An MDP-based Approximation Method for Goal Constrained Multi-MAV Planning under Action Uncertainty

Lantao Liu[1] and Nathan Michael[2]

***Abstract*— This paper presents a fast approximate multi-agent decision theoretic planning method extended from the well-known Markov Decision Process (MDP). Our objective is to plan motions for a team of homogeneous micro air vehicles (MAVs) toward a set of goals, such that each MAV at any state at any moment follows an action policy toward a unique goal, while considering action uncertainty. We pursue an efficient formulation by first considering a deterministic abstraction of the stochastic system based on approximate initial paths. These deterministic and decoupled sub-problems are converted to the stochastic domain and improved by individual agents or a subset of agents. The resulting decoupled formulation requires processing of a partial state space and enables online operation given applications with emerging tasks.**

I. Introduction and Related Work

A challenge when deploying teams of small-scale micro air vehicles (MAVs) is consideration of the action uncertainty that can arise due to environmental disturbances with implication on the ability of the team to follow the planned path. This paper tackles *motion planning* and *goal assignment* for multi-MAV systems while considering stochasticity of uncertain action outcomes. Specifically, we develop a decision theoretic planning method that guides a team of homogeneous MAVs with action uncertainty to avoid obstacles and eventually reach mutually exclusive goals.

Motion (or trajectory) planning for autonomous robots has been well studied [9, 17]. Deterministic path planners (e.g., A* or Dijkstra's based algorithms) are efficient in terms of computational performance; and many probabilistic path planners (e.g., RRT or PRM based methods) perform well in exploring high dimensional space [11, 12, 22]. However, these methods do not always account for uncertainty and therefore do not include the cost or penalty induced from unexpected actions. Moreover, methods unaware of uncertainty do not consider the MAVs' deviation from planned nominal paths. Arriving at any locations off the paths may require a re-planning process if global optimality needs to be guaranteed.

An extremely powerful tool, the Markov Decision Process (MDP) has been widely-adopted for modeling autonomous decision-making under uncertainty [1, 21]. However, the MDPs become more complex and computationally expensive when multiple agents need to be coordinated free of conflicts, which are termed multi-agent MDPs (MMDPs) [3]. In order to produce a collective behavior that accurately maximizes the total expected reward for all agents, the most popular way to model an MMDP is to expand the action space to a form of joint action space contributed by all agents [3, 18, 20], which requires much more computational effort than the single agent case. Works that are related to this presented approach also include the Decentralized MDPs (Dec-MDPs) or more generally the Decentralized Partially Observable MDPs (Dec-POMDPs) [2, 6]. A majority of efforts have been focused on solving the interactions among agents, assuming limited/indirect communication [8], imperfect and local observations [5, 19], or constrained agent behaviors [15, 18].

Different from approaches mentioned above, the objective of our work is to generate action uncertainty aware policies leading homogeneous MAVs to unique goals. Consequently, the optimal solution is subject to several constraints: control policy optimization, goal assignment (goal constraints), action uncertainty, and collision avoidance. Combining all these constraints into one bulk mathematical program and computing accurate solutions require examining all stochastic outcomes over some (possibly infinite) horizon, which can be computationally prohibitive.

One way to formulate the goal-constrained planning under uncertainty problem is to unify the MDPs with task allocation (assignment) mechanisms. For example, approaches of [7, 10] formulate and address the problem in the fashion of allocating distributed or loosely-coupled MDPs. In order to adapt to the MDP model, the transition probability of [4] is used to model the assignment uncertainty, i.e., the probability distribution of assigned tasks in future. However, in such a model neither action uncertainty nor trajectory planning is considered, which is exactly the problem we wish to tackle.

In this paper, we are proposing a new approximation framework for MMDPs. Our major objective is to improve the computational efficiency for online application purposes, without expanding neither the state space nor action space (i.e., we do not use joint action space). In this way, computational complexity is maintained at the level of the single agent planning case. We achieved the efficiency by first abstracting the stochastic problem into a deterministic counterpart and obtaining an initial solution to decompose the problem. Then the sub-problems are transformed back to stochastic domain and solved by individual agents. We validate important performances in simulation and show that this approach is practically very fast and thus suitable for real-time multi-agent systems and online applications with emerging tasks.

[1]Lantao Liu is with the Department of Computer Science at the University of Southern California, Los Angeles, CA 90089, USA. lantao.liu@usc.edu.

[2]Nathan Michael is with the Robotics Institute at Carnegie Mellon University, Pittsburgh, PA 15213, USA. nmichael@cmu.edu

The work was conducted at Carnegie Mellon University. The authors gratefully acknowledge the support of ARL grant W911NF-08-2-0004.

II. Research Background and Preliminaries

Let $\mathcal{R}$ be the set of MAVs and $\mathcal{G}$ be the set of goals, and assume $|\mathcal{R}| \leq |\mathcal{G}|$. With awareness of action uncertainty, we wish to deploy the robots following certain control policies such that no two MAVs share the same goal. We assume fully observable states, therefore, such a planning framework can be perfectly modelled with an MDP, as follows.

