robots i, j, k establish a *k-connected topology* (a type of k-connected graph, where $k = 2$, in the graph theory). Specifically, k-connected topology can be in the form of four edges, so-called as quadrangle topology; five edges, so-called as pentagon topology; six edges, so-called as hexagon topology, and so forth. Both the critical connectivities e_{ij} and e_{ik} cause the robot i impossible to reach targets out of the coverage area of the robot j and the robot k. As illustrated in Figure 2(b), the k-connected topology with four robots n_5, n_6, n_7 and n_8 become anchors preventing the node n_8 to reach to the target T_2.

Algorithm 1 Hierarchical Connectivity Removal

Given Cn_i
if $subA_i = 1$ **then**
    if $e_{ij} > e_{ik}$ **then**
        e_{ij} *removed*
    else
        e_{jk} *removed*
    end if
end if
if $subA_i = 0$ *and* $R_j \cap R_k \neq \emptyset$ **then**
    if $e_{ij} > e_{ik}$ **then**
        e_{ij} *removed*
    else
        e_{ik} *removed*
    end if
end if

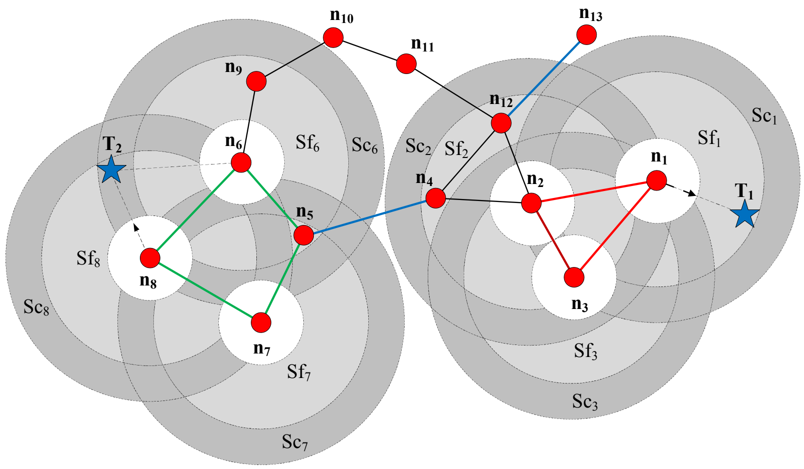


Fig. 2. Connectivity topologies: a) A *triangle topology* of the robot n_1 - red triangle - consists of two critical robots n_2 and n_3 making two critical connectivities e_{12} and e_{13} within Sc_1, and the connectivity $e_{23} \leq r_c$. b) A *k-connect topology* (quadrangle topology) of the robot n_8 - green polygon - consists of three critical robots $\{n_5$,n_6, $n_7\}$ in which the intersection of two groups of the robots n_6 and n_7 is non-empty, i.e. $R_6 = \{n_5\}$, $R_7 = \{n_5\}$, $R_6 \cap R_7 = \{n_5\} \neq \emptyset$. c) An *one-connected topology* of the robot n_5 - blue line - between the robot n_5 and the robot n_4 is the critical connectivity e_{45}, i.e. $R_5 = \{n_6, n_7, n_8\}$, $R_4 = \{n_1, n_2, n_3\}$, $R_5 \cap R_4 = \emptyset$. d) A *k-connected topology* vs. a *one-connected topology*: any robot in the group $\{n_4, n_5, n_6, n_9, n_10, n_11, N_12\}$ consists of either *one-connected topology* or *k-connected topology* with its neighbours, depending the neighbourhood level ℓ($\ell \leq 3$ for *k-connected topology* and $\ell \geq 4$ for *one-connected topology*)

.

One-connected topology: A robot i has only one critical neighbour j, and vice versa the robot j has only one critical neighbour i. If the robots i and j have the neighbouring robots in $N \cap N_i$ connected directly or indirectly through intermediate robots, denoted R_i, R_j, and the intersection of the two groups is a empty set, $R_i \cap R_j = \emptyset$, the robots i and j make a *one-connected topology*. The connectivity between the robot i and the robot j must be preserved for the network intercommunication. If $R_i \cap R_j \neq \emptyset$, the critical connectivity e_{ij} causes the robot i impossible to reach its target as illustrated in Figure 2(c).

