Centralized and Decentralized Assignment and Planning of Trajectories for Multiple Robots

Sabaritha Sekar
School of Engineering and Applied Science
University of Pennsylvania
Email: sabsekar@seas.upenn.edu

Nitin J. Sanket
School of Engineering and Applied Science
University of Pennsylvania
Email: nitinsan@seas.upenn.edu

Sarath Kumar Barathi
School of Engineering and Applied Science
University of Pennsylvania
Email: sarathba@seas.upenn.edu

***Abstract*—Solutions for the goal assignment and trajectory planning problem in multi-robot systems is presented in this report. This is done by implementing Concurrent Assignment and Planning of Trajectories. This algorithm has two variants, namely, Centralized and Decentralized. The algorithm to be used in a particular scenario is chosen depending on our knowledge of the environment. Various experiments on the variation of parameters with respect to number of robots and communication range were conducted to analyze the robustness of the solution and a comparative study of the two solutions was performed.**

$X(t)$	Robot Position
G	Goals
ϕ^*	Distance squared assignment matrix
ϕ	Assignment matrix from Hungarian Algorithm
$\gamma^*(t)$	Trajectory equation
t_0	Start time
t_f	Final time
h	Sensing range
R	Robot radius

I. Introduction

Multi-robot path planning has gained momentum in the recent years [1] [2] due to its prospective application in a wide variety of fields. Simultaneous deployment of multiple robots requires goal assignments such that the trajectories so generated are collision-free (safety). Some strategies of multi-robot path planning include use of evolutionary algorithms [3], graph based methodologies and centralized approaches. We will be focussing on the last approach for this project.

II. Problem Definition and Solutions

The problem statement was to generate optimal and safe trajectories for multiple robots in a known obstacle-free two or three dimensional environment. Optimality is defined as the minimization of the total distance traveled by all the robots to reach their respective assigned goal locations. In order to solve this problem, we will decouple the goal assignment and the trajectory planning problems. This is done using Concurrent Assignment and Planning of Trajectories (CAPT). Some of the solutions we came up with for the aforementioned problems were:
(a) Assuming robots as interchangeable, i.e., it does not matter which robot goes where, we solve the goal assignment problem using the Hungarian algorithm. This assumes that the planner knows the state of the robots, goals and the environment. Hence this algorithm is called Centralized CAPT or C-CAPT.
(b) Solution (a) might not be feasible always, hence in this solution called Decentralized CAPT or D-CAPT, we assume that the planner does not have the complete information of the system. However, each robot has a area around it wherein it can communicate with the robots in its purview, this concept is used to solve the assignment problem locally. As we can guess, this will not yield the most optimal solution but it will ensure safety.

III. Centralized-CAPT

C-CAPT [4] seeks to find optimal trajectories and safe trajectories for multiple robots concurrently in a known environment. This is done by minimizing a cost functional, which in this case will be the sum of squared distances between every start to every goal. Given all the start locations, a goal is assigned to it such that the path from the robot to goal is optimal. The distance squared assignment matrix is given by:

$$\phi^* = \arg\min_{\phi} \sum_{i=1}^{N}\sum_{j=1}^{N} \phi_{ij} D_{ij}$$

where D_{ij} is the distance from start to goal, given by

$$D_{ij} = \|x_i(t_0) - g_j\|^2$$

From this, the goal assignment matrix can be obtained using the Hungarian Algorithm as ϕ of $\mathcal{O}(N^3)$ From the goal assignment, the trajectory can be obtained as:

$$\gamma^*(t) = (1 - \beta(t)X(t_0) + \beta(t)(\Phi G + (I_{Nn} - \Phi\Phi^T)X(t_0))$$

Where,

$$\beta(t) = \alpha_0 + \alpha_1 * t$$

Since we considered 1^{st} order robots,

$$\alpha_0 = \frac{-t_0}{t_f - t_0}, \alpha_1 = \frac{1}{t_f - t_0}$$

and,

$$\Phi = \phi \otimes I_n$$

where $\otimes$ is the kronecker product.

