

Robust Decentralized Task Assignment for Cooperative UAVs

Mehdi Alighanbari* and Jonathan P. How†
Aerospace Controls Laboratory
Massachusetts Institute of Technology

This paper investigates the problem of decentralized task assignment for a fleet of cooperative UAVs. It extends the analysis of a previously proposed algorithm to consider the performance with different communication network topologies. The results show that the second communication step introduced during the planning phase of the new algorithm is crucial for sparse networks because the convergence rate of the information consensus algorithms can be quite slow. Further analysis shows that the selection of the candidate plans communicated during this planning phase has a significant impact on the performance of the overall algorithm. A comparison of the performance and computation of four selection approaches clearly shows the importance of correctly accounting for the potential actions of the other UAVs, even though that tends to be more computationally expensive. A modification of the original candidate plan selection algorithm is also presented, which further improves the overall performance by increasing the robustness to inconsistencies in the information across the team.

I. Introduction

To ensure scalability and flexibility of high-level control systems, various distributed architectures have been developed for the task assignment process.¹⁻⁴ Within some distributed frameworks (*e.g.*, *implicit coordination*), each vehicle determines its own mission by simultaneously choosing tasks for all vehicles in the fleet using a centralized planning algorithm⁵ and then executing its own plan. To ensure consistency, information is shared to update the situational awareness (SA).⁶⁻⁸ Hierarchic approaches typically assume the formation of sub-teams that use locally dense communication networks to share information (states, measurements, and plans). Communication between sub-teams would be limited, although it is assumed to be available if necessary to exchange resources. These two approaches reduce the reliance on a central planner system, thereby increasing the rate that the planning system can react to pop-up threats and/or targets of opportunity, increasing the robustness to failure, and ensuring that the control system degrades gracefully. However, it is essential that these distributed control decisions be well coordinated to maintain good overall performance. A key problem is that achieving tight coordination typically requires that the vehicles exchange large quantities of information about the environment, their current states,

*Research Assistant, MIT Department of Aeronautics and Astronautics, mehdi_a@mit.edu

†Associate Professor, MIT Department of Aeronautics and Astronautics, jhow@mit.edu

and their future intentions. Communication on this scale will not always be possible and it also increases the visibility of the vehicles to threats.

Constraining the communication limits situational awareness, which raises two key issues: first, that decisions must be made based on incomplete information and second, that information may be inconsistent across the fleet, potentially leading to a less cooperative behavior. Thus one of the primary challenges is that these high-level algorithms must be modified to make them much less reliant on having “perfect, global” situational awareness while still obtaining reasonable performance. The authors investigated this issue in Ref. [9] and presented new insights on the issue of identifying the relative significance of various types of information that could be communicated. That paper also presented a new approach (called Robust Decentralized Task Assignment (RDTA)) to avoid conflicts in the decentralized planning wherein the UAVs communicate a set of candidate plans and plan independently using that information. This approach ensures that the resulting plans computed by the team are non-conflicting and that the plan chosen to be executed by each team member is feasible for that vehicle.

While Ref. [9] demonstrated the benefit of adding the second communication phase, it also showed that the set of candidate plans chosen to be communicated can have a significant effect on the overall performance. The second stage of planning is based on this set of candidate plans and, because this is consistent between all UAVs, the final plan is guaranteed to have no conflicts. However, the quality of the final plan strongly depends on these candidate plans. For example, if the candidate plan set includes a good/optimal feasible plan where all the UAVs are assigned to targets without any conflicts, that will lead to high performance. But if this set lacks a good feasible plan, then the final plan will not contain conflicts, but its performance could be quite poor, since in these cases, some UAVs might not be assigned to any targets. This paper investigates several different methods for creating these candidate sets to improve the performance of the overall algorithm and compares them using extensive simulations.

To further improve the average performance of the algorithm, two additional modifications are considered to increase robustness of the assignment process. In particular, to reduce the potential of conflict in the set of candidate plans (typically leading to suboptimal performance when non-conflicting plans are chosen), the communicated candidate plan sets must include a very diverse set of waypoints and targets. This concept is implemented in the RDTA algorithm by penalizing the similarity in the candidate set. Simulation results are presented to show that this improves the average performance of the algorithm. Finally, this paper also extends the results in Ref. [9] to consider the algorithm performance with different network topologies, with a particular focus on how best to allocate the communication resources between the different parts of the algorithm (information consensus and planning consensus).

II. Robust Decentralized Task Assignment

To develop the algorithms in this paper several assumptions are made. First, the set of waypoints (tasks) have been identified for the team of UAVs. The location of the waypoints are represented by a $N_w \times 2$ matrix B . Waypoint scores (values) are represented by the vector $S = [s_1 \dots s_{N_w}]$.

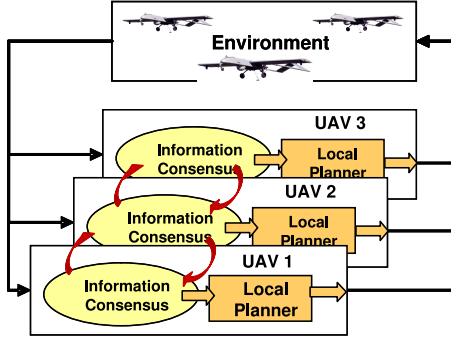


Figure 1. Implicit coordination approach using information consensus following by independent planning.

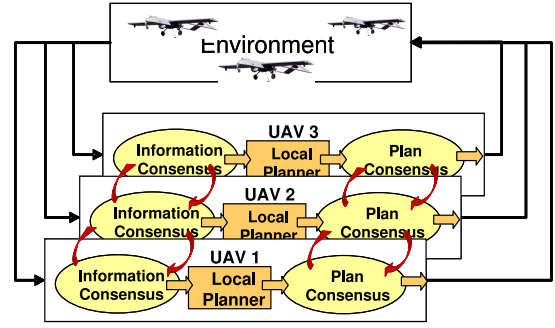


Figure 2. Robust Decentralized Task Assignment algorithm which adds an additional round of communication during the plan consensus phase.

