Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

A Robust Hierarchical Approach To Multi-stage Task Allocation Under Uncertainty

Dongxu Li, *Student Member, IEEE* and Jose B. Cruz, Jr., *Life Fellow, IEEE*

Abstract—**A multi-stage task allocation problem is difficult to solve when the search space is large. The intrinsic uncertainty imbedded in military operations makes the problem more challenging. Scalability and robustness are recognized as two main issues. A new hierarchical algorithm is proposed to attack this problem. The algorithm has two levels. The upper level provides mutual coordination among all decision-makers; while the decision-making at the lower level is decentralized. The algorithm is not only a tasking planner but also an online feedback controller. Computational demand of the algorithm is divided into two parts. The most computationally intensive part can be implemented off-line, and the complexity of the online part is reduced significantly. Simulations show that this is potentially a good method for solving multi-stage resource allocation problems involving a large number of vehicles and tasks with robust performance.**

I. Introduction

Unmanned Aerial Vehicles (UAV) have shown great potential value in reducing the human workload in future military operations. A fleet of UAVs can work more efficiently and effectively than a single UAV, but requires a complex high-level tactical planning tool. This kind of problem involves significant challenges, which include team composition, sensor information, optimal task allocation and optimal path planning etc. In this paper, we focus on a multi-stage task allocation problem, where only target strike is concerned. A typical scenario is to assign M identified targets to a team of N available UAVs in a partially known environment. The utility of every possible assignment pair is determined according to the position and the capability of each UAV and each target. An optimal allocation scheme is searched by maximizing the sum of the utilities of all assignments. In the case when $N < M$, more than one target may be assigned to a UAV, and that UAV proceeds to the assigned targets in sequence. We call it a multistage task allocation problem, which has characteristics of both resource allocation and scheduling. This problem is very difficult to solve due to a large search space, such that scalability is of concern. Another challenge results from environmental uncertainties. In a battle field, it is likely that new pop-up targets are identified or some UAVs are destroyed in the middle of operations. The algorithm must be robust enough to compensate for these changes.

Extensive research has been done recently in this field [1]-[8]. In [2]-[4], task allocation has been formulated in the form of Mixed-Integer Linear Programming (MILP). The timing issue is addressed [3]-[4]. In this approach, the problem is treated as a deterministic optimization problem with known parameters. Although the solutions preserve global optimality, the MILP suffers from poor scalability due to the NP-hardness of such problems [11]. On the other hand, military situations are inherently dynamic and uncertain because of the UAV's sensing limitation and adversarial strategies. Thus, replanning is needed whenever the information is updated. Without considering the computation demand, the extreme behavior of churning phenomenon is discussed in [5]. A potential disaster of the simple replanning approach is illustrated, where tasks may never be accomplished. Thus, heuristics and ad-hoc methods are considered during replanning in [2]. In [5], an approach to limiting the rate of change by replanning in the frequency domain is introduced. In [6]-[7], uncertainty is taken into account in terms of optimization parameters, and risk management techniques in finance are utilized. In [6], a nonlinear integer programming problem is formulated with a constraint based on a risk measure by conditional value-at-risk. In [7], a robust approach based on the Soyster formulation on the expectation of the target scores is proposed. These approaches are based on solving combinatorial optimization problems, where solutions are conservative and scalability is still a big issue. An alternative approach in dealing with uncertainties is to formulate a stochastic optimal control problem based on the method of Model Predictive Control (MPC) [8]-[9]. With online optimization in MPC with necessary constraints, control becomes feedback of the state. However, due to the limited horizon, the actual plan is no longer optimal and feasibility can be a potential problem [9].

In this paper, different from the approaches mentioned above, a new method of solving a multistage UAV-task allocation problem is introduced. The approach combines optimization and feedback control. We assume sufficient communication capabilities among UAVs. Our goal is to design an algorithm that must be: 1) computationally scalable, i.e., capable of dealing with a relatively large number of UAVs and targets; 2) adaptive, i.e., the algorithm is able to response to environmental changes promptly; 3) robust, where controls are feasible under possible

Manuscript received March 7, 2005. This research was sponsored by the Collaborative Center for Control Science at the Ohio State University, under Grant F33615-01-2-3154 from the Air Force Research Laboratory (AFRL/VA) and the Air Force Office of Scientific Research (AFOSR).

D. Li (Tel: 614-404-5494, email: li.447@osu.edu) and J.B. Cruz (email: jbcruz@ieee.org) are with the Department of Electrical and Computer Engineering, The Ohio State University, Columbus, OH 43210 USA.

0-7803-9568-9/05/$20.00 ©2005 IEEE

uncertainties.