Hierarchical Connectivity Removal: We propose the hierarchical procedure for removing connectivities allowing the mobile robots to move towards the targets while still preserving the network if they fall into one of two cases, *triangle topologies* and *k-connected topologies*. No connectivity is removed if only *one-connected topologies* exist:

Note that this decision is rather arbitrarily chosen since the neighbours with the longer connectivity tends to escape from the network preservation. However, the shorter connectivity can be chosen as well.

2.3 Decentralised Control

Behavioural Control (BC) Inspired from behavioural control in [1][2], velocity vector of the robot i, v_i^t, is synthesised by cohesion v_i^c, separation v_i^s, and alignment v_i^a velocity vectors.

$$\mathbf{v}_i = \alpha \mathbf{v}_i^c + \beta \mathbf{v}_i^s + \gamma \mathbf{v}_i^a \tag{4}$$

where α, β, γ are factors to adjust cohesion, separation, and alignment for the overall behaviour.

Cohesion $\mathbf{v}_i^c$ is the vector driving the robot i towards its neighbouring robots satisfying $j \in Sf_i \cup Sc_i$. Separation $\mathbf{v}_i^s$ is the vectors that drives the the robot i away from its neighbours satisfying $j \in Sa_i$. Alignment $\mathbf{v}_i^a$ is the vector guiding the robot i towards the centre of target cloud when all the robots are moving toward the the target cloud. Alignment $\mathbf{v}_i^a$ becomes the vector for the robot i towards its assigned target when the first robot of the network reached the first target.

The factors α, β, γ are normalised (all the robots are assigned with the same α and γ) in order to guarantee the smooth movement of the whole network of mobile robots towards the target cloud.

Connectivity Maintenance (CM) The decentralised control for connectivity preservation aims at keeping all the robots connected in the network for data intercommunication - no robot is disconnected from the network at any time. This decentralised control is only activated when a robot has critical robots and critical connectivities. The control is synthesised by the current velocity of the robot and its critical robots' relative positioning as in (5):

$$\mathbf{v}_i^{t+1} = \sum_{j \in C_i} (\mathbf{v}_i^t + \mathbf{v}_{op\_ij}^t) \tag{5}$$

where v_i^t is the current velocity of the robot i, and $v_{op\_ij}^t$ is the projection of v_i^t on the edge of the robots i and the robot j, e_{ij}. At an instance, the modified velocity of the robot i for preserving connectivity with the critical neighbour j must satisfy the condition: $\|\mathbf{v}_i^{t+1}\| < \min\limits_{j \in C_i} \frac{r_c - \|\mathbf{r}_{ij}^t\|}{2\Delta t}$, where r_{ij} is the relative distance between the robot i and the neighbour j. If $\|\mathbf{v}_i^{t+1}\| \geq \min\limits_{j \in C_i} \frac{r_c - \|\mathbf{r}_{ij}^t\|}{2\Delta t}$, the velocity v_i^{t+1} is normalised by a factor $\xi < \frac{\min\limits_{j \in C_i} (\frac{r_c - \left\|\mathbf{r}_{\mathrm{ij}}^t\right\|}{2\Delta t})}{\|\sum\limits_{j \in C_i} (\mathbf{v}_i^t + \mathbf{v}_{op\_ij}^t)\|}$ to ensure that the robot i cannot disconnect with the robot j, $r_{ij} \leq r_c$. Since then, the velocity of the robot i is recalculated by the numerical factor ξ:

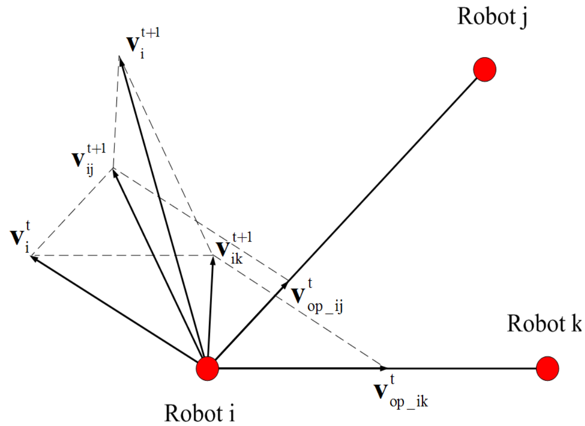


Fig. 3. Decentralised control for connectivity preservation

$$\mathbf{v}_i^{t+1} = \xi \sum_{j \in C_i} (\mathbf{v}_i^t + \mathbf{v}_{op\_ij}^t) \tag{6}$$

