2022 30th Mediterranean Conference on Control and Automation (MED)
June 28 - July 1, 2022. Athens, Greece

2022 30th Mediterranean Conference on Control and Automation (MED) | 978-1-6654-0673-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/MED54222.2022.9837240

Multi-Robot Task Allocation Framework with Integrated Risk-Aware 3D Path Planning

Yifan Bai, Björn Lindqvist, Samuel Karlsson, Christoforos Kanellakis and George Nikolakopoulos

Abstract— **This article presents an overall system architecture for multi-robot coordination in a known environment. The proposed framework is structured around a task allocation mechanism that performs unlabeled multi-robot path assignment informed by 3D path planning, while using a nonlinear model predictive control(NMPC) for each unmanned aerial vehicle (UAV) to navigate along its assigned path. More specifically, at first a risk aware 3D path planner D^*_+ is applied to calculate cost between each UAV agent and each target point. Then the cost matrix related to the computed trajectories to each goal is fed into the Hungarian Algorithm that solves the assignment problem and generates the minimum total cost. NMPC is implemented to control the UAV while satisfying path following and input constraints. We evaluate the proposed architecture in Gazebo simulation framework and the result indicates UAVs are capable of approaching their assigned target whilst avoiding collisions.**

I. Introduction

Taking advantage of agility in three-dimensional space, Unmanned Aerial Vehicles(UAVs) have shown influential impact in many application aspects, including search and rescue [1], infrastructure inspection [2], area mapping [3], etc. For decades, one of the most popular areas of aerial robotics is the case of multiple UAVs systems, which commits to having a group of UAVs performimng some collective behavior so as to improve efficiency and robustness. While deploying a UAV fleet is an extremely complex problem, which requires solving several sub-problems: (1) Task assignment problem, namely addressing the question of how a set of tasks should be allocated to a set of robots so as to achieve the overall system goals; (2) Coordination issue, that is how the robots negotiate and coordinate; (3) Path planning, finding trajectories for robots to complete the tasks; (4) Control scheme, which enables robots follow its assigned trajectory.

A. Related works

Considering the complexity of deploying multiple robotic system, the assignment-planning problem is often decoupled into two subproblems. The first subproblem is the multi-robot task allocation (MRTA), an NP (non-deterministic polynomial-time) hard optimization problem [4]. Mathematically, MRTA can be modeled as the optimal assignment problem (OAP) [5], or the multiple traveling salesman problem (mTSP) [6]. Numerous methods have been developed to solve MRTA problems, mainly categorized in optimization-based and market-based approaches. Bellingham [7] solved the task allocation problem using mixed integer linear programming (MILP) solution and the proposed solution performed well for a heterogeneous robotic fleet in a dynamic environment whilst satisfying various constraints. Other optimization methods [8], such as genetic algorithm, ant colony algorithm, simulated annealing algorithm have also been used to solve task allocation problems. Regarding market-based approaches, a distributed sequential auction scheme that takes the UAV communication range limitations into account and a systematic procedure for the auction process were introduced in [9]. The second subproblem is path planning. Decades of development has witnessed the maturity of single agent path planning research. Graph-based methods, such as the Dijkstra algorithm, the A^* algorithm, and the D^*lite algorithm [10], sample-based method like Rapidly Exploring Random Trees (RRT) [11] and Probabilistic Road Map, mathematics model based methods and bio-inspired methods have been widely used in various scenarios. In the recent years, more effort has been put on multi-agent path finding (MAPF). When the number of agents is relatively small and the task is to find an optimal, minimal-cost solution, MAPF can be formalized as a global, single-agent search problem based on A^* [12]. Bennewitz [13] presented a decoupled and prioritized method, thereby avoiding combinatorially hard planning problems typically faced by centralized approaches. State-of-art Conflict Based Search (CBS) algorithm [14] performs a search on a conflict tree, it can handle hundreds or even thousands of agents to find an optimal solution of MAPF.