In the next section, we formulate the problem and derive its complexity. Motivation from a special case of the problem with a single stage is discussed, and a hierarchical algorithm is introduced in section III. Greedy decision-making under uncertainty is investigated and the computational complexity of the algorithm is analyzed. In section IV, examples are presented to demonstrate the performance. In the last section, we conclude the paper with suggestions for the future work.

II. Problem Formulation and Analysis

Given N UAVs and M targets, the objective function of a multi-task allocation problem is formulated as

$$\max\left\{J(u_1,u_2,\cdots,u_N)\right\}=\max_{u_1,\cdots,u_N}\left\{\sum_{i=1}^{M}p_i^{(j)}m_i\left(D-t_i\right)\right\}. \quad (1)$$

Here u_j denotes the ordered target set assigned to UAV j, $u_j=\left\{T_1^j,\cdots,T_{n_j}^j\right\}$ wherein T_k^j denotes a target; $p_i^{(j)}$ is the probability for target i of being destroyed by UAV j; m_i is the i^{th} target's value that implies its relative importance; t_i denotes the time that it takes for target i to be destroyed, which it is estimated by the distance to travel by the UAV to which target i is assigned; D is a large number such that $D-t_i>0$ for $\forall\ i$. Note that the order in u_j indicates the tasking sequence for UAV j, and assume that $\bigcup_{j=1}^{N}u_j=T$ and $u_i\cap u_j=\phi,\forall i\neq j$, where T is the set of targets. According to (1), it is desired that more important targets be damaged with less time, so that the mission risk at later stages is reduced.

The problem can be generalized as follows. Consider the resource set $A=\{a_1,...,a_N\}$ and the task set $B=\{b_1,...,b_M\}$. Define $G_A=\left\{G_{a_1},G_{a_2},...,G_{a_N}\right\}$ with $\bigcup_{j=1}^{N}G_{a_j}=B$ and $G_{a_i}\bigcap G_{a_j}=\varnothing$ for $\forall\ i\neq j$, where G_{a_j} is an ordered set associated with a_j. Define $\bar{G}\triangleq\left\{G_{a_j}\right\}$ and $\bar{G}_A\triangleq\{G_A\}$. Define mapping c_j: $c_j(\cdot):\bar{G}\to\mathbb{R}$, $1\le j\le N$. The objective is

$$J\left(u_1,u_2,...,u_N\right)=\sum_{j=1}^{N}c_j\left(u_j\right).$$

Clearly, J maps elements in $\bar{G}_A$ to $\mathbb{R}$. The corresponding optimization problem is

$$\max_{\{u_1,u_2,...,u_N\}}\left\{J\left(u_1,u_2,...,u_N\right)\right\}=\max_{\{u_1,u_2,...,u_N\}}\left\{\sum_{j=1}^{N}c_j\left(u_j\right)\right\}. \quad (2)$$

Notice that problem (2) is NP-hard [11]. Generally speaking, there is no efficient algorithm for large N and M. We derive the complexity using the size of the search space as a measure.

Theorem 1: (i) If $N<M$, the complexity of problem (2), ξ, is $M!\cdot\sum_{n=1}^{N}C_N^n\cdot C_{M-1}^{n-1}$, where $C_n^k=n!/\left[k!(n-k)!\right]$.

(ii) $\sqrt{2\pi}\left(2^N-1\right)\cdot M^{(M+1/2)}/e^M<\xi\left(N,M\right)<\sqrt{2\pi}\cdot 2^{N+M}\cdot M^{(M+1/2)}/e^M$ asymptotically with M.

Proof: (i) Suppose only n resources have assignments. The number of n-partitions of set B is C_{M-1}^{n-1}. The number of cases for choosing n resources out of A is C_N^n. Then, the number of allocation schemes is the summation from 1 to n, $\sum_{n=1}^{N}C_N^n\cdot C_{M-1}^{n-1}$. The total number of permutation of the tasks in B is $M!$. It follows that $\xi=M!\cdot\sum_{n=1}^{N}C_N^n\cdot C_{M-1}^{n-1}$.

(ii) On one hand, $\sum_{n=1}^{N}C_N^n\cdot C_{M-1}^{n-1}>\sum_{n=1}^{N}C_N^n=2^N-1$.