A. Markov Decision Process (MDP)

***Definition** 2.1:* An MDP $\mathcal{M}$ is defined by a 4-tuple $\mathcal{M} =< S, A, T, C >$, where

- S is a countable set of states s.
- A is a countable set of stochastic actions a.
- $T_a(s, s') = \Pr(s_{t+1} = s' | s_t = s,\ a_t = a)$ gives the transition probability that an agent moves to a state s' when it executes the action a from s .
- $C_a(s, s')$ is a non-negative cost for performing action a on s and reaching s'.

A control policy π is a complete mapping from states to actions so that the agent applies the action $a_t \in A$ in state $s_t \in S$ at time t. If the action is independent of time, the policy is called *stationary* and it is simply denoted by a, as assumed above. Starting from state s_0 for infinite time t, let the sequence of future actions be $\{a_1, a_2, \cdots, a_t, \cdots\}$ and the sequence of future states be $\{s_1, s_2, \cdots, s_t, \cdots\}$; then the total *value* (cost) for state s_0 over an infinite horizon can be expressed as

$$V(s_0) = \sum_{t=0}^{\infty} \gamma^t C_a(s_t, s_{t+1}), \tag{1}$$

where $a_t = \pi(s_t)$ and $\gamma \in [0, 1]$ is a discount factor for discounting future costs at a geometric rate.

***Definition** 2.2:* The **Q-value** of a state-action pair (s, a) is defined as the the one-step look-ahead *value* of state s if the immediate action a is performed:

$$Q(s, a) = \sum_{s' \in S} T_a(s, s')[C_a(s, s') + \gamma V(s')]. \tag{2}$$

The MDP is to find an optimal policy π^* satisfying

$$V_{\pi^*}(s) \equiv V^*(s) = \min_{a \in A} Q(s, a),\ \forall s \in S. \tag{3}$$

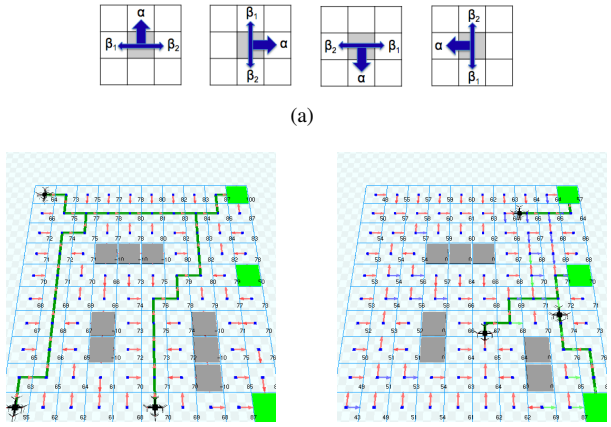
When $\gamma < 1$, there exists a stationary policy that is optimal. In this case, V^* is the unique solution to the Bellman Optimality equations:

$$V^*(s) = \min_{a \in A} \sum_{\forall s' \in S} T_a(s, s')[C_a(s, s') + \gamma V^*(s')]. \tag{4}$$

From Eq. (4), the optimal action policy $\pi^*(s)$ can be obtained

$$\pi^*(s) = \arg\min_{a \in A} \sum_{\forall s' \in S} T_a(s, s')[C_a(s, s') + \gamma V^*(s')]. \tag{5}$$

Employing Bellman's principle of optimality avoids enumerating solutions naively. In particular, value iteration (VI) and policy iteration (PI) are the most widely used dynamic programming strategies for solving MDPs. In this paper, we use VI as a substrate to develop a higher level multi-agent decision theoretic planning framework.



(a) (b) (c)

Fig. 1. (a) MAV action uncertainty model. Probabilities β_1, β_2 lead to undesired states; (b) With standard single-agent MDP, MAVs greedily choose their goals (green cells), causing conflicts; (c) Goal-oriented planning generate policies leading MAVs to different goal states.

B. Stochastic Shortest Path and Goal-Oriented Trajectories

The discrete-time MDP targeting cost minimization formulated in Sect. II-A can be extended to a *stochastic shortest path (SSP)* problem [16], with an extra set of assumptions from which the SSP derives its special properties.

***Assumption** 2.1:* The state set of SSP includes goal/terminal states. Each goal state $s_g \in S$ is zero-cost and absorbing. This means that $T_a(s_g, s_g) = \Pr(s_g | s_g, a) = 1$ and $C_a(s_g, s_g) = 0$, $\forall a \in A$.

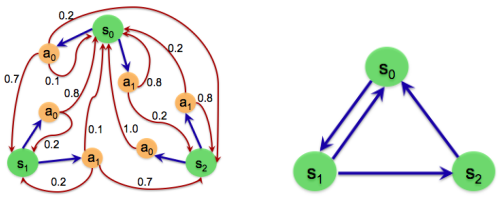
If we wish to plan a trajectory leading to some specific goal state s_g, the trajectory is called a *goal-oriented trajectory*, defined as below.

***Definition** 2.3:* A **goal-oriented trajectory** from a starting state s_0 is a finite sequence $s_0, a_0, s_1, a_1, \cdots, s_g$, where $s_g \in G$ is the specified goal at which the agent needs to arrive [16]. The value $V(s_i)$ of any state s_i under π is the total expected cost that incurs before the agent reaches the goal s_g. A trajectory can be in the *expected sense*.