Hierarchical Connectivity Removal (HCR) The decentralised control with connective maintenance guarantees all the robots well connected through network. However, on one hand, connectivity preservation prevents the mobile robots to move towards the assigned targets. On the other hand, there might exist more than one connectivity or inter-connectivity between two mobile robots so that some of connectivity or interconnectivity are removable to allow the mobile robots to reach the targets. The hierarchical connectivity removal in Algorithm 1 is only triggered if there exist at least two critical neighbours. That is, the mobile robot are still working with the connectivity maintenance but intelligent removing unnecessary connectivities to accelerate the process of multiple target tracking.

3 Multi-Target Tracking

All the targets are unknown to the robots due to their limited sensing range at the beginning. We assume that the network of mobile robots knows the direction of the target cloud so the whole network is sent towards that direction for exploration and target tracking. Initially, the network of mobile robots moves towards the target cloud with the behavioural control. If the first robot detects a target, the multi-target tracking algorithm is triggered for multiple target tracking.

Algorithm 2 Multi-Target Tracking

```
for all free robots do
  search for unoccupied targets
  if observe unoccupied targets then
    move to nearest target
    if can not reach then
      become local minimal node
    else
      become anchor
      if anchor observe unoccupied targets then
        call free robots through network communication
        become landmark attracting free robots
      end if
    end if
  end if
  search for local minimal nodes
  if local minimal nodes then
    move to local minimal node
    if can not reach then
      become local minimal node
    end if
  end if
  search for farthest lanmarks
  if farthest anchor then
    move to farthest landmark node
    if can not reach then
      become local minimal node
    end if
  end if
  {insert Algorithm 3 here if more robots needed}
end for
```

One robot can only track and hold one target. If a robot observes number of targets, the target in the shortest distance is selected. The robot moves towards and occupies the selected target if the distance between them less than the detecting range, set $0.05r_c$. Once the robot successfully occupied the target, the target is marked as *occupied target* that is no longer occupied by the other robots. This robot becomes an *anchor* - a stationary node of the network. If this anchor sees *unoccupied targets* within its sensing range, it informs the other robots about unoccupied targets through the network. This anchor plays the role as the landmark in the network in order to direct the other free robots to move towards these free targets through the network intercommunication. If the robot move towards the selected target but cannot occupy it according to constraints of connectivity maintenance, it becomes a *local minimal node* of the network that expects to receive assistance from their peers to get off this position.

If a robot has been not assigned a target, but it has higher priority to move towards the nearest *local minimal node* in order to assist this trapped robot to escape this position such that the trapped robot can move towards its assigned target. If there is no *local minimal node* in the network, the robot moves towards the most farthest *anchor* from the centre of the network, where there might exist high possibilities of *unexplored targets*.

Note that when the robot are moving towards the assigned target, it requests a number of its nearest neighbours who have not been assigned tasks to follow. Thanks to this technique, we can achieve a twofold acceleration of the multi-target tracking: 1) if the robot becomes an anchor, it has high possibility of observing new *unoccupied targets* that can be occupied immediately by the following robots; 2) if the robot becomes a *local minimal node* of the network, it can receive assistance of its peers immediately to get off from this situation.

We also assume that all the robots can communicate with their peers in order to update the network status in terms of *occupied targets*, *unoccupied targets*, *anchors*, *local minimal nodes*, and *assigned robots*, or *free robots* if they are well-connected in the network. All the robots of the network work with the decentralised control described as in Algorithm 2.

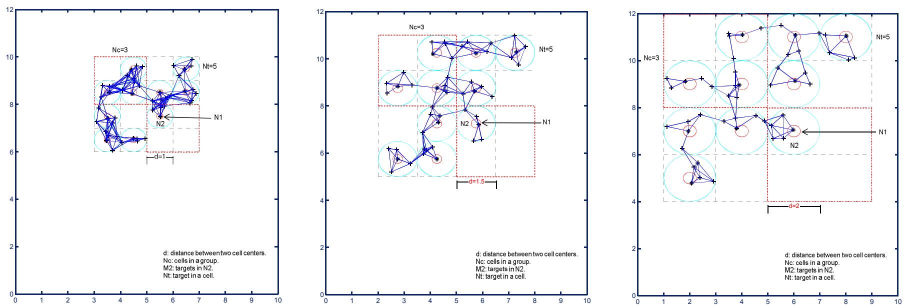
If there exists a number of local minimal nodes in the network, the local minimal nodes can request to add more robots into the network by Algorithm 3. We assume that the a mother robot of the network can send free robots into the network. Once the added robots reached the local minimal nodes, the new synthesis of force vectors for the decentralised control of the local minimal nodes is changed, allowing the local minimal nodes to escape from the trapped locations, then towards the assigned targets.