On the other hand, by Cauchy-Schwarz Inequality,

$$\sum_{n=1}^{N}C_N^n\cdot C_{M-1}^{n-1}<\sqrt{\sum_{n=1}^{N}\left(C_N^n\right)^2\cdot\sum_{n=1}^{N}\left(C_{M-1}^{n-1}\right)^2}<\sum_{n=1}^{N}C_N^n\cdot\sum_{n=1}^{N}C_{M-1}^{n-1}$$
$$=\left(2^N-1\right)\cdot\sum_{n=1}^{N}C_{M-1}^{n-1}<2^N\cdot\sum_{n=0}^{M-1}C_{M-1}^{n}=2^{N+M-1}$$

Thus, $2^N-1<\sum_{n=1}^{N}C_N^n\cdot C_M^n<2^{N+M-1}$.

By Stirling's approximation, $M!\simeq\sqrt{2\pi}\cdot M^{(M+1/2)}/e^M$, then $\sqrt{2\pi}\left(2^N-1\right)\cdot M^{(M+1/2)}/e^M<\xi<\sqrt{2\pi}\cdot 2^{N+M-1}\cdot M^{(M+1/2)}/e^M$ asymptotically with M. □

Finally, let us compare the formulation of (2) with the MILP. In the MILP, the objective can be $J=\sum_k\sum_i\sum_j c_{kij}x_{kij}$ with some necessary constraints. Here k is the index for stages; $x_{kij}\in\{0,1\}$, the binary decision variable, and $x_{kij}=1$ when b_i is assigned to a_j at stage k and $x_{kij}=0$ if it is the opposite case. In a problem with N resource elements, M tasks and K possible stages, the search space associated with the MILP formulation has $2^{N\cdot M\cdot K}$ points. This number is much larger than the upper bound derived in theorem 1. By taking into account the specific structure of the task allocation problem, the formulation of (2) creates an "easier" problem by eliminating a number of infeasible solutions.

III. Hierarchical Robust Algorithm

Task allocation is a centralized problem, which requires global information and intensive computation capability. Its solution is globally optimal, but sensitive to local changes. A small disturbance may cause drastic changes in a global optimum. In addition, local failure may be fatal for centralized systems. However, in a decentralized approach, decision- making is based on local information, such that a system has more prompt response to the environment with less computational demand. Systems become less sensitive to small disturbance and more robust against local failures. With these advantages, optimality is compromised. In general, centralized control is more suitable for deterministic or applications with mild changes; while a decentralized approach is advantageous in applications involving rapid changes and significant uncertainties. Due to the uncertain nature of UAV applications, we take a partially decentralized approach, where a tradeoff between optimality and robustness is taken into account. In this paper, a hierarchical structure of decision-making is created, in which each UAV is viewed as an independent Decision-Maker (DM) with its own sensing and decision-making capability.

A. A Single Stage Assignment Problem

In this section, we focus on a simplified assignment problem with a single stage, where each resource element a_i in A has no more than one assignment from B. This problem can be written as

$$\max J = \max\left\{\sum_{j=1}^{N}\sum_{i=1}^{M} c_{ij}x_{ij}\right\} \tag{3}$$

Subject to $x_{ij}\in\{0,1\}$, $\sum_{i=1}^{N} x_{ij}\le 1$ and $\sum_{j=1}^{M} x_{ij}\le 1$.

Here, x_{ij} is the decision variable; $c_{ij}\ge 0$ denoting the individual utility when a_i is matched to b_j.

In our approach, each a_i is viewed as an independent DM. Clearly, without additional coordination, a greedy DM a_i chooses task b_j by maximizing its payoff. To overcome potential conflicts that more than one a_i takes the same b_j, we introduce a priority sequence $\{a_{k_1},...,a_{k_N}\}$ for coordination. Given a sequence, DMs in A make choices sequentially among the currently available resources in B. After each of the a_i's has one assignment, objective (3) can be evaluated. In this way, a mapping between sequences and $\mathbb{R}$ can be established. A two-level algorithm can be designed as follows. The upper level is to find a proper sequence; whereas at the lower level, each a_i makes an individual decision in sequence. The following theorems prove that a proper sequence provides a valid coordination for all DMs. We first consider $N=M$ and then other cases.

Lemma 2: If $N=M$, in solution $X^*=\left(x_{ij}^*\right)_{N\times N}$ of problem (3), there exists $x_{mn}^*=1$, $1\le m,n\le N$ such that $c_{mn}=\max_{1\le j\le N}\{c_{mj}\}$.