The policy computed for a goal-oriented trajectory is termed a *goal-oriented policy*. We are especially interested in stationary policies that reach a goal state with probability 1 from any initial state.

Although the MDP stochastic system described above allows multiple goal states (absorbing states, Assumption 2.1) to simultaneously exist, a direct employment of VI or PI on the single-agent state/action space does not address the goal assignment problem, and actually each MAV selects its goal (and action policy) in a greedy manner. See Fig. 1 for an illustration.

A popular way to integrate the goal assignment and the MDP-based planning frameworks is to model each agent with an MDP and then employ the task allocation strategy to coordinate individual MDPs (e.g., see [7, 10]). However, this requires the state space and action space to be augmented to accurately include all possible outcomes (all combinations of single-agent uncertainties) that may occur in all future moments. In such case, the size of the state space and action space can be increased to as large as $|S|^n$ and $|A|^n$ for n



(a) (b)

Fig. 2. (a) Stochastic system of MDP, where s represents state and a denotes action; (b) The corresponding approximated deterministic graph.

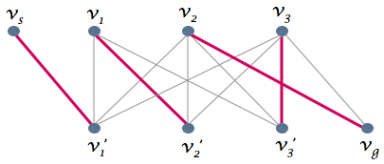


Fig. 3. Bipartite graph illustrating trajectory generation for one agent, where $V = \{v_1, v_2, v_3\}$, $V' = \{v_1', v_2', v_3'\}$, $V_s = \{v_s\}$, $V_g = \{v_g\}$. Matched edges are in red bold, others are unmatched edges; goal trajectory v_s—v_1—v_2—v_g in $G^d = (V, E)$ is obtained from the augmenting path v_s—v_1'—v_1—v_2'—v_2—v_g in the matching graph $\tilde{G} = (V, V', \tilde{E})$.

agents, which makes the problem intractable even for a small number of agents if the single-agent state/action space is already large.

III. Efficient Planning via Approximation and Distribution

We propose a method without expanding neither the action space nor the state space, such that the computational complexity is maintained at the level of single agent planning case. The time complexity is further improved by decoupling the problem and distributing the computational efforts. At a higher level, our proposed framework can be summarized with four main steps:

(A.) Abstract the problem and approximate a solution. Transform the stochastic system to a deterministic counterpart, which simplifies and abstracts the problem for an initial solution.

(B.) Decouple the problem. Find conflict-free deterministic paths for all MAVs in an extremely fast manner with existing deterministic solvers. Each agent obtains an initial path and a corresponding sub-problem.

(C.) Convert back to stochastic domain. Starting from states on the obtained initial path, each agent locally improves policies via running VI on states nearby the paths;

(D.) Improve global solution. Check the feasibility of final trajectories and adjust them hierarchically if necessary.

In greater detail, we present and discuss the four steps in the following subsections.

A. Stochastic Problem Abstraction and Approximation

In this step, the MDP stochastic transition model is approximated with a deterministic graph.

***Definition** 3.1:* The **approximated deterministic graph** is defined as $G^d = (V, E)$, where V corresponds to the vertex set of all states and E is the edge set. Assume an action a on state s transits to K possible states $S^K = \{s_k\}$ $(k = 1, \cdots, K)$ each of which has a transition probability $T_a(s, s_k)$ $(\sum_{k=1}^{K} T_a(s, s_k) = 1)$. The succeeding state with maximal probability is chosen as the expected next state:

$$s' = \max_{s_k \in S^K} T_a(s, s_k) \tag{6}$$

Then the corresponding deterministic edge $e = (s, s') \in E$ is added with an expected edge weight:

$$w(s, s') = \sum_{k=1}^{K} T_a(s, s_k) C_a(s, s_k). \tag{7}$$

In essence, each action of such approximation has a greedy impact—it transits to the state with the least immediate cost, thus, a trajectory planned on such graph can be regarded as the solution to a 1-step horizon planning problem (i.e., $\gamma = 0$). An example is shown in Fig. 2. In this way, the stochastic property is eliminated and the resulted deterministic graph abstracts the problem at a higher level and thus can be used to compute initial approximated solutions.

B. Problem Decoupling via Deterministic Goal-Oriented Trajectories

With approximated deterministic graph G^d, goal-oriented trajectories for all MAVs can be computed via existing deterministic planning methods, which are strongly polynomial suitable for online computations. Specifically, multi-agent goal constrained trajectories planning is a multi-source multi-goal (MSMG) shortest path problem. One efficient way to solve a MSMG is through transforming $G^d = (V, E)$ to a bipartite (matching) graph $\tilde{G} = (V, V', \tilde{E})$ [14]. The essence is recapitulated as follows.