Algorithm 3 Adding Robots

```
if unoccupied targets observed by local minimal nodes or anchors through network communication then
    repeat
        mother sends a free robot towards anchors or local minimal nodes
    until all targets occupied
end if
```

4 Simulations and Discussions

In [25], we have proved that the developed decentralised control is capable of not only maintaining connectivity of the mobile robots but also intelligently removing unnecessary connectivities in order to expand coverage of the robot network and accelerate multi-target tracking process. In this paper we investigate scalability of the proposed decentralised control of multiple mobile robots for multiple target tracking while preserving the inter-networking communication through all the mobile robots. We evaluate the scalability through numerous different types of scenarios - probability distribution of scenario generation - at different difficulty levels - probability distribution of targets.



(a) d=1.0; N_t=5, N_c=3 (b) d=1.5; N_t=5, N_c=3 (c) d=2.0; N_t=5, N_c=3

Fig. 4. Examples of experimental scenario generation

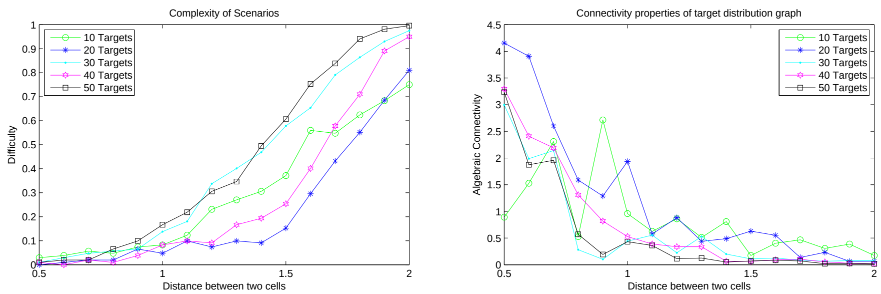
4.1 Experimental Scenario Generation

Rules for Scenario Generation: M targets are distributed with the Gaussian random distribution within the mxn area. A generated experiment scenario is selected if it satisfies the condition: at least a inter-communication link connecting all the targets with relative distance between a target and its immediate neighbours on the inter-communication link less than r_c exists. This is simply checked by the algebraic connectivity of the target graph. The area m x n is divided into cells with the size dxd. Two cyclic areas, $N1$ and $N2$, with the radius, $r_1 = 0.125d$ and $r_2 = 0.5d - r_1$, are created in every cell respectively. In both $N1$ and $N2$ areas, one target is placed in the $N1$ and M_2 targets are placed in $N2$ with the Gaussian random distribution, that is, there are $N_t = M_2 + 1$ targets in one selected cell. To diversify the scenario, we gather a four cells in a group, and number of cells are selected to fill up with targets, $N_c \leq 4$. Distribution of scenario generation is dependant to the parameters d and N_c while distribution of targets in a scenario is dependant to the parameters d,N_t, and N_c. We can create a *dense* scenario if d decreases and M_2 and N_c increase, or a *spare* scenario if d increases and M_2 and N_c decrease as examples illustrated in Figure 4.

Complexity of Scenarios: Complexity of scenarios is an index reflecting the possibility of the network of mobile robots to reach targets in specific scenarios. This rate is related to the difficulty of generating scenarios that are qualified for experiments, measured by the fraction of unsuccessfully generated scenarios and total generated scenarios.

$$f_{diff}(d, N_t, N_c) = \frac{S_{tol} - S_{suc}}{S_{tot}} \tag{7}$$

where S_{tot} represents the total generated scenarios and S_{suc} represents successfully generated scenarios.