Proof: Prove by contradiction. Suppose there exists no such $x_{mn}^*=1$. For any i, $0\le i\le N$, there is an unique j such that $x_{ij}^*=1$ in X^*. Define function $f(i)=j$ for any i. There is at least one j' with $c_{ij'}=\max_{1\le m\le N}\{c_{im}\}$. Define $g(i)=j'$. Clearly, $f(i)\ne g(i)$, $\forall\ 0\le i\le N$. Both $f(\cdot)$ and $g(\cdot)$ are one-to-one functions and have a unique inverse, i.e., $i=f^{-1}(j)$ ($i=g^{-1}(j')$). Starting from any i, construct a sequence as:

$$i\to g(i)\to f^{-1}(g(i))\to g\left(f^{-1}(f(i))\right)\to f^{-1}\left(g\left(f^{-1}(g(i))\right)\right)\to\cdots.$$

In this sequence, a circulation involving up to N elements takes place. Consider one complete circulation as

$$k\to f^{-1}(k)\to g\left(f^{-1}(k)\right)\to f^{-1}\left(g\left(f^{-1}(k)\right)\right)\to\cdots\to k.$$

Then, $c_{f^{-1}(k),k}+c_{f^{-1}\left(g\left(f^{-1}(k)\right)\right),g\left(f^{-1}(k)\right)}+\cdots+c_{f^{-1}(g(\cdots)),g(\cdots)}$

$$<c_{f^{-1}(k),g\left(f^{-1}(k)\right)}+c_{f^{-1}\left(g\left(f^{-1}(k)\right)\right),g\left(f^{-1}\left(g\left(f^{-1}(k)\right)\right)\right)}+\cdots+c_{f^{-1}(g(\cdots)),k}$$

Thus, if $x^*_{f^{-1}(k),k}$, $x^*_{f^{-1}\left(g\left(f^{-1}(k)\right)\right),g\left(f^{-1}(k)\right)}$ …, $x^*_{f^{-1}(g(\cdots)),g(\cdots)}$ are replaced by $x_{f^{-1}(k),g\left(f^{-1}(k)\right)}$, $x_{f^{-1}\left(g\left(f^{-1}(k)\right)\right),g\left(f^{-1}\left(g\left(f^{-1}(k)\right)\right)\right)}$, …, $x_{f^{-1}(g(\cdots)),k}$, the payoff is increased. Thus, $X^*=\left(x_{ij}^*\right)_{N\times N}$ cannot be optimal. This is a contradiction and the theorem is proved. □

Remark 2: Lemma 2 states that in an optimal solution of (3), at least one DM achieves its maximum payoff.

Theorem 3: If $M=N$, there exists a sequence $\{a_{k_i}\}_{i=1}^{N}$, $a_{k_i}\in A$, such that a solution to the following multi-level programming problem in (4) is a solution to problem (3).

$$\max J_1(a_{k_1})=\max_{x_{k_1 j}}\left\{\sum_{j=1}^{N}c_{k_1 j}x_{k_1 j}\right\},\ \text{subject to}\ \sum_{j=1}^{N}x_{k_1 j}\le 1$$

$$\vdots$$

$$\max J_n(a_{k_n}\mid a_{k_1}...a_{k_{n-1}})=\max_{x_{k_n j}}\left\{\sum_{j=1}^{N}c_{k_n j}x_{k_n j}\right\},\ \text{subject to}\ \sum_{j=1}^{N}x_{k_n j}\le 1$$

$$\vdots$$

$$\max J_N(a_{k_N}\mid a_{k_1}...a_{k_{N-1}})=\max_{x_{k_N j}}\left\{\sum_{j=1}^{N}c_{k_N j}x_{k_N j}\right\},\ \text{subject to}\ \sum_{j=1}^{N}x_{k_N j}\le 1$$

Subject to $x_{ij}\in\{0,1\}$, and $\sum_{i=1}^{N}x_{ij}\le 1$. (4)

Proof: Start with N resources, apply lemma 2 to find a_{k_1} that achieves its best utility by choosing task j_i from the currently unassigned tasks. Eliminate resource a_{k_1} and the j_i^{th} task and the problem is reduced to a similar one with one less dimension. Follow the same procedure until all N resources have one task. The resulting sequence of a_i's is a valid sequence. □

Remark 3.1: Theorem 3 states that with a proper sequence, the sequential greedy decision-making determines an optimal solution of the problem in (3).

Remark 3.2: Priority sequences virtually provide valid coordination for independent DMs. This sequence does not imply the order that a_i's make decisions in reality.

Similar conclusions can be extended to other cases.

Corollary 4: If $N<M$ or $N>M$,

i) If $X^*=\left(x_{ij}^*\right)_{N\times M}$ is a solution to problem (3), then there exists $x_{mn}^*=1$ in X^*, $1\le n\le N$ and $1\le m\le M$ such that $c_{mn}=\max_{1\le j\le M}\{c_{mj}\}$.

(ii) There exists a sequence of the elements in A, such that a solution to (4) is a solution to (3).