IV. Decentralized-CAPT

D-CAPT is used to plan an optimal, safe trajectory in an environment where the interaction between the robots are local. A sensing range h is defined for communication such that $h > 2\sqrt{2}R$, where R is the radius of the robot. When robots come within communicating range of each other, they learn new information about the neighbors. The robots communicate information about their current state and their assigned goals.From this information, $u_{ij}^T w_{ij}$ can be computed. The robots will collide if,

$$u_{ij}^T w_{ij} < 0$$

where $u_{ij} = x_j(t_c) - x_i(t_c)$ and $w_{ij} = x_j(t_f) - x_i(t_f)$ Then the goals of these to robots are reassigned as the other's (interchanged). The trajectory from any point to the goal is calculated as

$$x_i(t) = \left(1 - \frac{t - t_c}{t_f - t_c}\right) x_i(t_c) + \left(\frac{t - t_c}{t_f - t_c}\right) f_i$$

t_c is the time at which the trajectory changes due to swapping of goals.

V. Experiments and Results

All experiments were performed on a PC with Intel ®Core^{TM}i7 Processor with 12GB of RAM.

A. C-CAPT on 2-dimensional linear robots

Multiple experiments were performed on C-CAPT by varying N and observing its effect on time taken. A plot of time taken versus N is shown in Fig. 1. We can observe that it varies as $\mathcal{O}(N^3)$ (Found using curve fit in MATLAB[5]), which is as expected as the Hungarian algorithm takes most of the time. Videos demonstrating C-CAPT running on 10 and 100 robots are shown in Figs. 2 and 3 respectively. We can observe that C-CAPT finds optimal and collision free paths for the robots. All the videos are also on YouTube and can be found at https://www.youtube.com/watch?v=vxlR1C6s1EY&feature=youtu.be.

B. C-CAPT on a team on quadrotors

The experiments corresponding to the following cases were performed.

1) Case 1 ($N = M$: When the number of robots is equal to the number of goals, the Hungarian algorithm[6] is used to solve the assignment problem.

2) Case 2 ($N > M$): When the number of robots is greater than the number of goals, goals are assigned to M robots using the Hungarian algorithm. The remaining $N - M$ robots hover at their initial locations, i.e., they are unused.

3) Case 3 ($N < M$): When the number of goals is greater than the number of robots, N goals are assigned to the robots using the Hungarian algorithm. After these goals are reached, C-CAPT is performed again on the remaining goals with the previously reached goals as their start locations.

A Video demonstrating all the three cases is shown in Fig. 4. We can observe that C-CAPT performs well in all the three cases/scenarios. Also a plot of minimum distance between any two quadrotors is shown in Fig. 5 for all three cases.

C. D-CAPT on 2-dimensional linear robots

Multiple experiments were performed on D-CAPT by varying h and N and observing their effects on number of swaps, sub-optimality, number of messages sent and time taken. A plot of number of swaps versus N is shown in Fig. 6. We can observe that it varies as $\mathcal{O}(N)$ (Found using curve fit in MATLAB). A plot of number of messages versus N is shown in Fig. 7. A plot of sub-optimality versus N as compared to C-CAPT for the same start and goal locations is shown in Fig. 8. We can observe that it varies as $\mathcal{O}(ae^{bN^2})$ (Found using curve fit in MATLAB). Time of execution versus N is shown in Fig. 9. We can observe that it varies as $\mathcal{O}(N^3)$ (Found using curve fit in MATLAB). A plot of number of messages versus $\frac{h}{R}$ is shown in Fig. 10. A plot of sub-optimality versus $\frac{h}{R}$ as compared to C-CAPT for the same start and goal locations is shown in Fig. 11.

VI. Effect of Communication Delay (τ_d) on h in D-CAPT

If $\tau_d << \frac{\delta d}{V}$, where δd is the incremental distance traveled with a velocity V, assuming that V stays constant over δd, there is negligible effect on h, i.e., we do not have to increase h to ensure safety. In contrast, if $\tau_d \geq \frac{\delta d}{V}$, which is highly realistic if we have a noisy medium and acknowledgement-reacknowledgement protocol is being used for data exchange, the safety of the robots is compromised, increasing the probability of collision. In order to restore the safety to zero-delay case, we will have to grow h to

$$h' = h + V_{max}\tau_d$$

where, V_{max} is the maximum velocity of the robot.