In contrast with decoupled solutions, there are also some research combining the target assignment and path finding problem. Ma [15] presented a hierarchical algorithm, which has outstanding performance in terms of large scale robot teams. Turpin [16] proposed a concurrent assignment and trajectory planning strategy, including a centralized solution that minimize a cost functional based on square of velocity and a decentralized solution, which allows reassignment and re-planning. The simulation results demonstrate the centralized algorithm that offers globally optimal trajectories, while the decentralized algorithm yields sub-optimal but safe trajectories. However, the centralized solution is accomplished under the assumption that the convex hull of initial locations and goal positions with the Minkovski sum of a ball of a certain radius is collision-free, which means there are no

The authors are with the Robotics and AI Team, Department of Computer, Electrical and Space Engineering, Luleå University of Technology, Luleå SE-97187, Sweden

Corresponding Author's Email yifan.bai@ltu.se

This work has been partially funded by the European Unions Horizon 2020 Research and Innovation Programme under the Grant Agreements No. 101003591 NEX-GEN SIMS and No. 869379 illuMINEation.

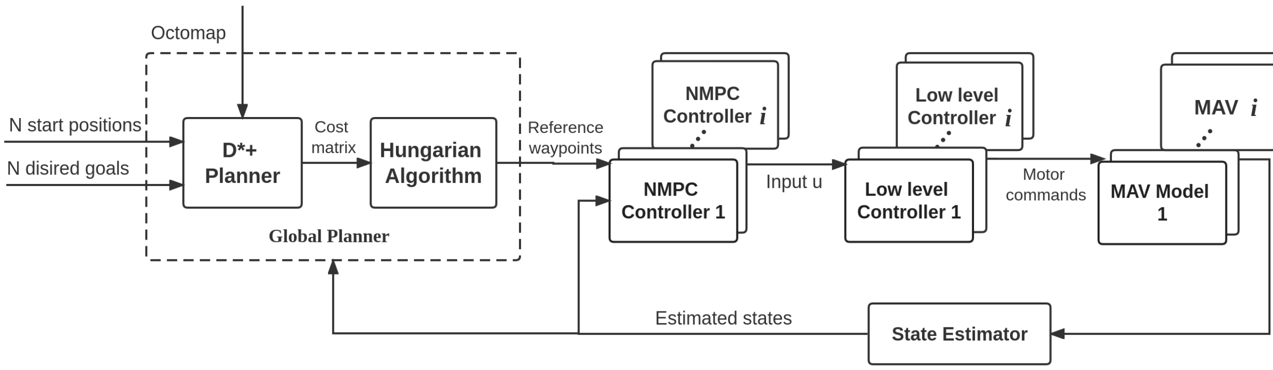

Fig. 1: Block diagram of the assignment-planning-control architecture. The dashed part highlights the contribution in the global planner, while the overall scheme demonstrates the integration of the global planner with the NMPC control of each aerial platform.

obstacles between start and goal regardless of the assignment, is unrealistic in the real world.

B. Contribution

In contrast to the previous state of the art, this article proposes a holistic architecture to solve the multi-robot task assignment, planning and control for holonomic UAV fleet. The main contribution is a generally applicable optimal assignment method based on the combination of the Hungarian algorithm and a 3D risk-aware path planner, which constitute the global planner of the holonomic multiple robots system.

The rest of the article is organized as follows. We begin with the mathematical formulation of the assignment problem in Section II. Section III introduces the integrated framework for the task assignment, path planning and nonlinear model predictive control for multi-robot system. Section IV demonstrates a Gazebo simulation result that proves the feasibility of proposed scheme. Finally, Section V is the conclusion and future works.

II. Problem Description

We consider N aerial robots navigating from their initial positions to N desired goal positions in a known 3-dimensional environment, indexed by 1, ... ,n.

We define the cost matrix $C \in \mathbb{R}^{N\times N}$, the element of which $c_{i,j}$ represents the traversal cost of generated path for the robot i traveling from its initial position to the goal j. The assignment matrix $\varphi \in \mathbb{R}^{N\times N}$, which assigns robots to goals:

$$\varphi_{i,j} = \begin{cases} 1 & \text{if robot } i \text{ is assigned to goal } j \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Since each goal is required to be assigned to a different robot, we have

$$\varphi^{\mathrm{T}}\varphi = I_N \tag{2}$$

Where I_N is the N by N identity matrix.