Proof: (i) For $N<M$ ($N>M$), create $|N-M|$ dummy resources in A (tasks in B), and all of the related utilities are zero. Then the same argument in lemma 2 and theorem 3 can be applied. Part (ii) follows. □

B. Hierarchical Partially Decentralized Planner

For a multi-stage problem, sequential decision-making and cooperation play important roles in the algorithm. Consider the following individual objective function for DM a_j,

$$\max\left\{J_j\left(u_j \mid B^k\right)\right\} = \max_{u_j}\left\{c_j\left(u_j \mid B^k\right)\right\}. \tag{5}$$

In (5), subscript j denotes the DM a_j; B^k is the set of the unsigned tasks at stage k when a_j is making decisions. Argument u_j in (5) implies that DM a_j seeks the best strategy to fulfill all the unsigned tasks. According to (5), a specific optimization problem related to (1) can be written as

$$\max_{u_j}\left\{J_j(u_j \mid B^k)\right\} = \max_{u_j}\left\{\sum_{i=1}^{M_k} p_i^{(j)} m_i\left(D-t_i\right) \mid B^k\right\}. \tag{6}$$

Here, M_k denotes the number of tasks in B_k. This is a scheduling problem.

The hierarchical algorithm is described as follows. We start with the lower level. Given sequence $\{a_1, a_2, ..., a_N\}$, the first DM a_1 is the first to choose and suppose $\{b_1, ..., b_{M^1}\}$ is the best sequence for problem (6). Then, DM a_1 proceeds to the first task b_1. Meanwhile, there are $N-1$ DMs for cooperation. At the moment, a_1 is going to b_1 and this information is shared by all other DMs. The second DM a_2 faces a similar problem with $M_2 = M_1 - 1$ tasks, and it goes to the first task in the solution sequence. DM a_3 follows the same procedure and so on until all N DMs have exactly one assignment. There are still $(M-N)$ unsigned tasks ($N < M$). The time to finish the current assigned task is calculated for each DM. The DM with the minimum time is the next to make a decision. The same procedure is followed until all M tasks are assigned. In the case when more than one DM finishes its current task at the same time, the next DM is chosen randomly. With the lower level, objective (2) can be evaluated provided a sequence of DMs. The upper level is to solve for an optimal sequence according to (2). The flowchart is shown in Fig. 1.

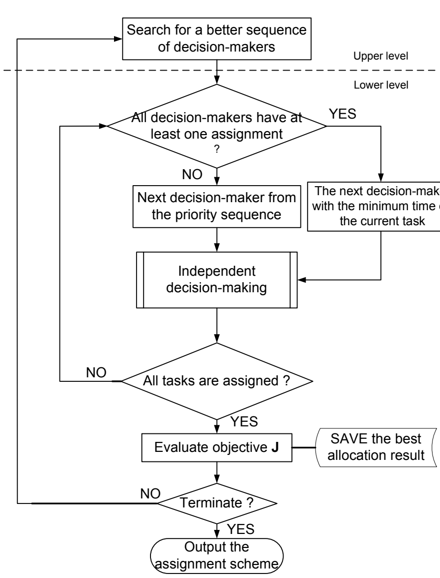

Fig.1. Flowchart of the Hierarchical Planner

C. On-line Decentralized Feedback Controller

Fig.1 illustrates the procedure of the algorithm for planning under the assumption that each UAV can successfully fulfill its assigned tasks, which may not be true in reality. During the online implementation, decisions at the lower level are made based on the latest information. In this case, the lower level acts as an online feedback controller. Without unexpected events during actual operations, the implementation will coincide with the plan. However, it is likely that some uncertain event, such as loss of UAVs or pop-up of new targets, might happen. The feedback mechanism at the lower level helps compensate for the uncertainties such that the algorithm is robust. Finally, the optimization for each DM is similar to the procedure of Open-Loop-Feedback Control [12]. A DM solves a complete optimal tasking sequence but only executes the first control in the sequence.