VII. Proposal for Reduction of Sub-Optimality in D-CAPT Using C-CAPT Locally

As we've seen from the previous sections D-CAPT results in suboptimal paths when compared with C-CAPT. In D-CAPT, it is assumed that the memory χ of each robot is enough to store it's respective goal location. However, in real life each robot has sufficient memory to store more than one goal location. When more than two robots cluster together as shown in Fig. 12. Robot 1 and 2 are in each-others' sensing range and so are 2 and 3. If each robot can store all the significant (in

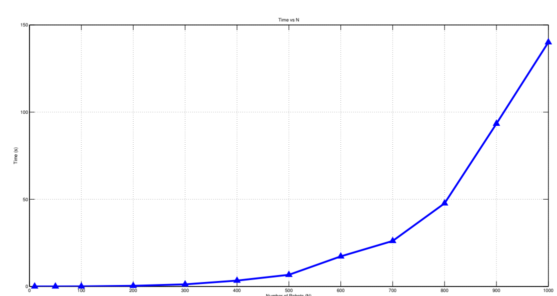

Fig. 1. Variation of time taken with respect to N for C-CAPT.

terms of cost) locations it has been to or explored and the respective goal locations and this information can be shared among all the robots via an ad-hoc like network. We can solve the path planning problem using the Hungarian algorithm and Dijkstra algorithm for path planning. However a simple swap is much faster than solving the Hungarian algorithm on the fly. We can model the time to perform Hungarian algorithm as a delay τ_d and to ensure safety we will have to increase h according to the previous section. This methodology will use up more resources in terms of memory and processing power on each robot, however on the plus side we will be reducing the amount of sub-optimality.

VIII. Benefits and Drawbacks of D-CAPT

A. *Benefits of D-CAPT*

- A centralized agent is not involved. Hence we do not need the knowledge of the entire map.
- Computational and memory complexity of operations reduces because Hungarian algorithm ($\mathcal{O}(N^3)$) is not used.
- Computations are done online. Hence it can adapt to changes in the environment.

B. *Drawbacks of D-CAPT*

- The path obtained is sub-optimal. Refer to Fig. 13, at 0:02s we can observe that the robots 1 and 5 perform self loops which are highly inefficient in terms of time and distance traveled. Also refer to Fig. 14, at 0:05s we can observe that the robots 3 and 4 over-travel the distance by almost half extra path length which is highly sub-

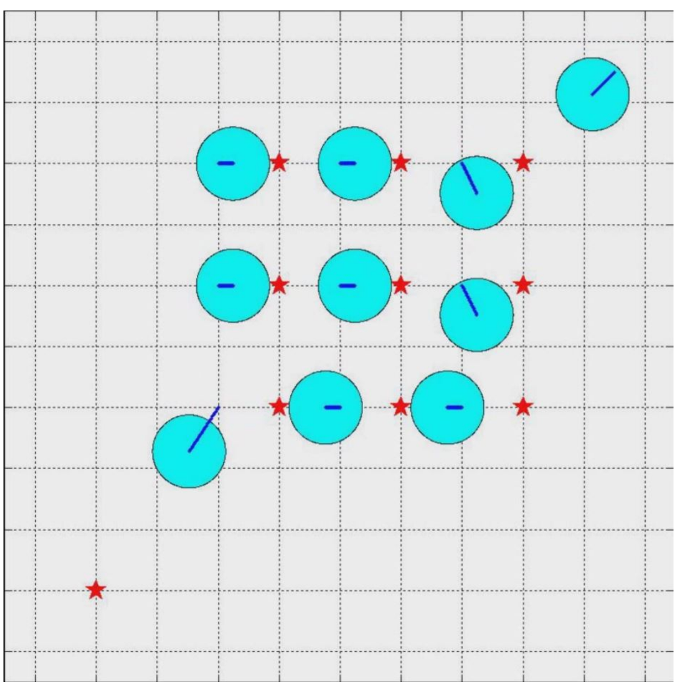

Fig. 2. Video showing the execution of C-CAPT on 10 robots.