Each team is made up of N_v UAVs with known starting points, speed, and capability (*i.e.*, strike, reconnaissance, etc.).

The UAV capabilities are represented by the $N_v \times N_w$ binary matrix K ; $K_{vw} = 1$ represents a UAV v capable of performing the task associated with waypoint w . It is also assumed that there are polygonal “No Fly Zones” in the environment.

Given this information, the problem is to assign the UAVs to the waypoints to optimally fulfill a specific objective. There are several possibilities for specifying this objective, depending on the problem type. This paper uses the time-discounted score formulation, which captures the notion of target score and is more realistic for real world problems.

A. Algorithm Overview

Robust decentralized Task Assignment (RDTA) algorithm is designed to overcome the shortcomings of existing decentralized TA algorithms such as implicit coordination. It is assumed that the UAVs have different *situational awareness* (SA), meaning that their information is both noisy and inconsistent. The RDTA algorithm⁹ has two major phases: (1) Information Consensus and (2) Planning.

In phase 1 of the algorithm, the UAVs communicate with each other to improve the consistency of the information. A simple consensus algorithm is used in this phase of the algorithm to reach the required degree of consistency. If $I_i(t)$ is the information of UAV _{i} at time t then the linear discrete form of consensus filter can be written as:

$$I_i(t+1) = I_i(t) + \sum_{j=1}^{N_v} \sigma_{ij} G_{ij}(t) (I_j(t) - I_i(t)) \quad (1)$$

where $G(t)$ represents the communication network and G_{ij} is 1 if there is a direct communication link from UAV _{i} to UAV _{j} and zero otherwise. σ_{ij} are positive constants that represent the relative confidence of UAV _{i} to UAV _{j} about their information.

Reaching consensus, however, is not always possible due to communication limits, noise in the communications, and possibly slow rates of convergence in a dynamic environment. Thus it is likely

that the second (planning) phase will have to be executed with a limited degree of consistency in the SA. This phase is when the UAVs calculate and implement their task plans. In the implicit coordination approach this is done by replicating the centralized assignment algorithm on each UAV. Given full consensus on the SA, and using exactly the same algorithm on each UAV, this method would create similar and non-conflicting plans for each vehicle. However, any differences in the SA (for example, if the consensus algorithm has not yet converged), could lead to a plan with conflicts. RDTA has a modified second phase that eliminates these possible conflicts. Phase 2 of RDTA has two stages:

Stage One: The UAVs use their updated information to generate a set of ρ candidate plans. The petal algorithm^{2,9} is a good choice for the proposed algorithm because it performs an optimization based on pre-generated feasible plans and is modified to be used in this stage.

The petal algorithm can be explained as follows. First, each UAV_{*i*} creates a list of all unordered feasible task combinations for every UAV_{*j*}, given its capabilities ($P_{ij} = \{p_{ij}^1, \dots, p_{ij}^{k_j}\}$ and $P_i = \bigcup_{j=1}^{N_v} P_{ij}$). Next, the length of the shortest path made up of straight line segments between the waypoints and around obstacles is calculated for all possible order-of-arrival permutations of each combination (these permutations are referred to as petals). The construction of these paths can be performed extremely rapidly using graph search techniques.² The time of visit for each waypoint w , t_w is estimated by dividing the length of the shortest path to that waypoint by the UAV's maximum speed. The time-discounted score is consequently calculated for each waypoint in each petal ($S_w(t) = \lambda^t S_w$, $\lambda < 1$ is the discount coefficient). The time-discounted score for each petal is the sum of the discounted score of its waypoints.

The algorithm produces a set of N_M petals and a vector of size N_M whose p^{th} elements, taken together, fully describe one permutation of waypoints for one UAV. These are set P_i , whose P_i^{pw} entry is 1 if waypoint w is visited by petal p and 0 if not and vector \mathbf{S}_i , whose element S_i^p is the time-discounted score of the petal p . This procedure is described in detail in Ref. [2].

Once these sets are created, a mathematical method is developed for allocating the waypoints to each UAV based on these scores and other constraints. The base of the task allocation problem is formulated as a Multidimensional Multiple-Choice Knapsack Problem (MMKP).¹⁰ The “knapsack” in this case is the complete mission plan. The vector corresponding to each of the N_M petals makes up the multi-dimensional weight. The “multiple-choice” comes from choosing which petal to assign to each of the N_v different UAVs (sets). The objective is to assign one petal (element) to each vehicle (set) that is combined into the mission plan (knapsack), such that the score of the mission (knapsack) is maximized and the waypoints visited (weight) meet the constraints for each of the N_w dimensions. The problem is

$$\begin{aligned}
\max_x J_i &= \sum_{p \in \mathcal{M}_i} S_i^p x_i^p \\
\text{subject to} \quad \forall w \in \mathcal{W} : & \sum_{p \in \mathcal{M}} P_i^{wp} x_i^p \leq 1 \\
\forall v \in \mathcal{V} : & \sum_{p \in \mathcal{M}_{iv}} x_i^p = 1
\end{aligned} \tag{2}$$

where $\mathcal{M}_i = \{1, \dots, N_M\}$ and $\mathcal{M}_{vi} \subseteq \mathcal{M}$ are the indexes of petals in P_i and P_{ij} respectively and $\mathcal{W} = \{1, \dots, N_w\}$ is the list of waypoints. The binary decision variable x_i^p equals 1 if petal p is selected, and 0 otherwise. The objective in this problem formulation maximizes the sum of the scores to perform each selected petal. The first constraint enforces that each waypoint w is visited at most once. The second constraint prevents more than one petal being assigned to each vehicle. Solving this MMKP selects a petal for each UAV $_j$, $(p_{ij}^{k_1^*})$, which is essentially a solution to the task allocation problem. Note that this process is repeated in each UAV $_i$ and therefore each UAV $_i$ generates plans for every single UAV in the team.

