Multi-Robot Coordination: C-CAPT and Modified CAPT Extended Algorithm Implementation

| Jay Davey  School of Engineering and  Applied Science  University of Pennsylvania jay.davey87@gmail.com | Eduardo Garcia  School of Engineering and  Applied Science  University of Pennsylvania edgarc@seas.upenn.edu | Caio Mucchiani  School of Engineering and  Applied Science  University of Pennsylvania caio@seas.upenn.edu |
| --- | --- | --- |

*Abstract*— Multi-robot coordination is one interesting application for micro-UAVs. In this sense, this paper includes the implementation of the state-of-the-art algorithms to coordinate the assignment of N-robots from an initial position S to a goal final position G. Initially, the linear assignment problem is done with the Hungarian algorithm implementation, and for the second part the CAPT algorithm is extended to the implementation of the Goal Assignment and Trajectory Planning algorithm (GAP) .

*Index Terms*—CAPT, multi-robot, GAP,C-CAPT.

Introduction

Multi-robot assignment tasks represent a challenge for the actual robotic scenario, and it is a problem that should be solved efficiently. In this paper, we use the algorithms discussed in [1] and [2] and the approach to the multi-robot assignment problem is done in two different manners: the first phase consists of an implementation of the centralized CAPT algorithm (C-CAPT) for circular first order robots that operates in a 2D obstacle free environment. For this phase, we discuss the algorithm in terms of runtime against number of robots (which varies from 10 up to 1000), with equal number of goals.

For the second phase, …

# C-Capt Algorithm

The concurrent assignment and planning of trajectories (CAPT) algorithm is defined considering an N multi-robot task assignment problem in which the robots are unlabeled (or it does not matter who and where robots are assigned) and assigned to M goal locations. The proposed modification of this problem consists of an obstacle-free environment and combines the subproblems of assignment and trajectory generation, seeking to provide a computationally tractable solution for large number of robots. This centralized version of the CAPT algorithm is known as a centralized CAPT (or C-CAPT).

# C-Capt for first order robots

As the first part of this paper, we implemented the C-CAPT algorithm assuming a 2D obstacle free workspace filled with circular first order robots. Under this scenario, and following [1] a few assumptions are made:

**(A1)** Interchangeable and homogeneous robots, and no preference for goal location among them

**(A2)** Circular robots with radius R

**(A3)** Obstacle free region (defined as

**(A4)** Random initial locations for the robots

**(A5)** Random locations for the goals

**(A6)** Robots fully actuated and perfect state knowledge

(A7) The robot stage area of flight is defined in

We define the space as an obstacle free region and as the Minkowski sum of the circular robot and the union of the initial and final location. The formal definition will not be addressed here since it can be found on [1]. The robot stage area represents a 2D area defined as follows (dimensions are in meters):

;

;

We also define the robot as a circle of radius R = 0.08 meters.

# Implementation of C-Capt for phase 1

Based on assumptions from A1 to A8, we implemented the C-CAPT using MATLAB. Since the interest of this Phase focus on runtime for this algorithm, the number of robots was varied according to the following vector:

Also, the number of trials for each number of robots was

## Start and goal location

The assignment for the start and goal position for this phase was done randomly within the limits of the number of robots and goals.

## Hungarian Algorithm

For the solution of the linear assignment problem, the Hungarian algorithm was implemented following the steps from [5]:

* **Step 0**: For a matrix (the cost matrix), where each element represents the cost assignment of one robot from a start to a goal position, we rotate the matrix so that the number of columns is equal or higher than the number of rows. Let k be the minimum value of the number of rows and columns;
* **Step 1:** For each element in the cost matrix, look for the smallest and subtract it from every present element in its row; Go to step 2.
* **Step 2:** Find a zero in the resulting matrix, and mark its row or column in case it has not being marked. Repeat this for each element in the cost matrix; Go to step 3.
* **Step 3:** Cover each marked column from Step 2. If k columns are covered, this corresponds to a complete assignment and the algorithm is done; otherwise, go to step 4.
* **Step 4:** Find an uncovered zero and prime it. In case there is no marked zero in the row contained this primed zero, go to step 5. Otherwise, cover this row, uncover the column containing the starred zero. Continue this process until there are no uncovered zeros left and save the smallest uncovered value; Go to step 6.
* **Step 5**:Alternate primes and zeros as:

Continue until the series terminates at a primed zero that has no marked zero in its column. Unmark each starred zero of the series, mark each primed zero of the series, erase all primes and uncover every line in the matrix; Return to Step 3.