The assignment-planning module seeks to find an optimal assignment and planning strategy that minimizes the sum of distance traveled by each individual robot, which is equivalent to

$$\begin{aligned} &\underset{\varphi}{\textbf{minimize}} \sum_{i=1}^{n}\sum_{j=1}^{n} \varphi_{i,j} C_{i,j} \\ &\text{subject to (1), (2)} \end{aligned} \tag{3}$$

To simplify the problem, the following assumptions are made:

- All robots are homogeneous and with no preference of goals.
- The initial positions and goal positions do not overlap.
- There is not any external disturbance or actuation error for the robotic systems.

III. Methodology

To tackle the problem described in Section II, we propose an exhaustive autonomy architecture composing of a 3D path planning, a task assignment and control module, as shown in Figure 1. First, taking N start positions, N goal positions and 3D occupancy map as input, D_+^* planner enumerates every start-goal combination and outputs corresponding path and traversal cost. By means of mathematical transformations of the cost matrix, the Hungarian algorithm can assign goals to each agent with a minimum cost. Finally, NMPC enables each UAV to follow its predetermined collision free path to the goal position. The rest of this Section elaborates on each module of this architecture.

A. 3D Path Planning

In order for the aerial robot to plan a 3D global path in a known environment, a 3D occupancy map is required to represent the three-dimensional volumetric space of the environment. In this work, Octomap [17], an efficient probabilistic 3D grid mapping framework based on octrees is used, which represents 3D models that include free, occupied and unknown areas. Each node in an octree that stores the occupancy probability of a voxel, which can be updated by the latest sensor measurement. The state (free, occupied and unknown) of a voxel depends on whether its occupancy probability exceeds a predetermined threshold. And the children of a node can be pruned if all of them have the same state, resulting in a substantial memory consumption. Octomap is more powerful in terms of memory usage, future updates compared to point cloud and fixed grid 3D representation.

As for path planning, there are a plenty of options. A^* is a best-first search algorithm that aims to find a smallest cost path based on a heuristic function. D^*lite [10] is an incremental heuristic search algorithm, which outperforms A^* in terms of capability of replanning when traversing unknown maps. However, D^*lite ignores the physical shape of robots and considers them as particles, which may lead to waypoints adjacent to obstacles and potential collisions. D^*_+ [18] tackles this problem by assigning different traversal cost for free, occupied and unknown voxels and adding a risk layer that increases the traversal cost for voxels in the proximity of occupied voxels. In this case, the planner will generate a moderate path with a safety margin next to obstacles. Thus, we select D^*_+ as our path planner.

B. Hungarian Algorithm

In this article, we focus on the linear balanced assignment problem, in which the number of agents and tasks are equal. The naive solution to find the minimal total cost is to enumerate all possible assignments and calculate the cost of each one, yet listing $n!$ assignments for n agents and n tasks is inefficient, with the computational complexity of $O(n!)$. The Hungarian Algorithm is capable of solving this problem with the computational complexity $O(n^3)$ [5]. The pseudocodes cost matrix generation and the Hungarian algorithm are depicted in Algorithm 1 and Algorithm 2. Some matrix operation functions to be clarified are the following ones. The $D^*_+.ComputeCost(A(i), T(j))$ is a function of D^*_+ that calculates the minimum traversal cost from position A_i to position T_j; the $MarkMatrix$ function marks as few rows and columns as possible to cover all zeros in the input matrix and returns two lists r_m c_m containing indices of marked rows and marked columns respectively; the $AdjustMatrix$ function takes current matrix λ and r_m, c_m as input, subtracts the lowest unmarked element from every unmarked elements and adds it to the elements that are marked twice.

Algorithm 1 Cost Matrix Generation

Input : A : List that stores all initial positions
T : List that store all goal positions
n : Number of the agents
Output : α : Cost matrix

Declare α, cost matrix of n by n doubles
for all $i = 1$ **to** n **do**
  for all $j = 1$ **to** n **do**
    $\alpha(i,j) = D^*_+.ComputeCost(A(i), T(j))$
  end for
end for

C. Nonlinear Model Predictive Control

By virtue of the 3D path planning and the Hungarian algorithm, each drone has a predetermined path to reach the assigned goal position. In the course of trajectory tracking, we use a nonlinear model predictive controller, which has the ability of anticipate future events and take actions accordingly and has been successfully used for UAV as [19] and [20].