The selected petal for UAV $_i$ will be the first candidate plan for UAV $_i$ and is added to the set, P_i^* , $P_i^* = \{p_{ii}^{k_1^*}\}$. After updating the set of feasible petals for each UAV,

$$P_{ij} = \begin{cases} P_{ij} & \text{if } (i \neq j) \\ P_{ij} - \{p_{ij}^{k_1^*}\} & \text{if } (i = j) \end{cases} \quad (3)$$

The optimization in 2 and update in 3 are repeated ρ times to create a list of candidate plans for UAV $_i$, $P_i^* = \{p_{ii}^{k_1^*}, \dots, p_{ii}^{k_\rho^*}\}$ and the scores associated with them, $S_i^* = \{S_i^{k_1^*}, \dots, S_i^{k_\rho^*}\}$. Each UAV $_i$ then communicates this set of ρ candidate plans, P_i^* and the scores associated with these petals to all other UAVs.

Stage Two: Each UAV $_i$ uses the set of candidate plans from all the UAVs, P_j^* 's and implements the petal p_{ii}^* that results from the optimization

$$\max_{x_{ij}^k} J_i = \sum_{j=1}^{N_v} \sum_{k \in \mathcal{M}_j^*} x_{ij}^k \cdot S_j^k \quad (4)$$

$$\text{subject to} \quad \sum_{k=1}^{N_{ij}} x_{ij}^k = 1; \forall j \quad (5)$$

$$\sum_{j=1}^{N_v} \sum_{k \in \mathcal{M}_j^*} x_{ij}^k p_j^k \leq \mathbf{1} \quad (6)$$

where $\mathcal{M}_j^* = \{k_1^*, \dots, k_\rho^*\}$. Algorithm 1 presents the summary of the Robust Decentralize Task Assignment algorithm. Note that this algorithm with $\rho = 1$ is essentially the implicit coordination algorithm.

III. Analyzing The Robust Decentralized Task Assignment

This section analyzes several important aspects of the RDTA algorithm and presents simulation results to highlight the effect of various design parameters on the performance.

A. Simulation setup

A simple scenario of 5 UAVs and 10 targets is used as the baseline for all simulations presented in the following sections. The simulations results are typically generated from 100 Monte Carlo

Algorithm 1 – RDTA Algorithm

```
for i = 1 : Number of UAVs ( $N_v$ ) do
  while (Not reached the desired degree of Consensus) do
    Send the current SA to neighbors,
    Receive other UAVs' SA,
    Update the SA using the consensus algorithm,
  end while
  Set  $P_i = \{\}$ ,
  for j = 1 : Number of UAVs ( $N_v$ ) do
    Enumerate all feasible plans for UAV $_j$ ,  $P_{ij}$ ,
    Calculate the time-discounted score of each plan,  $S_{ij}$ ,
     $P_i = P_i \cup P_{ij}$ ,
  end for
  Set  $P_i^* = \{\}$ ,
  for k = 1 :  $\rho$  do
    Solve the optimization in Eq. 2,
    Find the optimal plan for all the UAVs,  $p_{ij}^*$ ,
    Add  $p_{ii}^*$  to the list of candidate plans,  $P_i^* = P_i^* \cup \{p_{ii}^*\}$ ,
     $P_i = P_i - \{p_{ii}^*\}$ ,
  end for
  Send the set of candidate plans,  $P_i^*$  to all other UAVs,
  Receive other UAVs' candidate plans,
  Solve the optimization in Eq. 6,
  Find the optimal plan for all the UAVs,
  Implement the plan for UAV $_i$ ,  $p_i^*$ .
end for
```

simulations, with the position of the targets, (x, y) created randomly (these random numbers are uniformly distributed in $[200 \ 400] \times [200 \ 400]$). When comparing different algorithms, the exact same problems are solved in all cases using the same seed for the random number generators. To make the information inconsistent for different UAVs, a random number is added to the position of each waypoint. These random numbers are different for each target and each UAV. These random numbers are generated with a uniform distribution in the interval of $[-30\%, +30\%]$ of the value of the position. Each UAV then has its own version of the information, which is inconsistent with other UAVs. If $I_i = [I_i(1), \dots, I_i(m)]^T$ is the information vector (here position of waypoints) associated with UAV $_i$, then the level of inconsistency in the information is quantified using the inverse Sharpe ratio, ζ defined as,

$$\zeta = \sum_{k=1}^m \frac{\delta_I(k)}{\mu_I(k)} \quad (7)$$

where $\mu_I = [\mu_I(1), \dots, \mu_I(m)]^T$ and $\delta_I = [\delta_I(1), \dots, \delta_I(m)]^T$ are the mean and standard deviation of information vectors respectively. Communication between UAVs with a consensus algorithm, decreases the inconsistency in the information and consequently decreases ζ . Note that communication noise, new measurements and all the exogenous disturbances tend to increase ζ .

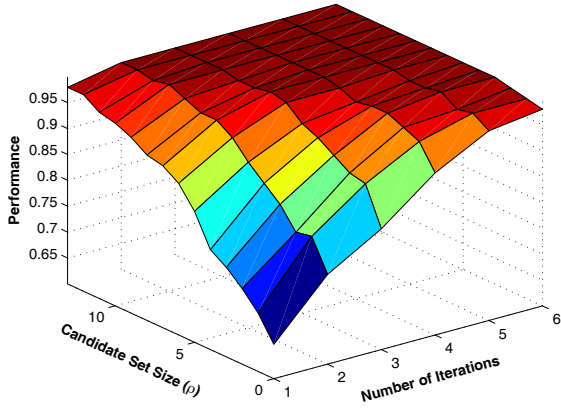


Figure 3. Effect of two important parameters (iterations for consensus and size of candidate petal set) on the performance.

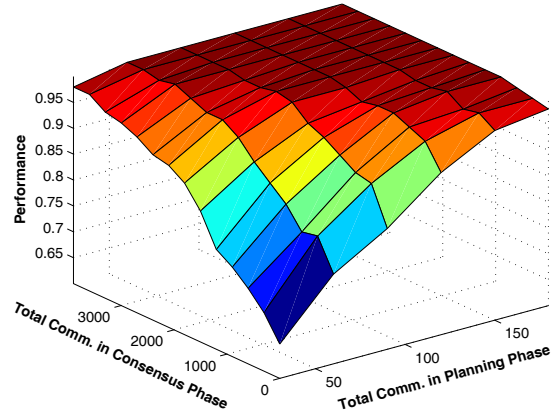


Figure 4. Effect on performance of communication in the two phases of algorithm (consensus and loop closure in the planning).

