Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 49-17 (2016) 314–319

PSO-based Optimal Task Allocation for Cooperative Timing Missions

Gyeongtaek Oh * **Youdan Kim** ** **Jaemyung Ahn** ***
Han-Lim Choi ****

* *Department of Mechanical and Aerospace Engineering, Seoul National University, Seoul, Republic of Korea (e-mail: laimo@snu.ac.kr)*
** *Department of Mechanical and Aerospace Engineering, Seoul National University, Seoul, Republic of Korea (e-mail: ydkim@snu.ac.kr)*
*** *Department of Aerospace Engineering, Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea (e-mail: jaemyung.ahn@kaist.ac.kr)*
**** *Department of Aerospace Engineering, Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea (e-mail: hanlimc@kaist.ac.kr)*

Abstract: An optimal task allocation algorithm based on Particle swarm optimization (PSO) is proposed for cooperative timing missions that require involvement of multiple agents. The optimal solution can be utilized in the centralized operation of multi-agent system as well as a benchmark solution of the decentralized task allocation algorithm. However, the optimal solution requires significant computations because this problem is known as NP-hard. Therefore, PSO-based approach can be an alternative because it can be used to obtain a near optimal solution with reduced computational load. In this study, the conventional PSO-based algorithm is modified by adopting graph theory. Numerical simulations are carried out to demonstrate the performance of the proposed algorithm.

Keywords: Task allocation, Multi-UAV planning, PSO, Cooperative timing mission, Cooperative control

1. INTRODUCTION

Cooperative control makes multiple agents perform effective and efficient operations for complex and dangerous missions. In cooperative control problems, this study focuses on the task allocation (TA) module that determines which agent conducts which tasks. Also, a type of tasks requiring multiple agents at the same time is considered. According to the well-known taxonomy for TA (Gerkey and Matarić (2004)), this kind of problem is called single-task robots(ST)–multi-robot tasks(MR) problem, which is referred to as a *coalition formation* problem where the coalition means a group of agents to conduct a common task (Sandholm and Lesser (1995)).

Previous research on the coalition formation problem can be classified as a centralized and decentralized approach. In the centralized approach (Beard et al. (2002); Sujit et al. (2008)), a ground station or a leader agent distributes tasks to coalitions by utilizing the information of the entire agents. In this approach, conflicts between agents do not occur, and an optimal solution can be obtained. However, a failure of the decision maker is a serious potential risk to the entire agents. In the decentralized approach (Shehory and Kraus (1998); Whitten et al. (2011)), on the other hand, all the agents participate in the decision making process. Because the agents negotiate with each other according to the designed rule, a breakdown of the single agent usually does not lead to a mission failure. However, the communication burden for the negotiation is generally heavier than that of the centralized approach, and it is hard to obtain the optimal solution.

In this context, the optimal coalition formation algorithm can be obtained by a centralized approach, which can be used as a benchmark solution for the decentralized approach. Usually, the optimal solution requires significant computations because the coalition formation problem is known as NP-hard problem (Gerkey and Matarić (2004)). To deal with this problem, a particle swarm optimization (PSO) based algorithm was introduced, and optimal solution of the coalition formation problem was obtained for cooperative timing mission (Sujit et al. (2008)). The solution was utilized as a benchmark solution of the distributed approach (Manathara et al. (2011)). The presented algorithm, however, does not guarantee the simultaneous arrival of the coalition to the assigned tasks.

In this study, the existing PSO-based algorithm for cooperative timing missions (Manathara et al. (2011)) is modified. To guarantee the simultaneous arrivals, a graph theoretic constraint and a cost function of each particle is introduced. Three scenarios of suppression of enemy air defenses (SEAD) mission are considered for case studies. Numerical results comparing with sequential greedy al-

10.1016/j.ifacol.2016.09.054

gorithm demonstrate the satisfactory performance of the proposed algorithm.

This paper is organized as follows: Section 2 describes the problem statement in detail. After a brief introduction of the centralized greedy algorithm in Section 3, the modified PSO-based TA algorithm is proposed in Section 4. Finally, numerical results and conclusion are summarized in Section 5 and 6, respectively.

2. PROBLEM STATEMENT

Let us consider a TA problem with N agents and M tasks where some task locations should be simultaneously visited by multiple agents as shown in Fig. 1. The number inside the parenthesis near each task location indicates the number of required agents for each task, which is defined based on the properties of the task. It is assumed that the time required to perform a task is negligible compared to the travel time of an agent.

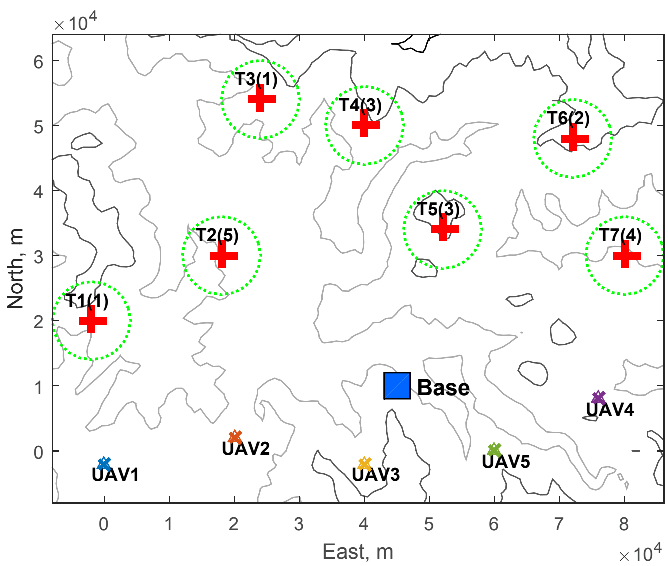

Fig. 1. Example scenario of cooperative timing mission.