Algorithm 2 The Hungarian Algorithm

Input : α : Cost matrix
Output : λ : Assignment matrix

$\lambda = \alpha$
for all $i = 1$ **to** n **do**
  $\lambda(i,:) = \lambda(i,:) - min(\lambda(i,:))$ {subtract row minima for all elements in the row}
end for
for all $i = 1$ **to** n **do**
  $\lambda(:,j) = \lambda(:,j) - min(\lambda(:,j)$ {subtract column minima for all elements in the column}
end for
Declare a integer $\sigma = 0$ that counts the marked zeros
Declare a list $r_m = []$ that will store the marked rows of λ
Declare a list $c_m = []$ that will store the marked columns λ
while $\sigma < size(\lambda, 0)$ **do**
  $r_m, c_m = MarkMatrix(\lambda)$
  $\sigma = len(r_m) + len(c_m)$
  if $\sigma < size(\lambda, 0)$ **then**
    $\lambda = AdjustMatrix(\lambda, r_m, c_m)$
  end if
end while

The nonlinear model [21] of the MAV system is shown as follows:

$$
\begin{aligned}
\dot{\mathbf{p}}(t) &= \mathbf{v}(t) \\
\dot{\mathbf{v}}(t) &= \mathbf{R}(\psi,\theta,\phi)\begin{pmatrix}0\\0\\T\end{pmatrix} + \begin{pmatrix}0\\0\\-g\end{pmatrix} - \begin{pmatrix}A_x & 0 & 0\\ 0 & A_y & 0\\ 0 & 0 & A_z\end{pmatrix}\mathbf{v}(t) \\
\dot{\phi}(t) &= \frac{1}{\tau_\phi}\left(K_\phi \phi_d(t) - \phi(t)\right) \\
\dot{\theta}(t) &= \frac{1}{\tau_\theta}\left(K_\theta \theta_d(t) - \theta(t)\right)
\end{aligned}
\tag{4}
$$

Where $p = [p_x, p_y, p_z]^\top$ and $v = [v_x, v_y, v_z]^\top$ are the position and velocity of the UAV respectively, R is the rotation matrix of the body frame B in the fixed inertial frame W expressed in frame W, T is the thrust produced by rotors, g is the gravitational acceleration, A_x, A_y, A_z indicate drag coefficients, τ_ϕ, K_ϕ and τ_θ, K_θ are the time constant and gain of inner-loop behavior for roll angle ϕ and pitch angle θ respectively. The state of the model is $x = [p, v, \phi, \theta]^\top$ and the system input is $u = [T, \phi_d, \theta_d]^\top$.

By discretization of (4) with the sampling time T_s it gives the prediction form (5):

$$x_{k+1} = f(x_k, u_k) \tag{5}$$

In the NMPC approach, an optimal control problems is solved iteratively on a finite prediction horizon N. The states and control inputs j steps ahead of the current time step k are denoted as $x_{k+j|k}$ and $u_{k+j|k}$. At each time step, NMPC calculates an optimal sequence of control actions $[u_{k|k}, ..., u_{k+N-1|k}]$ that minimize the predetermined cost function and applies the first control action to the controller.

In the sequel we introduce a cost function J, that penalizes the deviation of predicted position from reference position, the deviation of input from reference hovering input, the successive changes in control action.

$$J = \sum_{j=0}^{N-1} \Big(\underbrace{\left\| x_{\text{ref}} - x_{k+j|k} \right\| {Q_x}^2}_{\text{position error}} + \underbrace{\left\| u_{\text{ref}} - u_{k+j|k} \right\| {Q_u}^2}_{\text{Input penalty}} + \underbrace{\left\| u_{k+j|k} - u_{k+j-1|k} \right\| {Q_{\Delta u}}^2}_{\text{Input change penalty}} \Big) \tag{6}$$

where $Q_x \in \mathbb{R}^{8\times 8}, Q_u, Q_{\Delta u} \in \mathbb{R}^{3\times 3}$ are positive definite weight matrices that reflect relative importance of each term in cost function.

IV. Simulation Results

To evaluate the performance of the proposed scheme, we generated a custom world in the Gazebo simulation framework, where several obstacles in the shape of cylinders and cubes are enclosed by a square wall structure with a length and width of 18 meters and a height of 3.5 meters. Initially, the pipeline requires the known 3D occupancy map of the environment, which without loss of generality has been generated by the collected onboard sensor data from previous simulation runs. The resolution of the Octomap is set to 0.4, to facilitate computation of building search graph in D^*_+ planner. The Gazebo world and the resulting 3D occupancy map are shown in Figure 2 and Figure 3 respectively.