Note that a large ζ corresponds to a significant inconsistency in SA between the UAVs, which will impact the plans selected by each UAV. In the case of implicit coordination, where there is no feedback in the planning phase, this could result in conflicting plans for the team.

For most simulations, the petals are of size 2, which means that the UAV plans have less than or equal to two waypoints. In some cases this number is increased to 3. In a real-time implementation, these plans would be redesigned over time to ensure that all targets/tasks are completed. The communication networks for all simulations are generated randomly. These networks are all strongly connected networks which means there is a communication path from every UAV to any other UAV.

B. Effect of communication in the two phases of algorithm

The first analysis considers the effect of communication during the two phases of the algorithm on the performance. In the first phase, the consensus algorithm is used to improve the consistency of the information across the UAV team. This information is then used to produce a robust feasible plan for the fleet. To see the effect of communication in each phase, two parameters are varied in the simulations. In the first phase (consensus), the convergence of the information is directly related to the number of iterations. The amount of communication is also a linear function of the number of iterations. Therefore by changing the number of iterations, the amount of information communicated in the first stage can be directly controlled.

The second important parameter is the size of the candidate plan set that is communicated in stage 1 of the planning phase, ρ . The communication in this phase is also a linear function of this parameter and thus can be controlled as well. Figure 3 shows the result of the simulations introduced in Section A. The result shown is the average of 100 Monte Carlo simulations in which the positions of the targets were chosen randomly. Also the communication networks for these simulations are randomly-generated, strongly connected networks. For each scenario the number of iterations in consensus phase was changed from 0 to 7 and for each case RDTA was run with

different values of $\rho = 1, \dots, 5$. In this figure, the x -axis shows the number of iterations, the y -axis is the size of the candidate petal set (communicated petal set), ρ , and the z -axis is the performance of the algorithm (total score of the plan). The plot clearly shows that the performance of the algorithm increases as both parameters (number of iterations and the size of candidate petal set) increase.

To better show the relationship between performance and the communication in each phase, Figure 3 is transformed into Figure 4 in which the x - and y -axes are the communication in the information (consensus) and planning phases, respectively. Communication was measured using the following rules. In the information phase, in each iteration, each UAV has to communicate its information about the position of all targets to other UAVs. Assuming that the position of each target has two dimensions, then two words of information must be communicated for each target. There are 10 targets in this example and therefore the total communication (the number of words that all the UAVs communicate in all iterations) is

$$C_c = 20 \cdot r \cdot l \quad (8)$$

where $20 = 2 \times 10$ is the number of words that each UAV communicates in each iteration, r is the number of iterations in the consensus phase and l is the total number of links in the network.

In the planning phase, the petals and the score of each petal must be communicated. Each petal is a binary vector which can be interpreted to an integer number and transmitted as a word. Hence for each petal, each UAV must communicate a total of two words and therefore, in the planning phase, the total communication is:

$$C_p = 2 \cdot \rho \cdot l \cdot q \quad (9)$$

where the 2 is the number of words per candidate plan that UAV_{*i*} has to communicate, ρ is the size of the candidate plan sets, l is the number of links in the network, and q is the number of iterations required for all the UAVs to receive the candidate plan from any other UAV. Note that when there is no direct communication between two UAVs, these two UAVs communicate their candidate plan sets through other UAVs which act as relays. This is always possible since the assumption is that the networks are strongly connected. Therefore if the shortest path from UAV_{*i*} to UAV_{*j*} consists of u UAVs, then the total iterations needed for these UAVs to send their plans to each other will be $u + 1$ which is represented by parameter q in Eq. 9.

Figure 4 shows that increasing the communication in either axis (consensus or planning phase) improves the performance. However, the results also clearly show that communication in the planning phase is more efficient than in the information (consensus) phase in the sense that 200 words of communication in the planning phase has approximately the same effect on performance as 2000 words in consensus phase. The plot also shows that to maximize the performance, some communication in both phases is needed.

The performance curve versus number of iterations is shown for four different values of ρ (size of candidate plan set) in Figure 5. This can be thought of as four slices of Figure 3. In this figure $\rho = 1$ corresponds to the implicit coordination algorithm. The horizontal dashed line corresponds