The optimal solution of the TA problem is the *best* sequences of tasks for each agent that minimize the cost function. The considered problem can be formulated in terms of the integer programming problem as follows:

$$\underset{\mathbf{p}^{(i)},\forall i\in\mathcal{I}}{\text{Minimize}} \quad J = \max_{k\in\mathcal{K}} t_k(\mathbf{P}) \tag{1}$$

$$\text{subject to} \quad n(\mathbf{a}_k(\mathbf{P})) = Z_k, \ \forall k \in \mathcal{K} \tag{2}$$

$$\texttt{isDAG}(G) = 1, G = (\mathcal{K}, \mathcal{E}(\mathbf{P})) \tag{3}$$

where $\mathcal{I} \triangleq \{1, 2, ..., N\}$ and $\mathcal{K} \triangleq \{1, 2, ..., M\}$ are the index sets of the agents and tasks, respectively, $\mathbf{p}^{(i)}$ is a path vector representing the sequence of tasks allocated to agent i, t_k is the termination time of the task k, and $\mathbf{P}$ is the N by M path matrix having the information of the entire agents' path vectors. The i-th row of the path matrix represents the path vector of agent i, i.e., $\mathbf{P}(i, m) = k$ is equivalent to $\mathbf{p}^{(i)}(m) = k$ and $\mathbf{P}(i, m) = 0$ means $\mathbf{p}^{(i)}(m) = \emptyset$.

The cost function J means the mission completion time. Because some tasks should be visited at the same time, t_k is defined as the arrival time of the latest agent among the coalitions for the task k, which is expressed as

$$t_k(\mathbf{P}) = \max_{i\in\mathbf{a}_k} \ t_{ETA}(i, k) \tag{4}$$

where $\mathbf{a}_k$ is a set of indices of agents assigned to the task k, and $t_{ETA}(i, k)$ denotes the estimated time of arrival (ETA) of the agent i to the task k, which can be expressed as

$$t_{ETA}(i, k) = \begin{cases} d_{ik}/v_i, & \text{if } \ \mathbf{p}^{(i)}(1) = k \\ t_l + d_{lk}/v_i, & \text{otherwise} \end{cases} \tag{5}$$

where v_i is the average speed of the agent i. If $\mathbf{p}^{(i)}(1) = k$, that is, the task k is the first task of the agent i, then d_{ik} is the distance between the current location of the agent i and the task k. Otherwise, d_{lk} is the distance between the task l and the task k, where l is the index of the task conducted by the agent i prior to the task k, which can be expressed as

$$l = \mathbf{p}^{(i)}(b_k^{(i)} - 1) \tag{6}$$

where $\mathbf{p}^{(i)}(b)$ is the b-th element of $\mathbf{p}^{(i)}$, and index $b_k^{(i)}$ satisfies the following relationship:

$$\mathbf{p}^{(i)}(b_k^{(i)}) = k \tag{7}$$

Equation (2) defines a constraint on the size of the coalition for the task k where $n(\cdot)$ denotes the cardinality of a set, and Z_k is the number of required agents for the task k.

Note that Eq. (3) is introduced to disregard the TA results whose involved agents fail to simultaneously arrive at their allocated task locations. This process can be conducted by filtering out the TA results that generate the directed cycle in the dependency graph $G = (\mathcal{K}, \mathcal{E}(\mathbf{P}))$, which represents the precedence between the allocated tasks (Balmas (2004)). The directed edge set $\mathcal{E}(\mathbf{P})$ is defined as follows:

$$\mathcal{E}(\mathbf{P}) = \{(\mathbf{p}^{(i)}(b), \mathbf{p}^{(i)}(b+1)) | i \in \mathcal{I}, \ b \in \{1, ..., n(\mathbf{p}^{(i)}) - 1\}\} \tag{8}$$

Since the simultaneous arrival fails when the dependency graph contains a directed cycle, designating the type of the dependency graph as a directed acyclic graph (DAG), which is a directed graph with no directed cycles (Gross and Yellen (2003)), realizes the filtering. The function $\texttt{isDAG}(G)$ is one if the graph G is DAG and zero otherwise. Detailed explanation on the DAG constraint is provided in the next subsection.

The TA problem defined in Eqs. (1)∼(3) is a coalition formation problem where coalitions may overlap, which can be categorized as a set covering problem (SCP) (Shehory and Kraus (1998)). The original SCP is known as NP-hard (Cormen et al. (2001)). Since the original SCP does not consider the order of task allocation associated with Eq. (3), the problem considered in this study is at least as complex as the SCP. When the size of the problem becomes larger, the centralized scheme requires significant amount of time to determine the optimal solution. In this study, a PSO-based optimal TA algorithm is presented to solve the complex TA problem with less computation time.

Directed Acyclic Graph Constraint on Dependency Graph

The TA algorithm for cooperative timing missions should provide the *proper* order of waypoints so that the tasks