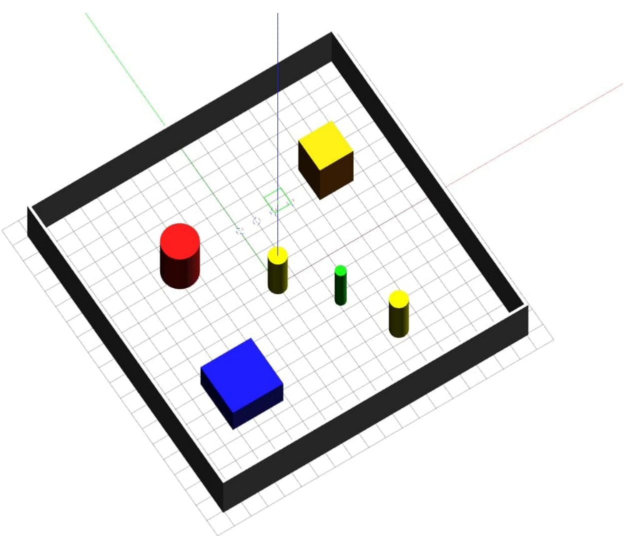

Fig. 2: Virtual world with obstacles in Gazebo simulation environment.

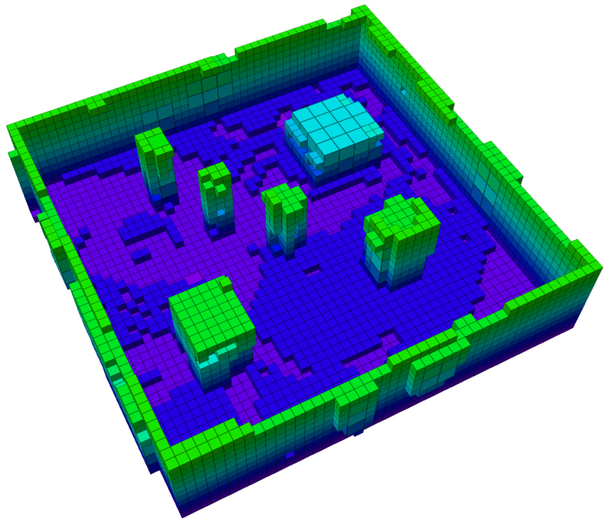

Fig. 3: Generated Octomap of the virtual world.

Lets consider a scenario where four tasks are assigned to four agents. The start positions of four agents are $A_1 = [0.0, 3.5, 0.0]$, $A2 = [1.0, 3.5, 0.0]$, $A3 = [2.0, 3.5, 0.0]$, $A4 = [3.0, 3.5, 0.0]$, and four goal positions are chosen randomly as long as they are neither outside the map nor occupied: $T1 = [-4.5, -7.0, -1.5]$, $T2 = [-7.5, 3.5, 2.3]$, $T3 = [6.5, -5.0, 1.0]$, $T4 = [8.0, 8.0, 0.8]$.

D^*_+ planner takes the generated Octomap as a map input and the safety distance is set to $r = 2$ voxels. Then, we enumerate combinations of start position of agents and goal positions and publish them to the D^*_+ planner and the paths and traversal costs are obtained. As shown in Table I, a cost matrix is built for this assignment. The assignment result is depicted in Table II that displays the minimum total cost of UAVs way to assign the goals: the first goal T_1 is assigned to agent 2 A_2, T_2 is allocated to $A1$, and T_3 T_4 are allocated to A_3 A_4 respectively. With the stored paths from four starting positions to corresponding goal positions, we set out to tune the NMPC parameters for the follow-up UAV control.

TABLE I: Cost matrix of UAVs approaching tasks

Cost \ Target / Agent	T_1	T_2	T_3	T_4
A_1	66.0915	46.5344	56.5262	50.3840
A_2	67.7483	50.5344	54.8693	50.3840
A_3	70.2337	56.5344	51.1127	42.7272
A_4	71.8905	60.5344	49.4558	41.0703

For the NMPC parameter tuning, Q_x = diag(6, 6, 40, 2, 2, 3, 8, 8), Q_u = diag(3, 10, 10), $Q_{\delta u}$ = diag(3, 15, 15). The NMPC prediction horizon is N = 20 with a sampling time of 50ms, indicating NMPC predicts states of the UAV within one second. The NMPC scheme is implemented in Optimization Engine (OpEn), an open-source code generation software for embedded nonlinear optimization, which is fully ROS-integrated [22].