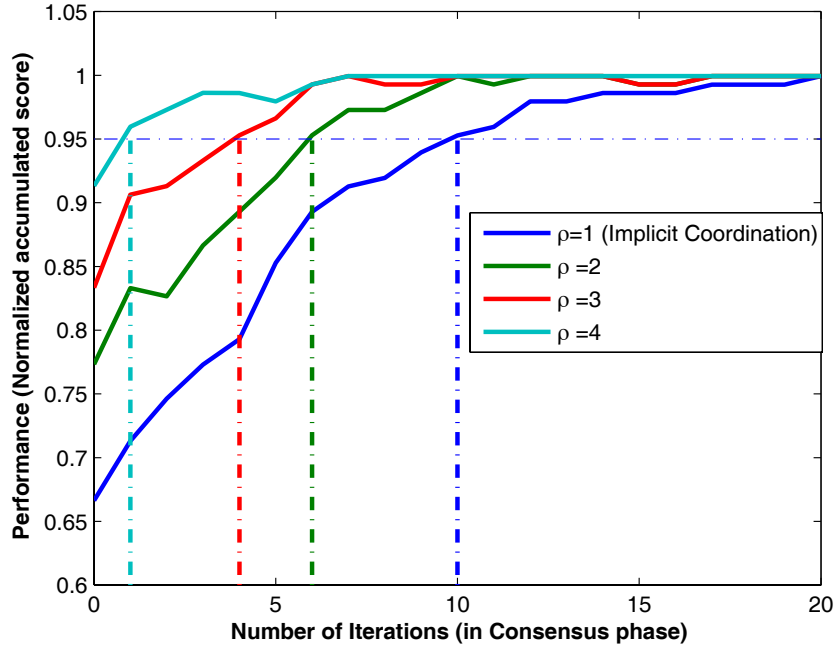


Figure 5. Performance versus the number of iterations

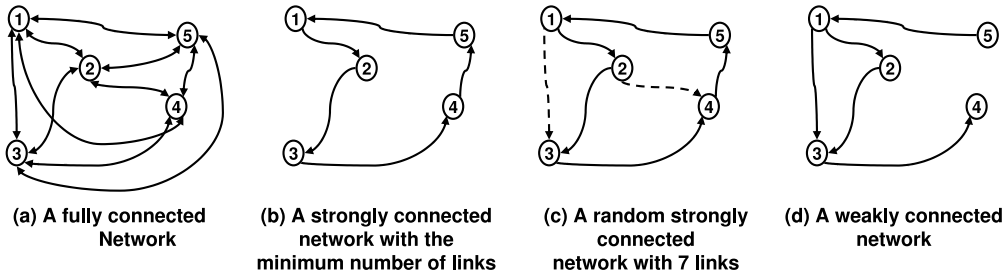


Figure 6. Creation of randomly generated strongly connected network topologies.

to 95% of the optimal performance, and the vertical dashed lines show the number of iterations needed for each case (different ρ) to reach the 95% limit. Note that for the implicit coordination case ($\rho = 1$), on average 10 iterations in consensus phase are needed to reach this performance limit. The required number of iteration decreases to 6, 4 and 1 by increasing ρ to 2, 3 and 4 respectively. Note that with only one iteration in the consensus phase and using RDTA with $\rho = 4$, the 95% limit can be reached. This figure clearly show that the *sufficient consistency level* is significantly lower for RDTA compared to the implicit coordination algorithm.

C. Effect of communication structure on the performance

The second analysis extends the previous results to communication architectures with different network topologies. The effect of the sparsity of the communication network on the RDTA performance is analyzed using the same simulation setup. In all the simulations considered it is assumed that the communication network is strongly connected, which means there is a directed path from

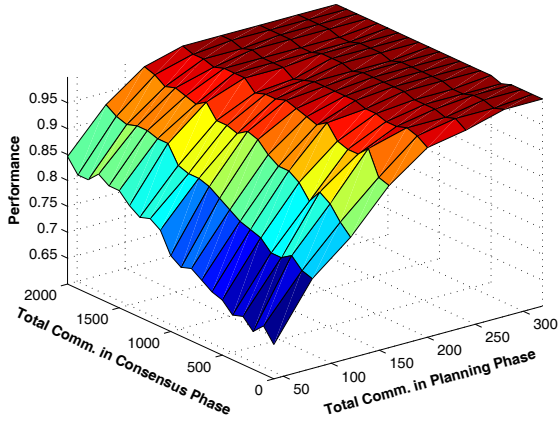


Figure 7. Strongly connected communication network with 5 unidirectional links.

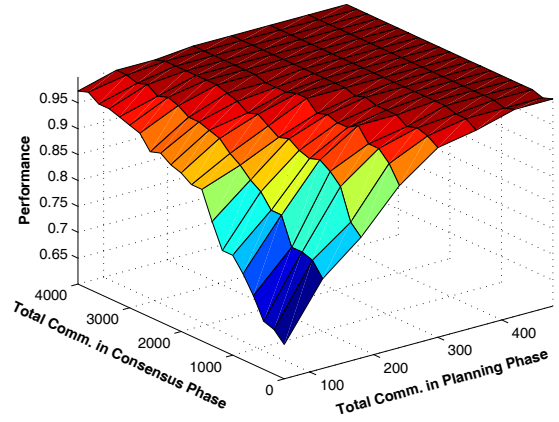


Figure 8. Strongly connected communication network with 10 unidirectional links.

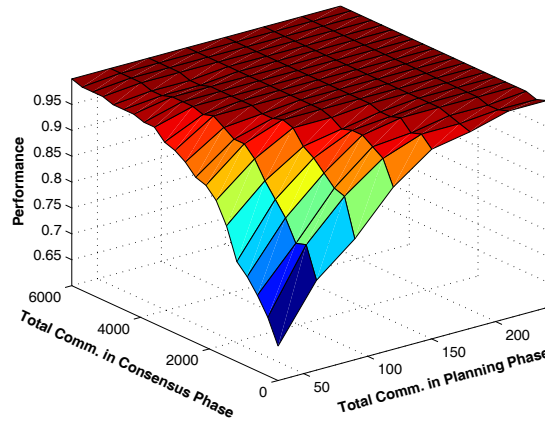


Figure 9. Strongly connected communication network with 15 unidirectional links.

Comparison of effect on performance of communication in the two phases of RDTA for different network topologies.

every UAV to every other UAV. So in creating scenarios for the simulations, a strongly connected graph with the minimum number of links is created and then additional links were added randomly to generate communication networks with 5, 10 and 15 links. Figure 6 presents different network topologies for a 5 node (UAVs) network. Fig. 6(a) shows a fully connected network in which each UAV directly communicates with every other UAV in the team. Fig. 6(b) shows a strongly connected network with minimum number of links (here 5). Fig. 6(c) shows a randomly generated strongly connected network with 7 links and Fig. 6(d) shows a network that is not strongly connected (weakly connected).