* **Step 6:** Add the minimum uncovered value to every element of each covered row, and subtract it from every element of each uncovered column. Return to Step 4 without altering any marked, primes, or covered lines.

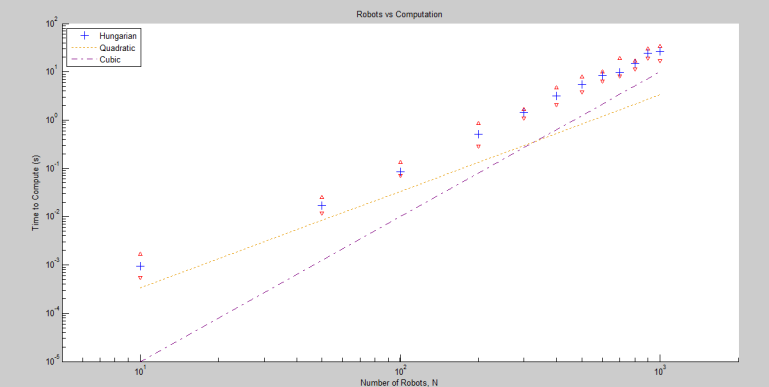
This algorithm was implemented on Matlab and the following results from this implementation.

## Runtime for CAPT implementation

For the implementation of the CAPT algorithm, start and goal locations for the 2D environment were generated randomly and the cost matrix defined as:

where *A* represents the cost matrix.

U sing Matlab, the implementation of the CAPT algorithm for N robots varying from 10 to 1000, navigating to the same number of goals in each case was done by following the process described in B. The result for the implementation (which can be found attached to this document) is as follows:



1. Runtime for Quadratic, Hungarian and Cubic methods for multiple robot assignment

We can see from the graph that the Hungarian method proved to be the most efficient one in terms number of robots against runtime. An interesting point in the graph is represented by the intersect between the cubic and the quadratic methods: for a number of robots greater than 500, the cubic method becomes more advantageous option and substitute for the quadratic one.

## Some Common Mistakes

# Implementation of the extended capt algorithm through the goal assignment and trajectory planning

Enter text

## Authors and Affiliations

Enter text

## Identify the Headings

Enter text

## Figures and Tables

Enter text

## Footnotes

Enter text

1. Table Type Styles

| Table Head | Table Column Head | | |
| --- | --- | --- | --- |
|  | Table column subhead | Subhead | Subhead |
| copy | More table copya |  |  |

a. Sample of a table footnote. *(table footnote)*

1. Example of a figure caption. *(figure caption)*

INSERT FIGURES HERE

# Acknowledgment

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

# References

1. M. Turpin, N. Michael and V. Kumar “CAPT: Concurrent assignment and planning of trajectories for multiple robots,” The International Journal of Robotics Research 2014,Vol 33(1)
2. M. Turpin ,K. Mohta , N. Michael and V. Kumar : “Goal Assignment and Trajectory Planning for Large Teams of Aerial"
3. W. Burgar, M. Moors, C. Stachniss, F.Schneider : “Coordinated Multi-Robot Exploration” University of Bonn- Germany
4. L. Luo, N. Chakraborty, K. Sycara: “Multi-Robot Assignment Algorithm for Tasks with Set Precedence Constraints”
5. “Munkres' Assignment Algorithm Modified for Rectangular Matrices”: available at <http://csclab.murraystate.edu/bob.pilgrim/445/munkres.html>