D. Greedy Decision-Making at the Lower Level

At the lower level, greedy decision-making can be applied, where only the next task is solved instead of the whole sequence. Then, the optimization problem in (6) becomes

$$\max\left\{J_j(b_l \mid B^k)\right\} = \max_{b_l \in B^k}\left\{p_i^{(j)} m_i\left(D-t_i\right) \mid B^k\right\}. \tag{7}$$

Under uncertainty, a decision based on (7) may be the same as that based on (6). In this section, a bound on uncertainty is given such that a solution to (7) is the first task in a sequence that solves (6). In this paper, uncertainty is characterized by the factor $p_i^{(j)}$, which includes the probability of survival for UAV j and the probability of destruction for target i. Both probabilities depend on target i and UAV j. For simplicity, we consider UAV j and use notations p_i^S and p_i^D to represent the probabilities respectively. Suppose that the events of survival and destruction are independent, then, $p_i^{(j)} = p_i^D \cdot \prod_{n=1}^{i-1} p_n^S$. Equation (6) can be rewritten as

$$\max_{b_l \in B^k}\left\{p_l^{(j)} m_l\left(D-t_0^l\right) + \max\left\{J_j^{k+1}(u_j \mid B^k / \{b_l\})\right\}\right\} \tag{8}$$

Here, $B^k / \{b_l\}$ is the set of the unassigned tasks without b_l; t_0^l denotes the time it takes for UAV j to reach target l. Define $z_l \triangleq p_D^l m_l\left(D-t_0^l\right)$, and denote $Z_1 \triangleq \max_l\{z_l\}$. Then, define $Z_2 = \max_l\left\{z_1, \cdots, z_{m-1}, z_{m+1}, \cdots, z_{M^k}\right\}$ with $m = \arg\max_l\{z_l\}$. Let the upper bound of the survival probability at any stages be $\bar{p} = \max\left\{p_i^S\right\}$. Clearly,

$p_i^{(j)} \le p_i^D \cdot \bar{p}^{(i-1)}$. Define $\lambda = Z_2/Z_1$.

Theorem 5: If $\bar{p} \le (1-\lambda)/(2-\lambda)$, a solution to (7), b_q, is a solution to (8).

Proof: For any $s \ne q$, because $p_i^{(j)} \le p_i^D \cdot \bar{p}^{(i-1)}$,

$$\begin{aligned} & p_s^{(j)} m_s\left(D-t_0^s\right)+\max\left\{\sum_{i=2}^{M^k} p_i^{(j)} \cdot m_i\left(D-\left(t_0^s+\Delta t_i^s\right)\right)\right\} \\ & \le z_s+\max\left\{\sum_{i=2}^{M^k} \bar{p}^{(i-1)} p_i^D \cdot m_i\left(D-\left(t_0^s+\Delta t_i^s\right)\right)\right\} \\ & \le z_s+\sum_{i=2}^{M^k} \bar{p}^{(i-1)} Z_1 = z_s+\frac{\bar{p}-\bar{p}^{M^k}}{1-\bar{p}} Z_1 \\ & = z_s+\frac{\bar{p}}{1-\bar{p}} Z_1-\frac{\bar{p}^{M^k}}{1-\bar{p}} Z_1 \end{aligned} \tag{9}$$

Because $\bar{p} \le (1-\lambda)/(2-\lambda)$; function $\bar{p}/(1-\bar{p})$ is increasing with respect to $\bar{p}$ and $z_s \le Z_2$,

$$\begin{aligned} (9) & \le Z_2+(1-\lambda) Z_1-\frac{\bar{p}^{M^k}}{1-\bar{p}} Z_1 \\ & = Z_2+(1-\frac{Z_2}{Z_1}) Z_1-\frac{\bar{p}^{M^k}}{1-\bar{p}} Z_1 = Z_1-\frac{\bar{p}^{M^k}}{1-\bar{p}} Z_1 \end{aligned} \tag{10}$$

Since $Z_1 = p_q^{(j)} m_q\left(D-t_0^q\right)$,

$$\begin{aligned} (10) & < p_q^{(j)} m_q\left(D-t_0^q\right)+\max\left\{\sum_{i=2}^{M^k-1} p_i^{(j)} m_i\left(D-\left(t_0^q+\Delta t_i^q\right)\right)\right\}-\frac{\bar{p}^{M^k}}{1-\bar{p}} Z_1 \\ & < p_q^{(j)} m_q\left(D-t_0^q\right)+\max\left\{\sum_{i=2}^{M^k-1} p_i^{(j)} m_i\left(D-\left(t_0^q+\Delta t_i^q\right)\right)\right\}. \end{aligned}$$

Thus b_q solves the problem (8). □

By theorem 4, with a decrease of probability of survival, the decision-making of UAV j becomes more and more myopic. Finally, solutions to (7) and (8) become the same.

E. Algorithm Analysis

The two-level approach is suboptimal from the optimization point of view. It is because a special structure has been imposed on the allocation calculation, and a best plan is searched within a subspace of the original search space. From the motivation problem with a single stage, this structure captures some characteristics of optimal allocations. On the other hand, the sacrifice of optimality provides a greater degree of freedom for improving robustness and adaptability, because the decentralized decision-making at the lower level acts as a feedback controller during the mission. So far, we have assumed that the information about the battlefield is shared by all UAVs through sufficient communication links.