The results of these simulations are presented in Figures 7–9. The total communication in each phase (x - and y -axes) are calculated using Eqs. 8, 9. Comparing the performance for these three network topologies, it is clear that much more communication is needed for a sparse network (Fig. 7) to reach the same level of performance. It is also shown that, to reach a certain level

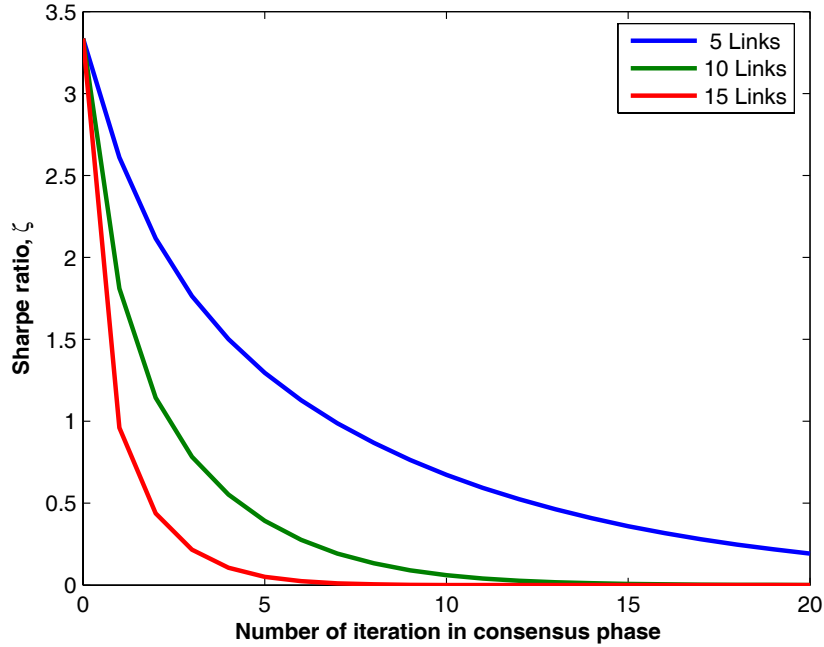


Figure 10. Comparing convergence rate of information in the consensus phase for 3 different network topologies.

of performance, communication in the planning phase is more effective than communication in the information consensus phase, thereby making the advantages of RDTA (and in particular communication in the planning phase) very apparent for sparse communication networks. In the cases of limited communication bandwidth and/or constraints in communication time, for sparse networks, the inefficiency of communicating raw information significantly degrades the performance and communication in planning phase (processed data: candidate plan set) is essential.

To get a better sense of the effect of communication on improving consistency, Figure 10 presents the inverse Sharpe ratio, ζ defined in (7) for these three communication topologies. The x -axis is the number of iterations in the consensus phase, or the number of times that the UAVs communicate with each other, and the y -axis is the inverse Sharpe ratio (a measure of inconsistency). The graphs clearly show that for sparse networks, the convergence rate for the information consensus decreases, and thus, with the same number of iterations (time spent in communication), the UAVs in a sparse network will have more inconsistencies in the information. This increased inconsistency in the data directly leads to more conflicts in the decentralized assignment, thereby decreasing the team’s performance. In this case, adding the communication during the planning phase has a large impact on the performance by significantly reducing the effect of the residual inconsistencies in the information.

D. Effect of the algorithm used in the first stage of planning

The third analysis addresses the issue of the impact of the algorithm used in the first stage of the planning. Section A discusses an approach which applies a centralized scheme to create a set of “*optimal coordinated plans*” for the set of candidate plans. However, this level of sophistication

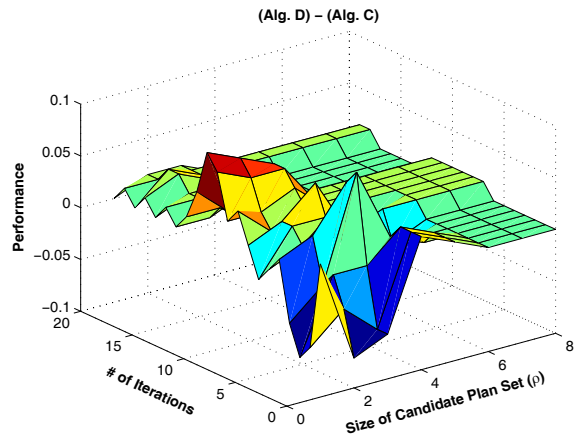


Figure 11. Compare algorithms D and C to show effect of greediness in time. z -axis is difference in performance $D - C$

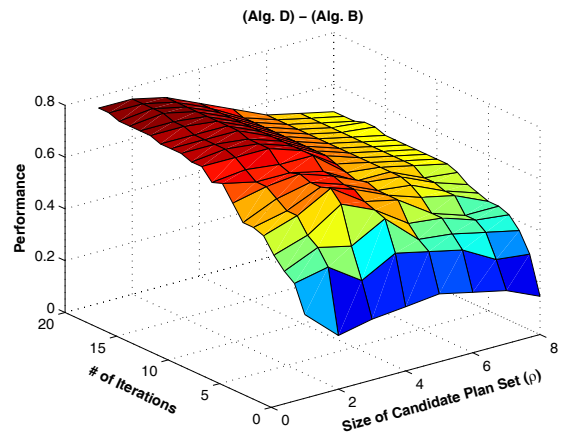


Figure 12. Compare algorithms D and B to show effect of coordination. z -axis is difference in performance $D - B$

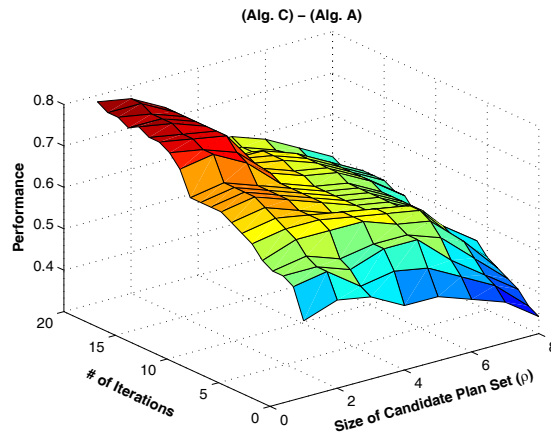


Figure 13. Compare algorithms C and A to show effect of coordination. z -axis is difference in performance $C - A$

(and its associated computational effort) may not be necessary to achieve good performance. The following investigates four alternatives for creating the candidate petals by varying the extent to which the plans account for future time and other vehicles in the team.