(a) Difficulty rate (b) Algebraic connectivity

Fig. 5. Complexity of scenarios ($N_c = 4$, $N_t = 5$, $0.5r_c - 2.0r_c$)

Definition 4. *Difficulty of scenarios is the rate of generating scenarios unsuccessfully over the total generated scenarios. If N scenarios are created and none of them is qualified for experiments, the difficulty rate = 1 (100%) and this type of scenario is extremely difficult. The difficulty increases when the rate increases.*

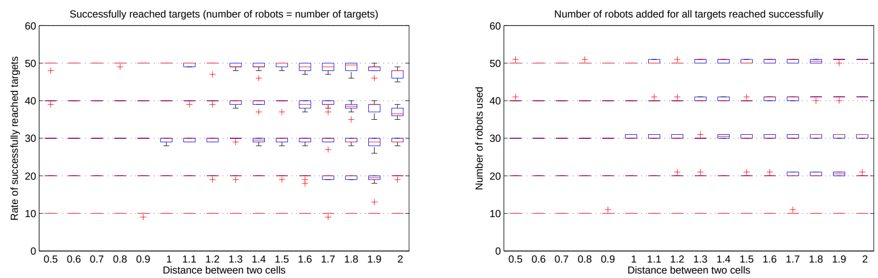
Using the Monte-Carlo method, we measured the difficulty of scenarios by generating 100 scenarios successfully for a set of parameters d,N_t, and N_c where N_c : 4, N_t : 5 d: $0.5r_c - 2.0r_c$. We chose five types of scenarios from 10 to 50 targets for statistical data as illustrated in Figure 5a. The difficulty rate is measured by 100 successfully generated scenarios with the total targets increased from 10 to 50. We observed the difficulty rate f_{diff}: close to 0 - most of generated scenarios usable for experiments - when d is approximately $0.5r_c$ and almost 1 - most of generated scenarios unusable for experiments - when d is approximately $2.0r_c$. The difficulty rate is also observed by the property of algebraic connectivity of the target graph as seen in Figure 5b. Indeed, connectivity property of the target graph is proportional to the complexity of the scenarios.

4.2 Results and Discussions

We examine the scalability of the decentralised control of large-scale network of mobile robots in three criteria:

– **Systematic scalability** (scalable with number of robots): whether the decentralised control can deal with different large number of mobile robots in the network.

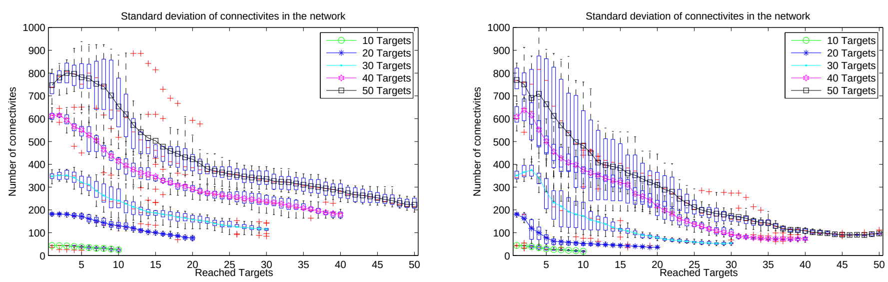
– **Spatial scalability** (scalable with different scenarios): whether the decentralised control can govern the large-scale network of mobile robots in different types of scenarios.



(a) Successfully reached targets vs. used robots

(b) Assistive robots added into the network

Fig. 6. Statistical results of robots used in the network and successfully reached targets



(a) $N_c = 4$, $N_t = 5$, $d = 1.0r_c$

(b) $N_c = 4$, $N_t = 5$, $d = 2.0r_c$

Fig. 7. Link reduction over time

- **Temporal scalability** (scalable in performance time): whether the decentralised control can handle the large-scale network of mobile robots in bounded time consuming w.r.t number of the robots and difficulty levels of scenarios.