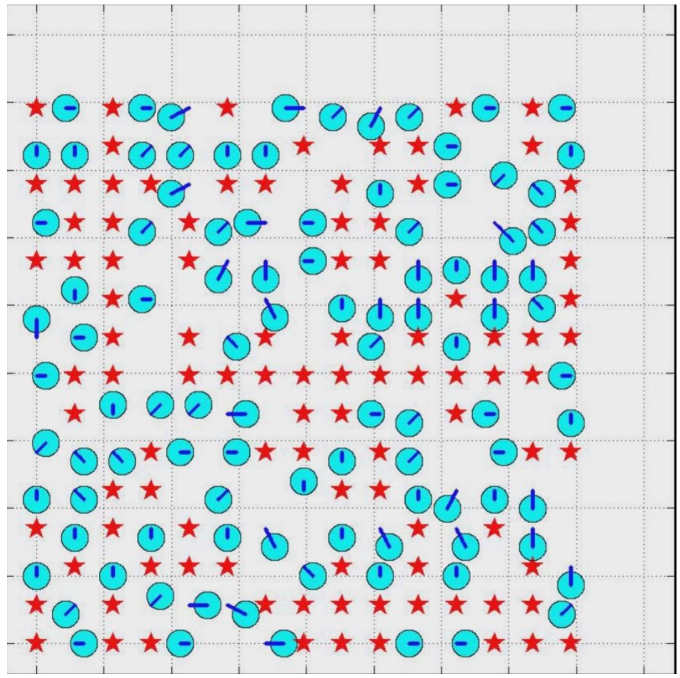

Fig. 3. Video showing the execution of C-CAPT on 100 robots.

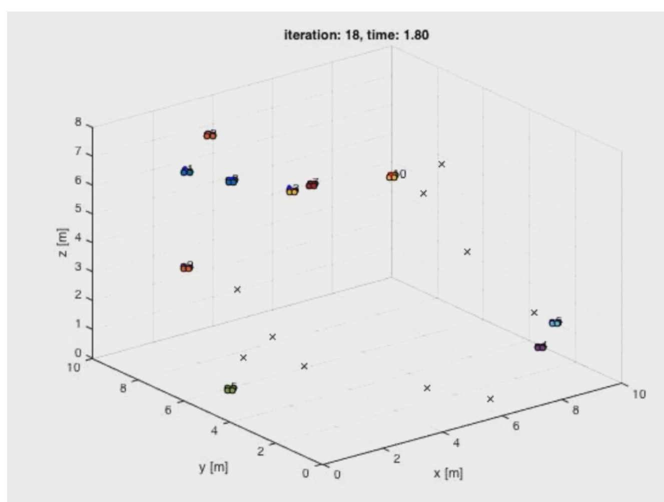

Fig. 4. Video showing the execution of C-CAPT on multiple quadrotors (simulated dynamics and PD controller).

optimal. These videos show how not knowing about the environment can lead to sub-optimality. Refer to Fig. 15, we can see how robot 1 has a cascading effect on all other robots, this is a special case demonstrating a high degree of sub-optimality.