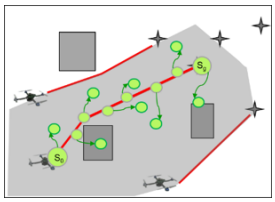
Briefly, a bipartite graph $\tilde{G}$ has two sets of nodes V and V', where V' is simply a copy of V such that $|V| = |V'|$, and an edge $\tilde{e} = (v_i, v_j') \in \tilde{E}$ connects the vertices $v_i \in V$ and $v_j' \in V'$ if there is an edge $e = (v_i, v_j) \in E \in G^d$. Edge $\tilde{e} = (v_i, v_j')$ is weighted the same as the counterpart edge (v_i, v_j)

$$\tilde{w}(v_i, v_j') = \tilde{w}(v_i', v_j) = w(v_i, v_j) \tag{8}$$

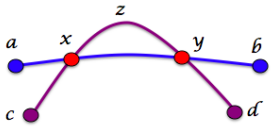
Besides that, a set of edges (v_i, v_i') is also added with weight $\tilde{w}(v_i, v_i') = 0$ for all states except the starts and goals.

A bipartite graph of this form well represents the assignment problem and can be solved by the Hungarian Algorithm [13], which manipulates *augmenting paths* consisting of *matched* and *unmatched* edges in order to find a solution where each vertex in V is uniquely matched (assigned) to a vertex in V' with the total cost minimized. Our MSMG trajectories are obtained by transforming the resultant augmenting paths back to the routing paths on G^d via eliminating all vertices except the goals from vertex set V' [14]. The concept is illustrated in Fig. 3.

It can be shown that the total length of all MSMG trajectories is globally shortest. This is because only the weights of matched edges are summed as feasible assignment costs. Formally, let $\mathbb{P}$ denote the set of obtained MSMG paths, since $\tilde{w}(v_i, v_i) = 0$ for all states except the starts and goals, the sum of weight $\mathcal{S}_a$ in the assignment matching



(a)



(b)

Fig. 4. (a) Local states fringing. Red lines are MSMG paths connecting to goals (stars). The green circles are fringe states and the shaded area represents local state space for the bottom-left MAV; (b) If path $a \to b$ is locally optimal, the refined path $c \to d$ will not cross path $a \to b$. Otherwise, it contradicts the local optimality assumption of path $a \to b$ and its segment $x \to y$.

result is

$$\mathcal{S}_a = \sum_{\forall (v_i,v_j)\in\mathbb{P}} \tilde{w}(v_i,v_j) + \sum_{\forall v_i \notin \mathbb{P}} \tilde{w}(v_i,v_i) = \sum_{\forall (v_i,v_j)\in\mathbb{P}} w(v_i,v_j), \tag{9}$$

which is exactly the total length of the MSMG routing paths. Therefore, the global optimal assignment solution corresponds to the globally shortest MSMG paths.

C. Policy Refinement of Stochastic Sub-problems

The problem is converted back to the stochastic domain in this step. With the obtained MSMG paths $\mathbb{P} = \{P_i\}$, each MAV $r_i \in \mathcal{R}$ grows a local copy of state set S^{P_i} initialized from only those states on the path $\mathcal{P}_i$, and incrementally connects and adds new neighboring states, called *fringe states* $\{s^{F_i}\}$, which can directly transit to S^{P_i}, as shown in Fig. 4(a).

***Remark** 3.1:* The fringe states stop growing when they encounter states on a different path. In other words, the nearby paths of neighboring MAVs form boundaries to the local state sub-space (boundary states included). Therefore, the global control policy only needs to be refined by MAVs within their partitioned sub-spaces.

The rationale for Remark 3.1 lies in that, assuming the nearby path is locally optimal, the path currently under refinement will not become a path that goes beyond the nearby path. This is because the *Principle of Optimality* [1] states that any path segment on the (locally) optimal path is also (locally) optimal. See Fig. 4(b) for an illustration of this principle. (Note, since the boundaries constituted from nearby trajectories may not be closed, a maximal number of fringing steps can be preset to terminate the fringe state growing process if such a condition occurs.)

Then, value iteration on S^{P_i} is carried out to refine policies on the local states. Retrieving goal-oriented paths based on updated policies yields refined trajectories, which in turn form updated boundaries for local sub-spaces. This procedure repeats until all trajectories and local policies become stable.

It is worth noting that a goal-oriented path here is actually a sequence of future actions, which form a "path" in an expected sense, as mentioned earlier. It does not mean that the MAV will absolutely follow such a path. Since every state maps to an action, even if the MAV deviates from the expected path, the (local) policy will guide the MAV to move towards its allocated goal state.

Algorithm 1: Proposed Approximation Method

1 Convert the MDP stochastic transition system into a deterministic graph G^d
2 On G_d, find mutually exclusive MSMG paths $\mathbb{P} = \{\mathcal{P}_i\}$ for all MAVs $r_i \in \mathcal{R}$
3 **foreach** *path* $\mathcal{P}_i \in \mathbb{P}$ **do**
4 Get the set of waypoint states $S^{P_i} \subset S$ on $\mathcal{P}_i$
5 $\forall s^{P_i} \in S^{P_i}$, propagate values backward
6 Iteratively grow fringe states s^{F_i} from S^{P_i} until states of another path are reached, $S^{P_i} \leftarrow S^{P_i} \cup \{s^{F_i}\}$
7 Execute Value Iterations on S^{P_i}
8 Retrieve improved trajectory $\mathcal{P}_i'$
9 **Stop** iteration if $\mathcal{P}_i'$ and policy converge
10 Check the feasibility of final policies. If mutual exclusiveness is violated between, e.g., paths $\mathcal{P}_k, \mathcal{P}_l \in \mathbb{P}$, improve them locally (**goto** Step 2).