TABLE II: Assignment result of the Hungarian Algorithm

Cost / Target / Agent	T_1	T_2	T_3	T_4
A_1	0	46.5344	0	0
A_2	67.7483	0	0	0
A_3	0	0	51.1127	0
A_4	0	0	0	41.0703

The trajectories of the four agents are shown in Figure 4, where the black dots indicate the initial positions of the four UAVs and the four red stars denote the four goal positions. Clearly all four agents are able to reach in the vicinity of assigned goal positions. Furthermore, we can observe that some UAVs do not reach the desired 3D coordinates precisely. This is acceptable since the goal point is regarded as reached if it is approached within 0.4m from the target waypoint.

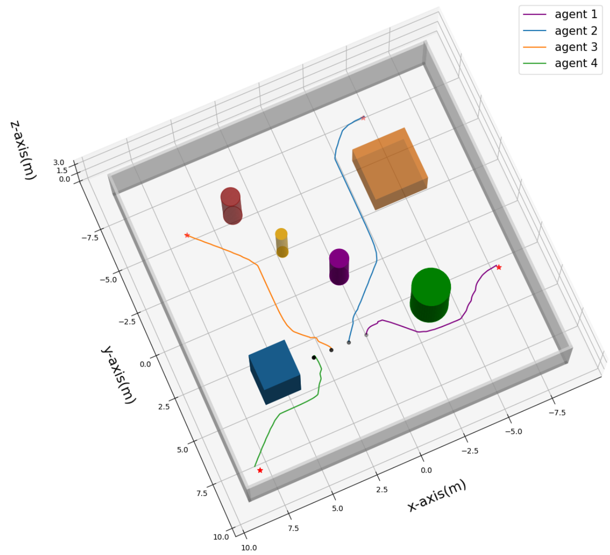

Fig. 4: Followed paths from the four agents during the first simulation. The highlighted stars denote the goal positions.

Figure 5 depicts the second simulation with different initial positions and different goal positions. Notably, agent A_1 is the closest agent to the goal T_2 in Euclidean distance (shown as grey dot line), while the Hungarian algorithm assigned goal T_2 to A_2 considering the minimum overall traversal cost.

V. Conclusions and Future Work

In this article, we proposed a novel and complete assignment, planning and control architecture for multiple-goal-multiple-UAVs system, which integrates D^*_+ algorithm, the Hungarian algorithm and the nonlinear model predictive controller for 3D path planning, task assignment and control respectively. The architecture was applied to a team of four UAVs in a known environment with four random goal positions. The Gazebo simulation result demonstrated that it is possible to find a one goal to one UAV assignment that

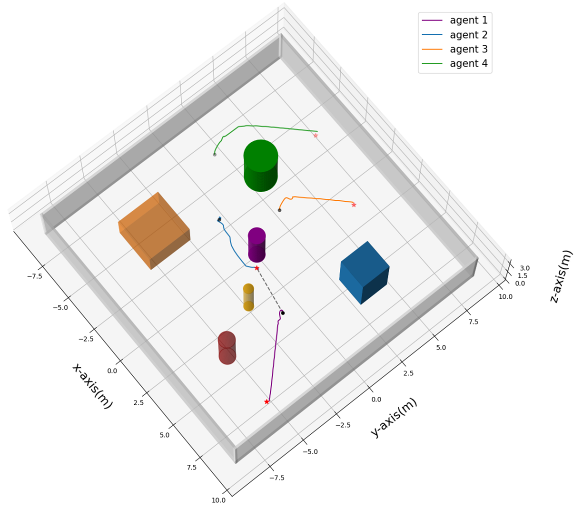

Fig. 5: Followed paths from the four agents during the second simulation. The highlighted stars denote the goal positions.

minimize the total traversal cost and four agents successfully follow the collision free path to reach the goal positions. As a clear direction for future work we are consider to apply this framework to multi-robot inspection and objects pick-up relocation using multiple agents.

Future work concerns extending the assignment problem to the case where the number of agents and the number of tasks are different. It will also be interesting to think over the assignment problem for heterogeneous robots.