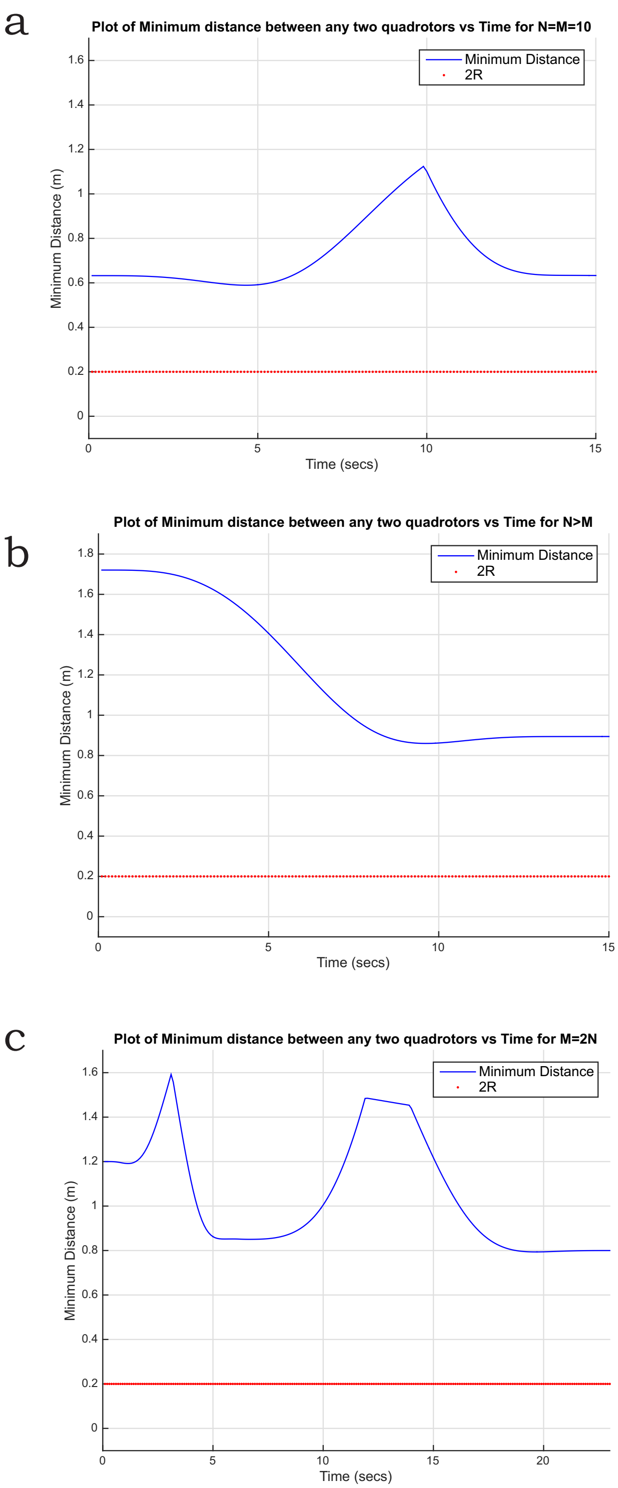

Fig. 5. Plot of minimum distance between any two quadrotors for (a) $N = M$, (b) $N > M$ and, (c) $M = 2N$.

- Communication delays is not explicitly modeled. This can cause collisions.
- Number of messages sent becomes very large as the number of robots grow, i.e., the same message will be

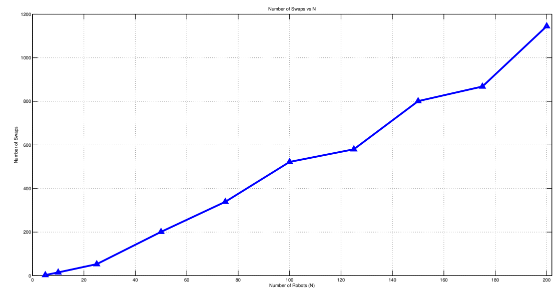

Fig. 6. Variation of Number of Swaps with respect to N for D-CAPT.

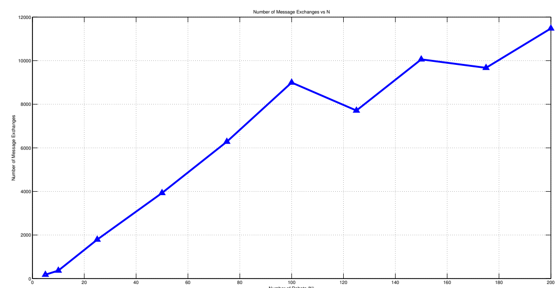

Fig. 7. Variation of Number of Messages sent with respect to N for D-CAPT.

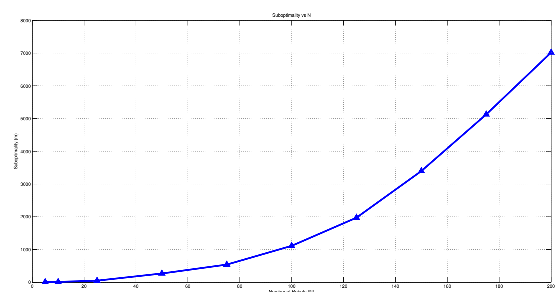

Fig. 8. Variation of Sub-optimality with respect to N for D-CAPT.

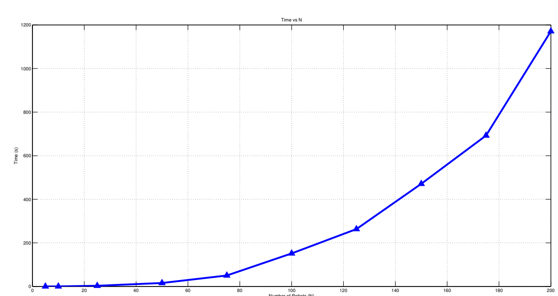

Fig. 9. Variation of Time to execution with respect to N for D-CAPT.

sent to multiple robots which is inefficient. This can be avoided by using our proposed methodology to run C-CAPT locally using an ad-hoc network. Refer to Fig. 16, we can observe the massive number of goal changes. In all the D-CAPT videos the robot extent is shown in orange color, the communication range is shown in green color, start locations are shown as blue squares, yellow lines denote the trajectories followed and the red stars show the goal locations. The robot number is shown inside the orange circle and the current assigned goal is shown as a number next to the red stars.