D. Global Solution Improvement

The last step is to check the feasibility of the refined trajectories and improve the global solution by interacting between local solutions when necessary. The possible infeasibility comes from the violation of mutual exclusive goal assignments. This happens since nearby paths, including their goals, are included as fringe states during the refining process. Thus set S^{P_i} contains multiple goals and it is possible that the retrieved path leads to a new goal belonging to another path, causing conflicts on this new goal.

To address this problem, conflicted MAVs are required to combine their local state sets S^{P_i}, which consequently form a *local* multi-agent goal-constrained planning problem. Then, the conflicted MAVs need to solve this local problem following similar procedures described above until the conflicts are resolved, which addresses the global goal-constrained planning for the whole multi-MAV system.

Finally, Algorithm 1 summarizes the proposed method.

IV. Discussion and Analysis

Since this presented method is an approximated planning strategy that aims at coordinating multiple MAVs, therefore two related important properties need to be discussed: the solution compared to the ground truth, and the collision awareness (avoidance) from obstacles and other agents.

A. Construction of Optimal Solution for Comparison

For a multi-agent system with each agent subject to uncertainty, directly computing the optimal solution is very difficult. This is because each agent needs to consider the uncertainty of others, and every agent plays a role as a dynamic obstacle to other agents. Instead of formulating the "ground-truth" as a complex mathematical program, we decompose the problem into two sub-programs — a maximization sub-program (Eq. (10)) for computing goal-oriented

optimal policies and a minimization sub-program (Eq. (11)) for generating globally optimal trajectories that takes into account goal assignment. Then the two sub-programs can be computed in subsequent epochs along MAVs' expected state transitions.

Sub-program 1 :

$$\begin{aligned} \text{Maximize}^{\dagger} \quad & \sum_{s} V^{\pi_j}(s) \qquad (10) \\ \text{Subject to } & C_a(s,s) = 0, \text{ if } s = s_j \in \mathcal{G}, \\ & T_a(s,s) = 1, \text{ if } s = s_j \in \mathcal{G}, \\ & V^{\pi_j}(s) \leq \sum_{s} T_a(s,s')[C_a(s,s') + \gamma V^{\pi_j}(s)] \\ & \forall s \in S,\ a \in A. \end{aligned}$$

Sub-program 2 :

$$\begin{aligned} \text{Minimize} \quad & \sum_{i=1}^{|\mathcal{R}|} \sum_{j=1}^{|\mathcal{G}|} V_{ij}^{\pi_j}(s) x_{ij} \qquad (11) \\ \text{Subject to } & \sum_{j=1}^{|\mathcal{G}|} x_{ij} = 1, \quad i = 1, 2, \cdots, |\mathcal{R}| \\ & \sum_{i=1}^{|\mathcal{R}|} x_{ij} \leq 1, \quad j = 1, 2, \cdots, |\mathcal{G}| \\ & x_{ij} = \{0, 1\}. \end{aligned}$$

More formally, in *Sub-program 1*, V^{π_j} represents the MDP values of the goal-oriented policy for the j-th goal $s_j \in \mathcal{G}$, which can be computed via VI in practice. Note, s_j is set as an absorbing state whereas all other goals are regarded as normal states. Each agent acts as an obstacle to others so that collision avoidance among agents is taken into account. The value $V_{ij}^{\pi_j}(s)$ in *Sub-program 2* denotes the expected value between the i-th MAV and j-th goal when the MAV is situated at state s.

The main steps for computing the ground truth are pseudo-coded in Algorithm 2. Algorithm 2 needs to maintain a vector of length $|\mathcal{G}|$ for each state $s \in S$ to store goal-oriented MDP values for all goals, requiring to run VI $|\mathcal{G}|$ batches for every state. The $|\mathcal{R}|$ MAVs collectively construct a $|\mathcal{R}| \times |\mathcal{G}|$ matrix, which can be used for task allocation with time complexity of only $O(|\mathcal{R}||\mathcal{G}|^2)$ [13]. The task allocation result then determines a goal-constrained optimal action for each MAV, which drives the MAV to move to future states leading to different goals. Algorithm 2 is repeated after MAV enters a new state, with VI being re-executed every time to dynamically update obstacle information, as each MAV acts as a dynamic obstacle.

B. Collision Awareness

One natural question is if the proposed approach is also aware of the dynamic obstacles. (More accurately, the collision avoidance is in fact collision *awareness* in the stochastic planning context, as it is extremely difficult for the absolute collision avoidance with a guarantee due to the stochastic nature of actions.)