It is worth noting that the hierarchical approach is conceptually identical to a two-level Stackelberg game problem [13] or bi-level programming [14], where the optimization at the upper level depends on that at the lower level. In this approach, a best priority sequence determines a starting point for the mission, and the decentralized feedback control is used at the lower level to improve robustness.

Suppose a greedy decision-making is applied at the lower level. The complexity of the lower level is

$$\xi_L = M + (M-1) + \cdots + 1 = M(M+1)/2.$$

The complexity of the upper level is $N! \cdot \xi_L$. Although this is a hard problem, the hierarchical method is still applicable because: 1) the number of UAVs is normally small compared to the number of tasks; 2) the optimization at the upper level can be implemented off-line.

IV. Simulation Results and Analysis

Small battles were created to illustrate the performance of the algorithm. A scenario with 3 UAVs and 8 targets were selected with the necessary parameters listed in Table I and II.

TABLE I
PARAMETERS OF UAVS AND TARGETS

UAV			
	UAV 1	UAV 2	UAV 3
Position	(5, 0)	(5, 0)	(5, 0)
Velocity	4	5	4

Target								
	1	2	3	4	5	6	7	8
Position	(1,10)	(2,2)	(3,7)	(7,1)	(8,8)	(4,3)	(5,9)	(6,5)
Value	5	7	4	6	5	4	4.5	4.5

TABLE II
PROBABILITY OF DESTRUCTION

	T1	T2	T3	T4	T5	T6	T7	T8
UAV1	0.9	0.9	0.9	0.7	0.7	0.7	0.7	0.7
UAV2	0.7	0.7	0.7	0.7	0.7	0.7	0.9	0.9
UAV3	0.7	0.7	0.7	0.9	0.9	0.9	0.7	0.7

Consider that the probability of survival is 1 and the global optimal plan generated according to objective (1) is illustrated in Fig. 2. In Fig. 3, the result of the hierarchical algorithm is presented. In both figures, the tasking sequence of each UAV is indicated by arrows, where UAV and target number are indicated. The performance index of the global optimal plan illustrated in Fig. 2 is 120.3, and that for the suboptimal plan in Fig. 3 is 110.7 with $D = 5$.

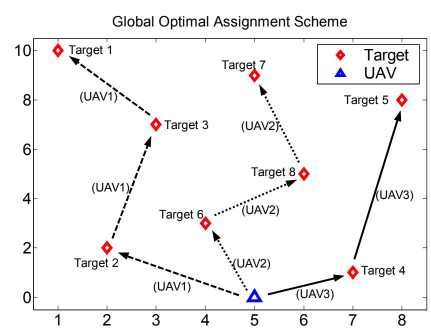

Fig. 2 Global Optimal Assignment Scheme

Next, we created 50 similar problems based on the parameters used in this scenario, where the positions of the targets and UAVs were randomly generated. The positions of the targets were assumed to be uniformly distributed in a 10×10 square, and the positions of the UAVs were uniformly distributed on the interval [0,10] on the x-axis. All

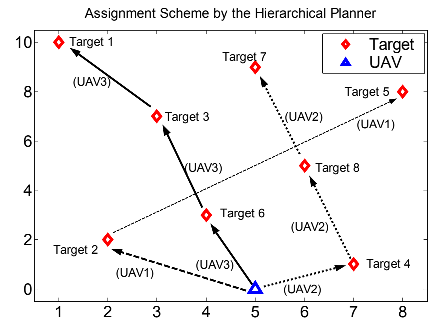

Fig. 3 Assignment Scheme by the Hierarchical Planner

of the three UAVs were assumed to have the same initial position. The values of the objective functions achieved by the hierarchical algorithm are plotted in Fig. 4, which are relative values normalized by the global optimum for each problem. To eliminate the influence of the constant D in (1), the origin of the performance index is set as the worst plan with the least objective evaluation according to (1) in each random problem. The plots shown in Fig. 4 are moving averages with a window size of 5. The averages of performance index of the suboptimal plan for a total of 50 problems by the algorithm are 0.85 and 0.88 for non-greedy and greedy DMs. Let the CPU time it takes to calculate the global optimal assignment plan be 1. The CPU times used by the hierarchical algorithm are 0.47 and 5×10^{-5} for non-greedy and greedy cases respectively.