We have collected data from 800 executed simulations (10 simulations for each of 16 parameter sets of $d : 0.5r_c - 2.0r_c$, $N_t = 5$, $N_c = 4$ in 5 types of scenarios) of 800 different scenarios filtered from hundreds of thousands generated scenarios by the rules of experimental scenario generation. The statistical results show that :

- The decentralised network of mobile robots is scalable with the number of robots as seen in Figure 6.a. The number of mobile robots in the network increases from 10 to 50 robots but the decentralised control can guarantee the high rate of successfully reached targets (minimum 90% in the worst cases of the most difficult scenarios, $f_{diff} \approx 1$). Moreover, the decentralised control

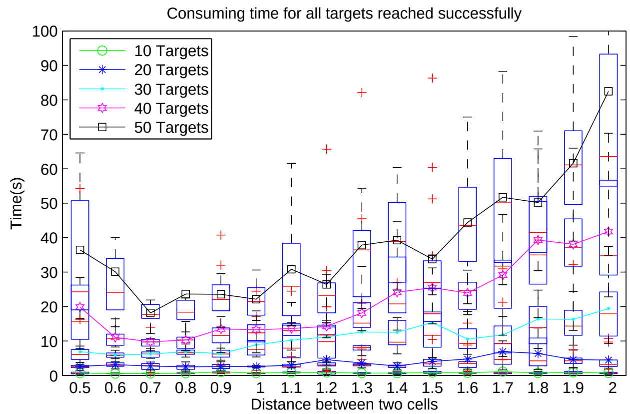


Fig. 8. Time consuming for all targets reached successfully

is also scalable with adding more assistive robots into the network when a number of local minimal nodes are found. The fact is that using the same decentralised control a few of assistive robots (maximum 2 assistive robots in the worst cases) are needed to get all local minimal nodes off the trapped positions in order to let them to reach their assigned targets successfully as seen in Figure 6.b.

– The decentralised network of mobile robots is scalable with different scenarios at different difficulty levels. The control ensures the absolute rate of successfully reached targets in the *dense* scenarios ($d \leq r_c$) while keeps the rate more than 90% in the worst cases of the *spare* scenarios ($d = 2r_c$) as illustrated in Figure 6. Thanks to the hierarchical connectivity removal intelligently reducing the network connectivities as seen in Figure 7, the network is capable of spatially expanding to reach the farthest targets that are impossibly observed and reached by the mobile robots with the decentralised control for connectivity maintenance only.

– The decentralised network of mobile robots is scalable with the bounded time consuming w.r.t number of the robots and difficulty levels of scenarios as shown in Figure 8. The mobile robots are capable of reaching all the targets in bounded interval, instead amount of time dramatically increasing to ∞, at any scale of the robots and any difficulty levels of scenarios, allowing possibilities of optimising usability of the large-scale network of the mobile robots according to diversity of system requirements and environmental conditions. However, the network is not temporally so consistent with the same types of scenarios as time consuming highly varies as illustrated Figure 8. A target cloud exploration strategy added to the multi-target tracking algorithm might be needed to optimise the robots' trajectories and interactions in order for consistence of time consuming.

5 Conclusion

We have presented the decentralised control for scalable large-scale network of mobile robot for multi-target tracking. Origin of this control is a behavioural control but upgraded with connectivity maintenance and hierarchical connectivity removal. Thanks to the upgrades, all the mobile robots not only preserve interconnectivity through the network but also remove unnecessary connectivities to allow them to reach all the targets. The Monte-Carlo simulation results demonstrate that the large-scale system is systematically, spatially, and temporally scalable with assignments of multiple target tracking.

Acknowledgement:
This research was supported in part by the University of Brunei Darrusalam (*UBD/PNC2/2/RG/1(259)*) and the Asia Research Centre and the Korea Foundation for Advanced Studies (*Developing Swarm Dispersion Algorithms of Multi-Robot Systems for Multi-Target Tracking* research project).