Algorithm 2: Ground-truth Multi-MAV MDP

foreach *goal* $s_j \in \mathcal{G}$ **do**
  Compute goal-oriented policy π_j and values $V^{\pi_j}(s)$ using Value Iteration, $\forall s \in S$
if *MAV* $r_i \in R$ *is situated at* s_i **then**
  Obtain goal-oriented policy values $V_{ij}^{\pi_j}$, $\forall s_j \in \mathcal{G}$
Construct assignment matrix $M_a = \left(V_{ij}^{\pi_j}\right)_{|\mathcal{R}| \times |\mathcal{G}|}$, $\forall r_i \in \mathcal{R}, s_j \in \mathcal{G}$
Compute optimal assignment $\phi : \mathcal{R} \to \mathcal{G}$
foreach *agent* $r_i \in \mathcal{R}$ **do**
  Retrieve optimal action policy based on assigned goal $s_{j'} = \phi(r_i)$:
  $\pi_i^*(s_i) = \arg\min_{a \in A} \sum_{\forall s_i' \in S} T_a(s_i, s_i')[C_a(s_i, s_i') + \gamma V_{ij'}^{\pi_{j'}}(s_i')]$.

Remark 4.1: The policies produced from our method always guide each MAV to the safest region while accomplishing its assigned goal state, avoiding not only static obstacles but also other agents.

For those static obstacles, the MDP model already takes into account of them and they are avoided. The property of dynamic obstacle awareness is due to individual agent's policy generation while solving the decoupled stochastic sub-problem (Sect. III-C), where each agent treats the expected trajectories of its nearby team members as boundaries/walls. In other words, action policies computed by different agents attempt to avoid the space that other agents are likely to visit, and thus avoid possible future collisions.

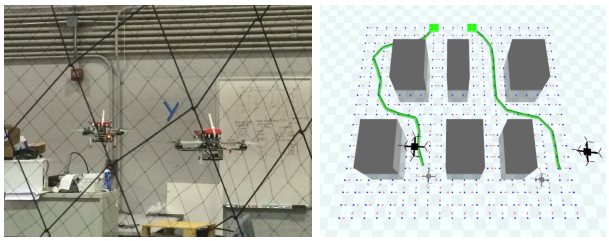
V. Experiments

We have validated the proposed method in simulation and demonstrated it on our physical air vehicles. The planner was run on a single core 1.60GHz Pentium processor with 2GB of memory, to emulate the onboard computing devices mounted on the MAV systems.

Specifically, following the classic scenarios in MDP literature, we tessellated the environment into grids so that each MAV is only allowed to move between adjacent grids. The transition dynamic model is depicted in Fig. 1(a). We set the transition probabilities as $\alpha = 0.8$ and $\beta_1 = \beta_2 = 0.1$ (in practice, transition probabilities can be obtained from offline disturbance testing), and fix the discounting factor as $\gamma = 0.9$. In addition, each action induces a cost of 1 and the ϵ value for VI is 0.1. Goal states are randomly generated with some arbitrary positive values (representing utilities), and obstacle states are preset with a negative value (representing penalty); the values of all other states are 0-initialized.

Figure 5(a) demonstrates a miniature multi-MAV system involving two micro air vehicles navigating towards des-

[†]Reason for objective maximization: Let T be the *Bellman Operator* such that the right hand side of Eq. (4) can be simplified as $(TV)(s) = \min_{a \in A} \sum_{\forall s' \in S} T_a(s,s')[C_a(s,s') + \gamma V(s')]$. Operator T has monotonicity property, i.e., $V_1 \leq V_2 \Rightarrow TV_1 \leq TV_2$. Eq. 4 can be expressed as an inequality: $V \leq TV$, which can be further chained as $V \leq TV \leq T(TV) \leq \cdots \leq T^n V = V^*$ as $n \to \infty$. The resultant inequality $V \leq V^*$ indicates that the solution V^* is optimal when the vector $V(s)$ is maximized.



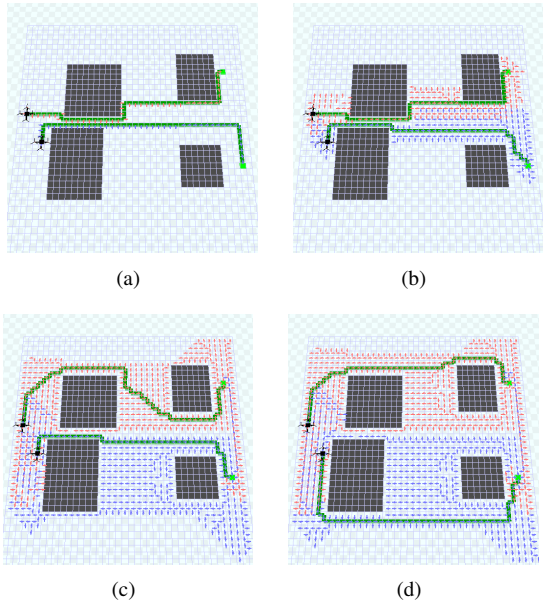
(a) (b)

Fig. 5. Experimental demonstration. (a) A miniature multi-MAV system with two micro air vehicles whose states are fully observed by a motion capture system; (b) Visualization of physical experiments where two MAVs navigate to two goals (green grids) among a group of obstacles (black blocks). Expected trajectories are planned across the safest regions.

ignated goal states and avoiding a set of obstacles. States of MAVs are captured by a motion capture system so that they are fully observable. Figure 5(b) is the visualization of corresponding physical experiments, from which we can observe that the trajectories are planned across the safest areas, indicating the awareness of uncertain action outcomes. Again, the trajectories here are planned in the expected sense. It does not mean that the MAVs will absolutely follow them. If an MAV deviates from its expected trajectory, the local action policy of its situated state will still lead the MAV towards its allocated goal state.