- Some special initial conditions (start locations) will lead to non collision free trajectories. So, we have to modify the trajectories to follow a specific sequence of trajectories if it detects an immediate collision.

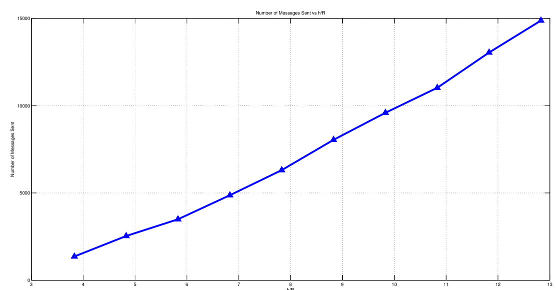

Fig. 10. Variation of Number of Messages sent with respect to $\frac{h}{R}$ for D-CAPT.

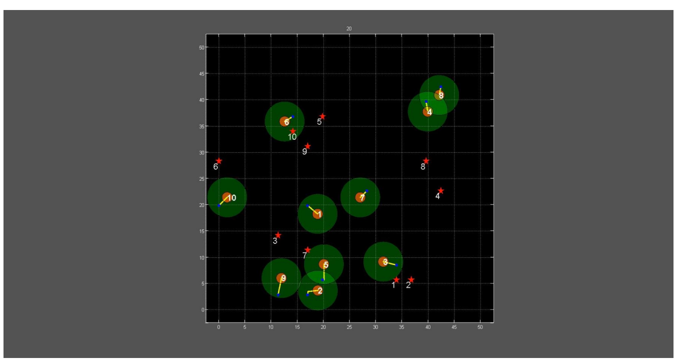

Fig. 13. Video showing sub-optimality of D-CAPT due to self loops (See 0:02s).

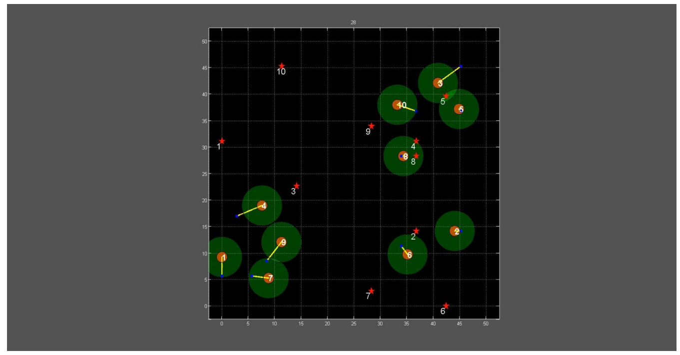

Fig. 14. Video showing sub-optimality of D-CAPT due to extra path traveled due to no knowledge of environment (See 0:05s).

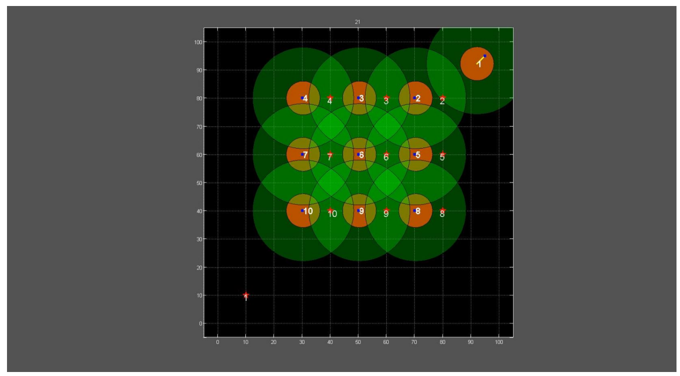

Fig. 15. Video showing sub-optimality of D-CAPT due to cascading effect.