References

1. Reynolds,C. W., Flocks, Herds, and Schools: A Distributed Behavioral Model. Computer Graphics, 21(4), pp.25-34, 1987.
2. Mataric, M.J., Designing and Understanding adaptive group behaviors. Adaptive Behavior, Vol. 4, pp.51-80, 1995.
3. Khatib, O., Real-time Obstacle Avoidance for Manipulators and Mobile Robots. *Int. J. Rob. Res.*,1986, 5, 90-99.
4. Elkaim, G.; Siegel, M. A., Lightweight Control Methodology for Formation Control of Vehicle Swarms. In *Proceedings of the 16th IFAC World Congress*, Prague, Czech Republic, 4–8 July 2005.
5. Reif, J.; Wang, H., Social potential fields: A Distributed Behavioral Control for Autonomous Robots. *Rob. Autonomous Syst.*, 1999, 27, 171-194.
6. Spears, W., Spears, D., Hamann, J. Heil, R. Distributed, Physics-based Control of Swarms of Vehicles. *Autonomous Robot.*, 2004, 17, 137-162.
7. Ge, S.S., Cui, Y.J., New Potential Functions for Mobile Robot Path Planning. *IEEE Trans. Rob. Autom.*, 2000, 16, 615-620.
8. Kim, H.D.; Shin, S., Wang, O.H., Decentralized Control of Autonomous Swarm Systems, Using Artificial Potential Functions: Analytical Design Guidelines. *Int. J. Intell. Rob. Syst.*, 2006, 45, 369-394.
9. Horward, A., Mataric, M., Sukatme, G., Mobile Sensor Network Deployment using Potential Fields: A Distributed, Scalable Solution to the Area Coverage Problem. In *Proceedings of the Sixth International Symposium on Distributed Autonomous Robotics Systems*, Fukuoka, Japan, 25–27 June 2002; pp. 229-208.
10. Mikkelsen, B.S., Jespersen, R., Ngo, T.D., Probabilistic Communication based Potential Force for Robot Formations: A Practical Approach. In *Springer Tracts in Advanced Robotics*, Vol 83, 2013, pp 243-253.
11. Tanner, G.H., Jadbabai, A., Pappas, J.G., Stable Flocking of Mobile Agents, Part I: Fixed Topology. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, Maui, HI, USA, 12 December 2003; pp. 2010-2015.

12. Tanner, G.H., Jadbabai, A., Pappas, J.G., Stable Flocking of Mobile Agents, Part II: Dynamic Topology. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, Maui, HI, USA, 12 December 2003; pp. 2016-2021.
13. Desai, P.J., A Graph Theoretic Approach for Modelling Mobile Robot Team Formations. *J. Rob. Syst.*, 2002, 19, 511-525.
14. Dong, W., Guo, Y., Formation Control of Nonholonomic Mobile Robots using Graph Theoretical Methods. *Lect. Notes Econ. Math. Syst.*, 2007, 588, pp. 369-386.
15. Ji, M., Egerstedt, M.. Distributed Coordination Control of Multi-agent Systems while Preserving Connectedness. *IEEE Trans. Rob.*, 2007, 23, pp.693-703.
16. Olfati-Saber, R. Murray, M.R., Consensus Problems in Networks of Agents with Switching Topology and Time-delays. *IEEE Trans. Autom. Control*, 49, pp.1520-1533.
17. L. Blazovics, K. Crorba, B. Forstner, and C. Hassan, *Target tracking and surrounding with swarm robots*, Conference and Workshops on Engineering of Computer-Based Systems, pp.135-141, 2012.
18. B. Jung, and G. S. Sukhatme, *Tracking Targets using Multiple Robots: The Effect of Environment Occlusion*, Autonomous Robots Journal, Vol. 13, No. 3, pp. 191-205, 2002.
19. B. Shucker, and J. K. Bennett, *Target Tracking with Distributed Robotic Macrosensors*, Military Communications Conference (MILCOM), Vol. 4, pp.2617-2623, 2005.
20. L. Parker, *Distributed Algorithms for Multiple Observation of Multiple Moving Targets*, Autonomous Robots, Vol.12(3), pp231-255, 2002.
21. La H.M., Sheng W., Dynamic target tracking and observing in a mobile sensor network, in *Robotics and Autonomous Systems* 60(2012) 996-2009.
22. Istvan H., Krzysztof S., Robot team coordination for target tracking usig fuzzy logic controller in game theoretic framework, in *Robotics and Autonomous System* 57(2009) 75-86.
23. David P., Rui P. Rocha, Distributed multi-robot patrol: A Scalable and fault-tolerent framework, in *Robotics and Autonomous Systems* 61(2013) 1572-1587.
24. Domagoj T., Rafael F., and Silvia F., Cooperative Multi-Target Tracking via Hybrid Modeling and Geometric Optimization, 17^{th} *Mediterranean Conference on Control and Automation*, Mekedonia Palace, Thessaloniki, Greece, June 24-26, 2009.
25. Pham .H.D., Pham .M.T, Tran .Q.V, Ngo. T.D, Accelerating Multi-Target Tracking by a Swarm of Mobile Robots with Network Preservation, in *Proceedings of International Conference of Soft Computing and Pattern Recognition*, 2013, December, Hanoi, Vietnam