Figure 6 illustrates the process of policy refinements by two MAVs. We can see that as local spaces partitioned by deterministic MSMG paths are exploited, the policies (and expected paths) are gradually improved to eventually avoid obstacles and narrow spaces.

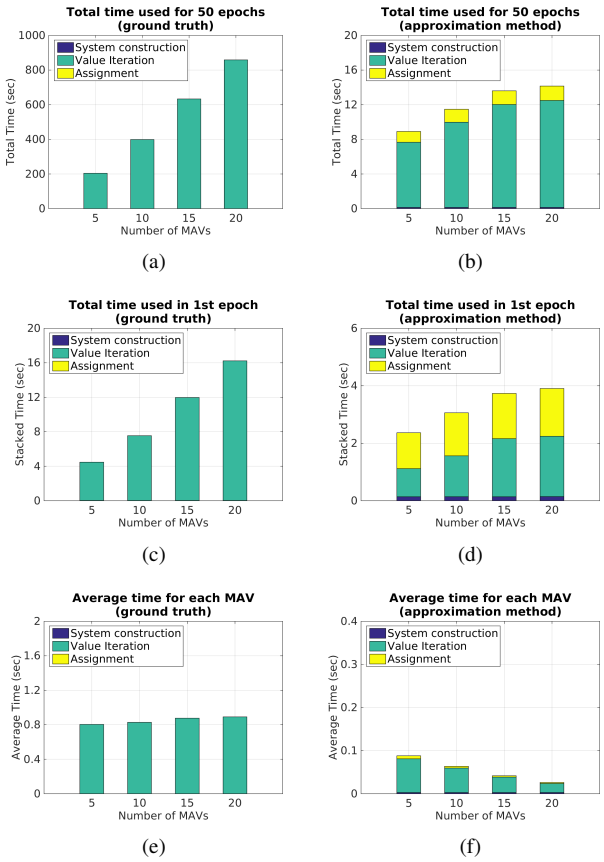
To evaluate the methods, we also conducted simulations with more MAVs and a greater number of states.



(a) (b)

(c) (d)

Fig. 6. Two MAVs refine policies locally. (a) Deterministic MSMG paths; (b)(c) As local spaces are exploited, the policies (and expected paths) are improved to avoid obstacles and narrow spaces; (c) Final refined policies and expected trajectories within the decomposed subspaces.



(a) (b)

(c) (d)

(e) (f)

Fig. 7. Time performance comparisons (data from fifty trials). (a)(b) Total time used for computing 50 epochs for the ground truth and the approximation method, respectively; (c)(d) The time used for computing the first epoch (the most expensive epoch); (e)(f) Average time for each MAV in the two methods.

A. *Run-time Comparison*

We first compared the running time between the ground truth and the proposed approximation method, by using the same number of states. Figures 7(a) and 7(b) show the total time used for computing 50 epochs (i.e., the goal-oriented trajectory of each MAV contains 50 future actions) from arbitrary initial states to an arbitrary set of goal states. The x axis represents the number of MAVs, which is manipulated from 5 to 20. The results clearly reveal that the approximation method requires significantly less overall computational effort and time.

The stacked bars in Figures 7(c)–7(d) detail the time allocation between algorithm components to compute the initial epoch — the most expensive stage consists of three parts: the time to construct the stochastic transition system, the time for initial VI, and the time for computing the goal assignment and retrieving goal-oriented trajectories. We observe that the time scale of the ingredients greatly differs between the two methods. For the ground truth method, the VI takes bulk of the time; in contrast, the assignment computation is an expensive component for the approximation approach. This is because the MSMG trajectories computed from the approximated deterministic graph require construction of an assignment matrix of size $(|S|+|R|)\times(|S|+|G|)$, which is

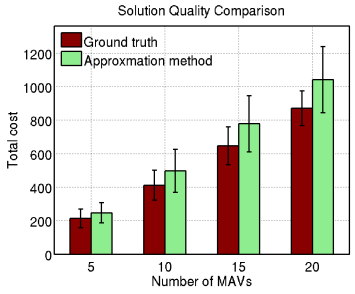


Fig. 8. Comparison of solution quality reveals limited sacrifice of optimality given the approximation method.

much larger than the assignment matrix in the ground truth method. Since each MAV only needs to process partial state space, the overall time used in the approximation method is still much less than that of the ground truth.

Figures 7(e) and 7(f) compare average time used by each MAV. Since we modelled both the proposed approximation method and the ground truth method in a decoupled framework, the computational workload can be carried out in a distributed manner for both methods (here we do not consider communication overhead). Figure 7(e) indicates that, the majority of time used by individual MAV is still VI, but the time is independent of the number of MAVs. In contrast, the time used for local VIs in the approximation method decreases as the number of MAVs grows. The histograms in Figures 7(e) and 7(f) reveal that the approximation method is at least eight times faster than the ground truth approach, and the more MAVs, the more significant speed benefit in our method. Such feature best applies to real-time multi-agent systems and online applications.