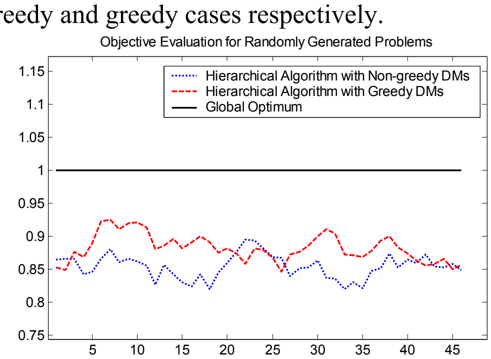

Fig. 4 Objective Evaluation for Randomly Generated Problems

For the scenario selected, the suboptimal results obtained by the hierarchical algorithm can reach about 90% of the global optimum on average. The result with greedy decision-making at the lower level is a little better. This is because the decision-making of each UAV depends not only on its own choices but also on the decision-making of other UAVs at later stages. However, only the former dependence is considered in (8). Furthermore, the hierarchical algorithm helps reduce computation dramatically. The computationally intensive part can be implemented offline, and the online optimization can be implemented to compensate for possible uncertainties. The failures in the extreme cases proposed in [5] can be prevented. This can potentially be a good online tactical operational tool for UAV applications.

V. Conclusions and Future Work

The hierarchical approach is motivated by a special problem with a single stage. It is a hybrid of optimization with decentralized feedback control. Computation has been divided into two parts, where the computationally intensive part can be done offline. Optimization at the upper level provides central coordination; whereas the decentralized feedback at the lower level enables prompt responses to environmental changes. Priority sequences capture the major characteristics of task allocation problems. The method generates good assignment plans with significant savings on computations. In practice, feedback at the lower level can compensate for possible uncertainties such that robustness is improved, especially when online replanning based on a centralized approach is computationally prohibitive.

The following issues need to be further investigated: 1) a fast algorithm for solving the upper level optimization problem, 2) coordination among DMs or replanning algorithm during operations when there are dramatic changes, 3) including practical constraints on timing or path planning.

References

[1] P.R. Chandler and M Pachter, "Hierarchical Control for Autonomous Teams", *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Montreal, Quebec, Canada, August, 2001.

[2] J. Bellingham, M. Tillerson, A. Richards, and J. How, "Multi-Task Allocation and Path Planning for Cooperating UAVs," *Second Annual Conference on Cooperative Control and Optimization*, Nov 2001.

[3] M. Alighanbardi, Y. Kuwata and J. How, "Coordination and Control of Multiple UAVs with Timing Constraints and Loitering", *Proceeding of the American Control conference*, 2003.

[4] C. Schumacher, P. Chandler, M. Pachter, and L. Pachter, "Constrained Optimization for UAV Task Assignment", *AIAA Guidance, Navigation, and Control Conference*, Providence, RI, August 2004.

[5] M. Alighanbari, L.F. Bertuccelli, and J.P. How, "Filter-Embedded UAV Task Assignment Algorithms for Dynamic Environments", *AIAA Guidance, Navigation, and Control Conference*, Providence, RI, August 2004.

[6] P. Krokhmal, R. Murphey, P. Pardalos, S. Uryasev, and G. Zrazhevsky "Robust Decision Making: Addressing Uncertainties in Distributions", In: S. Butenko et al. (Eds.) *Cooperative Control: Models, Applications and Algorithms*, Kluwer Academic Publishers, 165-185, 2003.

[7] L.F. Bertuccelli, M. Alighanbari and J.P. How, "Robust Planning for Coupled Cooperative UAV Missions", *43rd IEEE Conference on Decision and Control,* Atlantis, Paradise Island, Bahamas, 2004.

[8] C.G. Cassandras and W. Li, "A Receding Horizon Approach for Solving Some Cooperative Control Problems", *Proceedings of the 41st IEEE Conference on Decision and Control,* Las Vegas, NV, 2002.

[9] A. Richards and J.P. How, "Model Predictive Control of Vehicle Maneuvers with Guaranteed Completion Time and Robust Feasibility", *American Control Conference*, Denver, CO, 2003.

[10] M. Padberg, *Linear Optimization and Extensions, 2nd Edition*. Springer, New York, 1999.

[11] C.H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1982.

[12] D.P. Bertsekas, *Dynamic Programming and Optimal Control: Volume 1, 2nd Edition*. Athena Scientific, Belmont, Massachusetts, 2000.

[13] M. Simaan and J. B. Cruz Jr., On the Stackelberg strategy in nonzero-sum games, Journal of Optimization Theory and Applications, V. 11, 533 - 555, No. 5, 1973.

[14] B. Colson, P. Marcotte and G. Savard, Bilevel programming: A survey, A Quarterly Journal of Operations Research, 3, 87– 107, 2005.