- A) **Greedy in space and in time:** This method ignores all other UAVs in its task selection (uncoordinated). It is also greedy in time since it only assigns the tasks one at a time.
- B) **Greedy in space (uncoordinated petal):** This method is similar to the petal algorithm, since it bundles the tasks and plans using these petals. However, this algorithm only generates petals for the planning UAV; picking the petals with the highest score without any regard to the other UAVs (uncoordinated).
- C) **Greedy in time, non-greedy in space (coordinated):** In this algorithm tasks are selected one at a time and added to the path (greedy in time), but the choices of all UAVs

Table 1. Comparing the Results of Different Algorithms

Algorithm	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
Avg. Performance (normalized)	0.36	0.42	0.99	1.0
Avg. Computation Time (normalized)	0.20	0.22	0.93	1.0

are considered in selecting these tasks. This means that each UAV creates its petal with consideration of what tasks the other UAVs might be doing (coordination).

D) **Petal (coordinated)**: This algorithm is the one given in Section A from the original RDTA algorithm.

The simulation from Section A is used to compare these four algorithms with the goal of determining the trade-off in performance degradation versus computational improvement. Similar to previous setup, the inconsistency is in the position of targets. Figures 11–13 compare the results of the four algorithms. Note that in all these graphs the performance values are normalized to be between 0 and 1 where performance of 0 means the total accumulated score by UAVs is zero and the performance of 1 is the performance of the optimal solution. To compare the effect of greediness in time, the algorithms *D* (petal) and *C* (greedy in time, non-greedy in space) are compared in figure 11. The *z*-axis in this figure is the difference in the performance between algorithms *D* and *C* ($D - C$). The plot clearly shows that the performance of these two algorithms are very close, and that greediness in time does not appear to have a significant effect on the overall performance.

In Figure 12, the petal algorithm (*D*) is compared to the greedy-in-space algorithm (*B*), ($D - B$). The difference in the performance is always positive, which clearly shows the advantage of algorithm *D* over *B* and highlights the disadvantage of poor team coordination. By accounting for the preferences of the other UAVs in selecting the candidate petals, each UAV can increase the probability of finding a non-conflicting plan with good performance in the second stage of planning phase. These results show that if the coordination is ignored in the first stage of planning, it cannot be recovered in the second stage, and although the final plan is non-conflicting, it might not assign some UAVs to any targets to avoid conflicts. Figure 13 compares algorithms *C* and *A*, ($C - A$) and confirms that coordination with teammates is a bigger driver of performance than planning further into the future.

Table 1 summarizes the performance and computation time of these four algorithms. The numbers here are normalized by the maximum. The results clearly show the advantage of having coordination in the first stage of planning (Algorithms *C* and *D*). The performances of these two algorithms are significantly better than the uncoordinated algorithms, but at the same time, the computation time for these two algorithms (*C* and *D*) are significantly higher than Algorithms *A* and *B*. The decision on which algorithm to use will be problem specific. In the cases where longer computation time of the coordinated algorithms is acceptable, they would be the better choice. But in a case where the computation time is constrained, algorithms *A* and *B* would have to be investigated.

IV. Performance Improvements

This section presents two extensions to the original RDTA algorithm that are introduced to improve the performance of the algorithm. These modifications are in the petal selection in the first stage of planning phase. In stage 2 of the planning phase, the UAVs create the final non-conflicting plan based on the candidate plan set. The performance of the final plan strongly depends on the quality of these candidate plans. For instance, if the algorithm cannot create a non-conflicting plan in which all UAVs are assigned to targets, it will avoid conflicts by not assigning one or more of the UAVs to any targets. Although this plan is the optimal plan given the candidate plan set, the performance might be quite poor because it is under utilizing the UAV team.

Define the size of the candidate petal set that each UAV communicates to the other UAVs as ρ . In the original RDTA algorithm, these ρ petals are the best petals for that specific UAV and therefore can be very similar in the sense that they have common targets/waypoints. In the case of inconsistent information between the UAVs, this similarity in the petals, reduces the possibility of finding a feasible assignment where all UAVs can be assigned to their maximum number of targets, and therefore has the potential of decreasing the expected performance of the team. Thus the objective is to find an alternative method to pick these petals in the first stage of planning in order to increase the possibility of having a feasible plan (plan with no conflicts) in the second stage. In particular to reduce the potential of conflicts in the set of candidate plans, the communicated candidate plan sets must include a very diverse set of targets. Two approaches are presented and compared in the following. The first modification (A) is intuitively appealing, but is shown to actually decrease the performance. Modification B improves the performance over the nominal algorithm.

1. Modification A

In this modification, when a petal, p_{ii}^{k*} is selected and added to petal list P_i^* for UAV_{*i*}, then all of the remaining petals in P_i that have common waypoints with the selected petal get penalized. Penalizing the score of petals that are similar to the petals that are already added to the candidate plan set (have common targets) improves the chance that the petals with lower score that are not similar to the selected petals will be selected. This algorithm will avoid creating a candidate plan set which includes a set of petals with high scores, but very similar set of targets. For example, suppose that p_{ii}^{k*} contains waypoint w , then any petal in the P_i that also contains w gets penalized,

$$\text{If } (w \in p_{ii}^{k*} \ \& \ w \in p_{ii}^k) \Rightarrow s_{ii}^k = s_{ii}^k - \delta, \ \forall p_{ii}^k \in P_i$$

where δ is the penalty factor. To create the next petal for the candidate petal set, in the optimization, the new sets of petals with adjusted scores (penalized) are used. Since the score of the petals with similarities to the already selected petals are penalized, the new selected petals tend to be different to those already selected, which leads to more diversification in the final petal set.

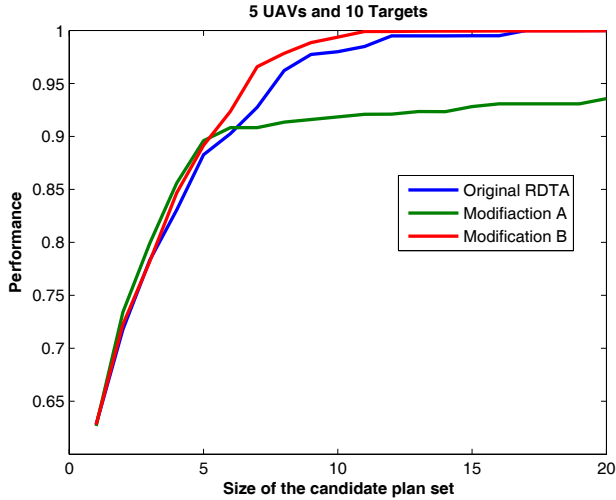


Figure 14. Comparing the performance of the original RDTA with its two modifications for a case of 5 UAVs and 10 targets.