B. Solution Quality Comparison

We then compared the solution quality of the proposed approximation method against the ground truth optimal approach. The metrics here is the total cost for all multi-MAV trajectories. From Fig. 8 we can see that the cost for the approximation approach is slightly inferior to the ground truth, indicating a tradeoff for the time efficiency obtained from local policy approximations.

Therefore, without sacrificing much solution quality, the approximation method significantly decreases the overall computational workload, and greatly reduces the practical runtime.

VI. Conclusion

We propose an efficient goal constrained decision theoretic planning method for multi-MAV systems subject to action uncertainty. By abstracting the stochastic transition system, we are able to compute initial deterministic MSMG paths that are then used for developing a decoupled approximation approach. Our method requires that each agent execute only local refinements and adjustments with consideration of only a partial state space. The proposed approach yields performance benefits including reduced computational effort, leading to fast online operation, with limited impact on solution quality.

References

[1] R. E. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ., 1957.

[2] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein. The complexity of decentralized control of markov decision processes. *Math. Oper. Res.*, 27(4):819–840, Nov. 2002.

[3] C. Boutilier. Planning, learning and coordination in multiagent decision processes. In *Proceedings of the 6th Conference on Theoretical Aspects of Rationality and Knowledge*, pages 195–210, 1996.

[4] T. Campbell, L. Johnson, and J. P. How. Multiagent allocation of markov decision process tasks. In *Proc. of the Amer. Control Conf.* IEEE, 2013.

[5] J. Capitán, M. T. J. Spaan, L. Merino, and A. Ollero. Decentralized multi-robot cooperation with auctioned POMDPs. In *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, pages 3323–3328, 2012.

[6] J. S. Dibangoye, C. Amato, and A. Doniec. Scaling up decentralized MDPs through heuristic search. In *Proc. of the Conf. on Uncertainty in Artificial Intelligence*, August 2012.

[7] D. A. Dolgov and E. H. Durfee. Optimal resource allocation and policy formulation in loosely-coupled markov decision processes. In *ICAPS*, pages 315–324, 2004.

[8] C. V. Goldman and S. Zilberstein. Communication-based decomposition mechanisms for decentralized mdps. *CoRR*, abs/1111.0065, 2011.

[9] G. M. Hoffmann, S. L. Wasl, and C. J. Tomlin. Quadrotor helicopter trajectory tracking control. In *Proc. of the AIAA Guidance, Navigation, and Control Conf.*, 2008.

[10] H. Hosseini, J. Hoey, and R. Cohen. A coordinated MDP approach to multi-agent planning for resource allocation, with applications to healthcare. *CoRR*, abs/1407.1584, 2014.

[11] S. Karaman, M. Walter, A. Perez, E. Frazzoli, and S. Teller. Anytime motion planning using the RRT*. In *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, May 2011.

[12] L. Kavraki, P. Svestka, J. Latombe, and M. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. Technical report, Stanford, CA, USA, 1994.

[13] H. W. Kuhn. The Hungarian Method for the Assignment Problem. *Naval Research Logistic Quarterly 2:83–97*, 2:83–97, 1955.

[14] L. Liu and D. A. Shell. Physically Routing Robots in a Multi-robot Network: Flexibility through a Three Dimensional Matching Graph. *Intl. J. Robot. Research*, 32(12):1475–1494, 2013.

[15] L. Matignon, L. Jeanpierre, and A.-I. Mouaddib. Distributed value functions for the coordination of decentralized decision makers. In *Proc. of the Intl. Conf. on Auton. Agents and Multiagent Syst.*, pages 1209–1210, 2012.

[16] Mausam and A. Kolobov. *Planning with Markov Decision Processes: An AI Perspective*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2012.

[17] D. Mellinger, N. Michael, and V. Kumar. Trajectory generation and control for precise aggressive maneuvers with quadrotors. *Intl. J. Robot. Research*, 31(5):664–674, 2012.

[18] F. S. Melo and M. Veloso. Decentralized mdps with sparse interactions. *Artificial Intelligence*, 175(11):1757 – 1789, 2011.

[19] F. A. Oliehoek, M. T. J. Spaan, S. Whiteson, and N. Vlassis. Exploiting locality of interaction in factored Dec-POMDPs. In *Proc. of the Intl. Conf. on Auton. Agents and Multi Agent Syst.*, pages 517–524, 2008.

[20] P. Plamondon, B. Chaib-draa, and A. R. Benaskeur. A multiagent task associated mdp (mtamdp) approach to resource allocation. In *AAAI Spring Symposium: Distributed Plan and Schedule Management*, pages 89–96, 2006.

[21] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., 1st edition, 1994.

[22] M. Turpin, K. Mohta, N. Michael, and V. Kumar. Goal assignment and trajectory planning for large teams of aerial robots. In *Proc. of Robot.: Sci. and Syst.*, Berlin, Germany, June 2013.