References

[1] S. Hayat, E. Yanmaz, T. X. Brown, and C. Bettstetter, "Multi-objective uav path planning for search and rescue," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 5569–5574.

[2] S. S. Mansouri, C. Kanellakis, E. Fresk, D. Kominiak, and G. Nikolakopoulos, "Cooperative coverage path planning for visual inspection," *Control Engineering Practice*, vol. 74, pp. 118–131, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0967066118300315

[3] P. Schmuck and M. Chli, "Multi-uav collaborative monocular slam," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 3863–3870.

[4] B. P. Gerkey and M. J. Matarić, "A formal analysis and taxonomy of task allocation in multi-robot systems," *The International Journal of Robotics Research*, vol. 23, no. 9, pp. 939–954, 2004. [Online]. Available: https://doi.org/10.1177/0278364904045564

[5] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[6] T. Bektas, "The multiple traveling salesman problem: an overview of formulations and solution procedures," *omega*, vol. 34, no. 3, pp. 209–219, 2006.

[7] J. Bellingham, M. Tillerson, A. Richards, and J. P. How, "Multi-task allocation and path planning for cooperating uavs," in *Cooperative control: models, applications and algorithms*. Springer, 2003, pp. 23–41.

[8] M. Badreldin, A. Hussein, and A. Khamis, "A comparative study between optimization and market-based approaches to multi-robot task allocation." *Advances in Artificial Intelligence (16877470)*, 2013.

[9] M. B. Dias, R. Zlot, N. Kalra, and A. Stentz, "Market-based multirobot coordination: A survey and analysis," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1257–1270, 2006.

[10] S. Koenig and M. Likhachev, "d^* lite," *Aaai/iaai*, vol. 15, pp. 476–483, 2002.

[11] S. M. LaValle *et al.*, "Rapidly-exploring random trees: A new tool for path planning," 1998.
[12] T. Standley, "Finding optimal solutions to cooperative pathfinding problems," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 24, no. 1, 2010, pp. 173–178.
[13] M. Bennewitz, W. Burgard, and S. Thrun, "Finding and optimizing solvable priority schemes for decoupled path planning techniques for teams of mobile robots," *Robotics and autonomous systems*, vol. 41, no. 2-3, pp. 89–99, 2002.
[14] G. Sharon, R. Stern, A. Felner, and N. R. Sturtevant, "Conflict-based search for optimal multi-agent pathfinding," *Artificial Intelligence*, vol. 219, pp. 40–66, 2015.
[15] H. Ma and S. Koenig, "Optimal target assignment and path finding for teams of agents," *arXiv preprint arXiv:1612.05693*, 2016.
[16] M. Turpin, N. Michael, and V. Kumar, "Capt: Concurrent assignment and planning of trajectories for multiple robots," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 98–112, 2014. [Online]. Available: https://doi.org/10.1177/0278364913515307
[17] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous robots*, vol. 34, no. 3, pp. 189–206, 2013.
[18] S. Karlsson, A. Koval, C. Kanellakis, A.-a. Agha-mohammadi, and G. Nikolakopoulos, "d^*_+: A generic platform-agnostic and risk-aware path planing framework with an expandable grid," *arXiv preprint arXiv:2112.05563*, 2021.
[19] B. Lindqvist, S. S. Mansouri, A.-a. Agha-mohammadi, and G. Nikolakopoulos, "Nonlinear mpc for collision avoidance and control of uavs with dynamic obstacles," *IEEE robotics and automation letters*, vol. 5, no. 4, pp. 6001–6008, 2020.
[20] B. Lindqvist, P. Sopasakis, and G. Nikolakopoulos, "A scalable distributed collision avoidance scheme for multi-agent uav systems," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 9212–9218.
[21] M. Kamel, T. Stastny, K. Alexis, and R. Siegwart, *Model Predictive Control for Trajectory Tracking of Unmanned Aerial Vehicles Using Robot Operating System*. Cham: Springer International Publishing, 2017, pp. 3–39. [Online]. Available: https://doi.org/10.1007/978-3-319-54927-9_1
[22] P. Sopasakis, E. Fresk, and P. Patrinos, "Open: Code generation for embedded nonconvex optimization," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6548–6554, 2020.