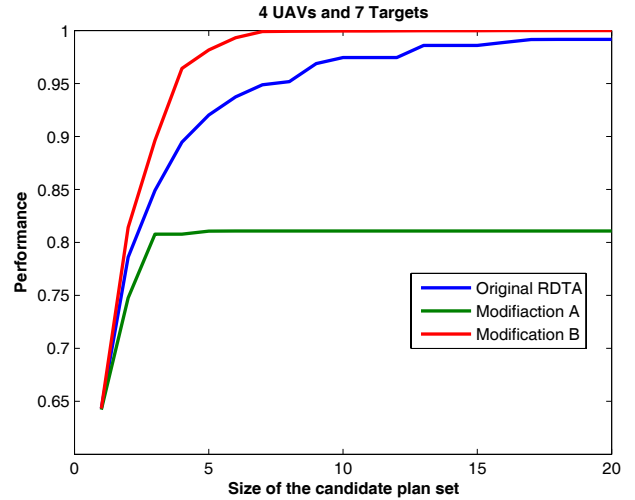


Figure 15. Comparing the performance of the original RDTA with its two modifications for a case of 4 UAVs and 7 targets.

2. Modification B

This approach uses the same penalty technique, but applies it to the petals associated with the planning UAV_{*i*} and all other UAVs. For example, consider the optimization by UAV_{*i*} which selects a set of petals for all UAVs, $\{p_{i1}^{k*}, \dots, p_{ii}^{k*}, \dots, p_{i1}^{k*}\}$. In modification A, only petals in P_{ii} that are similar to p_{ii}^k are penalized, but in modification B, this penalty is applied to all petals for all UAVs: $\forall j \in 1, \dots, N$

$$\text{If } (w \in p_{ij}^{k*} \ \& \ w \in p_{ij}^k) \Rightarrow s_{ij}^k = s_{ij}^k - \delta, \ \forall p_{ij}^k \in P_{ij}$$

3. Results

The same simulation as before is used to compare the impact of these modifications on the performance. In these simulations, however, the effect of consensus is not important and therefore is omitted. Figures 14 and 15 show the results for two sets of Monte Carlo simulations. These figures plot the performance of each algorithm (original, Modifications A and B) for different sizes of the candidate plan set. Figure 14 is the result of the base case of 5 UAVs and 10 targets and petal size of 2. Figure 15 presents the same results for a scenario of 4 UAVs, 7 targets with a petal size of 3. Both results clearly show that in modification B, the performance of the algorithm is improved by increasing the diversity and spanning a wider range of waypoints in the candidate plan sets, but the advantage of this modification over the original RDTA is more obvious in Figure 15. The plots also show that Modification A actually degrades the performance.

V. Conclusions

This paper analyzed the performance of the Robust Decentralized Task Assignment (RDTA) algorithm for different communication network topologies and showed that the communication dur-

ing the planning phase introduced in this new technique is crucial to achieve high performance. This is especially true for sparse communication networks where the slow convergence of the information consensus results in decentralized activity planning based on inconsistent data. To analyze the sensitivity of the overall performance to the candidate plan set, four selection algorithms are presented. A comparison of the performance for these algorithms clearly show the importance of accounting for the potential actions of other UAVs in the selection process. A modification of the original candidate plan selection algorithm was also presented to further improve the overall performance by increasing the robustness to inconsistencies in the information across the team.

Acknowledgements

Research funded under AFOSR grant FA9550-04-1-0458.

References

- ¹P. Chandler and M. Pachter, "Hierarchical Control for Autonomous Teams" proceedings of the AIAA *GNC*, Aug. 2001.
- ²J. Bellingham, M. Tillerson, A. Richards, J. How, "Multi-Task Assignment and Path Planning for Cooperating UAVs," *CCCO*, Nov. 2001.
- ³P. Chandler, M. Pachter, D. Swaroop, J. Fowler, *et al.* "Complexity in UAV cooperative control," *IEEE ACC* 2002, pp. 1831-1836
- ⁴E. King, M. Alighanbari, Y. Kuwata, and J. How, "Coordination and Control Experiments on a Multi-vehicle Testbed," *IEEE ACC* 2004.
- ⁵P. Chandler, "Decentralized Control for an Autonomous Team," *AIAA Unmanned Unlimited systems, Technologies, and Operations*, 2003.
- ⁶R.W. Beard and V. Stepanyan, "Synchronization of Information in Distributed Multiple Vehicle Coordinated Control," *IEEE CDC*, 2003.
- ⁷W. Ren and R.W. Beard, "Consensus of Information Under Dynamically Changing Interaction Topologies," *IEEE ACC*, 2004.
- ⁸R. Olfati-Saber and R.M. Murray, "Consensus Problems in Network of Agents With Switching Topology and Time-Delay," *IEEE transaction on Automatic Control*, Vol. 49, No. 9, Sept. 2004.
- ⁹M. Alighanbari and J. How, "Decentralized Task Assignment for Unmanned Aerial Vehicles" In *Proceedings of the IEEE CDC/ECC*, Dec. 2005.
- ¹⁰M. Moser, D. Jokanovic, and N. Shiratori, "An Algorithm for the Multidimensional Multiple-Choice Knapsack Problem," *IEICE Trans. Fundamentals*, vol. E80-A, pp. 582-589, Mar. 1997.
- ¹¹M. Alighanbari, Y. Kuwata, and J. P. How, "Coordination and Control of Multiple UAVs with Timing Constraints and Loitering," In *Proceedings of the American Control Conference*, June 2003.