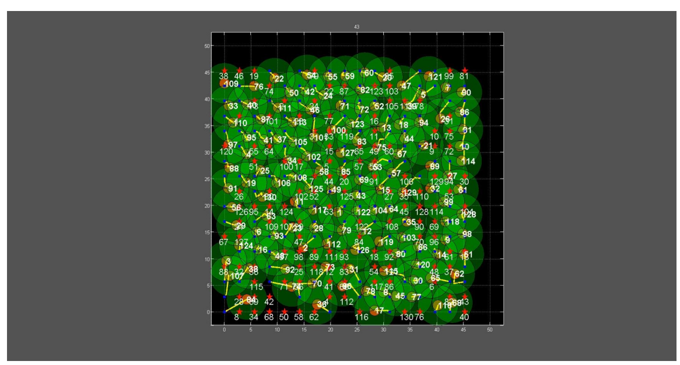

Fig. 16. Video showing large number of message exchanges, goal swaps in D-CAPT on a large number of robots which shows the inefficiency of information exchange.

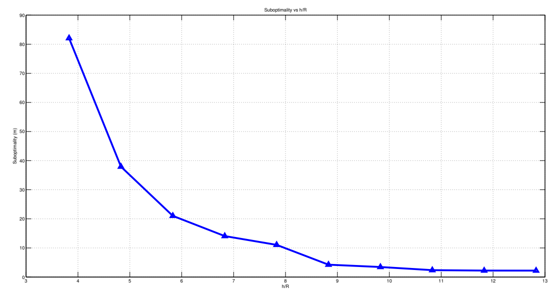

Fig. 11. Variation of Sub-optimality with respect to $\frac{h}{R}$ for D-CAPT.

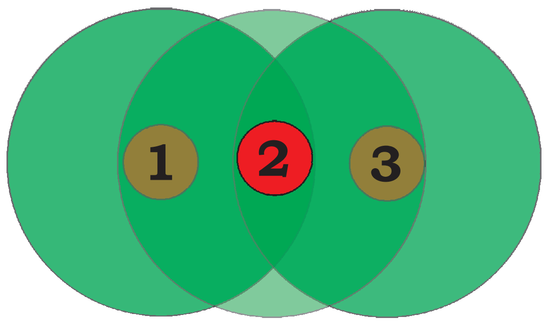

Fig. 12. Visualization of inefficient interaction between three robots.

IX. Conclusions

The centralized CAPT performed as expected by generating collision free time parameterized optimal trajectories satisfying the boundary conditions. The Hungarian algorithm decreased the time needed to solve the assignment problem. C-CAPT was tested on 2-dimensional circular robots and on 3-dimensional quadrotors. C-CAPT was also tested on $N = M$, $N > M$ and $M = 2N$, it performed as expected. C-CAPT had to be run twice for the last case. Due to the obvious disadvantages of C-CAPT being in real-world the whole environment is seldom known, a decentralized solution was proposed (D-CAPT). D-CAPT performed as expected to change the goal locations as to avoid collisions. However, as the number of robots increases, D-CAPT becomes inefficient in terms of number of messages sent and hence a solution to combat this was proposed. Also the effect of time delay on D-CAPT was studied. Effect of various parameters of C-CAPT and D-CAPT with respect to the number of robots was studied.

Acknowledgment

The authors would like to thank Dr. Vijay Kumar and all the Teaching Assistants of MEAM 620 course for all the help in accomplishing this project.

References

[1] Lynne E. Parker, *Path planning and motion coordination in multiple mobile robot teams*, Encyclopedia of Complexity and System Science, Springer (2009).

[2] C. Goerzen , Zhaodan Kong, and Bernard Mettler. *A survey of motion planning algorithms from the perspective of autonomous UAV guidance*, Journal of Intelligent and Robotic Systems, Vol. 57. No. 1–4 (pp. 65-100, 2010.

[3] Preetha Bhattacharjee, P. Rakshit, I. Goswami, A. Konar, and, A. K. Nagar, *Multi-robot path-planning using artificial bee colony optimization algorithm*, Third World Congress on Nature and Biologically (2011).

[4] Matthew Turpin, Nathan Michael, and Vijay Kumar. *Concurrent assignment and planning of trajectories for large teams of interchangeable robots*, International Conference on Robotics and Automation (ICRA) (2013).

[5] MATLAB, http://www.mathworks.com/.

[6] Hungarian Algorithm for Linear Assignment Problems (V2.3), http://www.mathworks.com/matlabcentral/fileexchange/20652-hungarian-algorithm-for-linear-assignment-problems--v2-3-