

## Technical Paper

Int'l J. of Aeronautical & Space Sci. 12(2), 163–174 (2011)  
DOI:10.5139/IJASS.2011.12.2.163



# Genetic Algorithm Based Decentralized Task Assignment for Multiple Unmanned Aerial Vehicles in Dynamic Environments

Hyunjin Choi\*, Youdan Kim\*\* and Hyounjin Kim\*\*\*

*School of Mechanical and Aerospace Engineering, Seoul National University, Seoul 151-742, Korea*

## Abstract

Task assignments of multiple unmanned aerial vehicles (UAVs) are examined. The phrase “task assignment” comprises the decision making procedures of a UAV group. In this study, an on-line decentralized task assignment algorithm is proposed for an autonomous UAV group. The proposed method is divided into two stages: an order optimization stage and a communications and negotiation stage. A genetic algorithm and negotiation strategy based on one-to-one communication is adopted for each stage. Through the proposed algorithm, decentralized task assignments can be applied to dynamic environments in which sensing range and communication are limited. The performance of the proposed algorithm is verified by performing numerical simulations.

**Key words:** Decentralized Task Assignment, Multiple unmanned aerial vehicles, Combinatorial Optimization, Genetic Algorithm, Negotiation

## 1. Introduction

The operation of multiple UAVs requires decision making processes such as task planning (Murray, 2007). Task assignment has been regarded as a combinatorial optimization problem in which combinations between UAVs and various tasks must be deciphered (Papadimitriou and Steiglitz, 1982). Examples of combinatorial optimization problems include the traveling salesman problem (TSP) or the vehicle routing problem. Finding exact solutions are very difficult because combinatorial optimization problems possess non-deterministic polynomial time (NP), which results in computational complexity. Two approaches have been developed to overcome this complexity. One approach is a mathematical programming approach such as mixed integer linear programming (MILP) (Chandler et al., 2002; Richards et al., 2002; Schumacher et al., 2004). The second approach is a meta-heuristic algorithm such as the genetic algorithm (GA) (Eun and Bang, 2009; Potvin, 1996; Shima et al., 2006) and particle swarm optimization (Cruz

et al., 2004). Mathematical programming approaches often provide solutions that are better in quality than solutions derived from meta-heuristic algorithms, but mathematical programming usually requires much more computation time than its counterpart. Conversely, the meta-heuristic approach obtains solutions quickly, however the quality of the solution may be poor.

Communication among multiple UAVs within dynamic environments also presents difficulties. Off-line task assignment sufficiently allows UAVs to perform missions in stationary environments. However, on-line task assignment becomes highly necessary in changing environments. If all UAVs share information, task assignment can be regarded as centralized.

The implementation of on-line task assignments to centralized systems is difficult because communication and computation loads. The centralized systems gather task information of each UAV and process the information to find an effective task assignment. Therefore, the amount of the information is enormous to the centralized systems. In actual

Copyright © 2011. The Korean Society for Aeronautical and Space Science

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

© Received 6 April, 2011, Accepted 1 June, 2011

\* Graduate Student

\*\* Professor, Corresponding author

E-mail: ydkim@snu.ac.kr Tel: +82-2-880-7398 Fax: +82-2-887-2662

\*\*\* Associate Professor

missions, communication between UAVs is restrictive. To compensate for communication restrictions, decentralized and distributed task assignment approaches have been developed (Alighanbari and How, 2005). Decentralization schemes require specified communication structures, such as an auction scheme or negotiation scheme. Furthermore, each UAV within a dynamic environment must decide and communicate the task plan.

This paper describes a decentralized task assignment algorithm for multiple UAVs within a dynamic environment. Each UAV is assumed to exchange environment information with other UAVs. The balance of the information between UAVs can be broken. Thus, a communication strategy based on negotiation (Sujit et al., 2007) is adopted to manage unbalanced situations. The strategy is divided into two stages. Stage I is the order optimization stage, and Stage II is the communications and negotiation stage. In Stage I, each UAV adjusts its task order, which in turn reduces costs by using GA. Task exchange occurs at Stage II. On-line decentralized task assignment can be performed through these stages. Each UAV is autonomously assigned proper tasks.

This paper is organized as follows. In Section 2, the problem is formulated by considering the task assignment scenario, combinatorial optimization, and path planning with cost prediction. In Section 3, the decentralized task assignment algorithm is proposed, which consists of a genetic algorithm for the first stage and negotiation for the second stage. Section 4 explains the results from the numerical simulation. Finally, Section 5 concludes the study.

## 2. Task Assignment Problem of Multiple UAVs

### 2.1 Missions for multiple UAVs

UAVs are widely used for surveillance and reconnaissance missions. Wide area search and munitions (WASM) or intelligence, search and reconnaissance (ISR) are examples of such missions incorporating the multiple UAVs.

This study documents the task assignment for a specified mission composed of multi-targets and multi-tasks. First, multiple UAVs and several known restricted regions were carefully evaluated. Then, task assignment was performed with respect to certain path constraints. The WASM mission was chosen that the UAVs would perform in this study. Thus, two sub-tasks for each target were designated as “Classification with Attack” and “Verification”. Thus, the total number of tasks was twice the number of targets. Accordingly, the UAV group was expected to sufficiently perform multiple tasks, as well as adjust the task order.

### 2.2 Combinatorial Optimization Problem

Multiple UAV task assignment is considered a combinatorial optimization problem. Combinatorial optimization possesses NP-hard computational complexity (Papadimitriou and Steiglitz, 1982); thus, a specified method for obtaining an optimal solution does not exist. The only known method is to search all possible cases. Typical search methods include the exact search method or the heuristic search method.

A formal combinatorial optimization problem follows a certain formulation. For an  $N_v$  amount of UAVs, the UAVs set  $\mathbf{V}$  can be defined as follows.

$$\mathbf{V} \triangleq \{V_1, V_2, \dots, V_{N_v}\} \quad (1)$$

The tasks set  $\mathbf{T}$  for  $N_t$  tasks are defined as follows.

$$\mathbf{T} \triangleq \{T_1, T_2, \dots, T_{N_t}\} \quad (2)$$

Now, the performance index and constraints are formulated as

$$\min \sum_{v=1}^{N_v} c^v(\chi^v) \quad (3)$$

subject to

$$\sum_{v=1}^{N_v} x_i^v = 1, \forall T_i \in \mathbf{T} \quad (4)$$

$$\sum_{v=1}^{N_v} \sum_{i=1}^{N_t} x_i^v = N_t \quad (5)$$

$$l_i(\chi^v) \leq b_i, \forall T_i \in \mathbf{T} \quad (6)$$

and

$$x_i^v \in \{0, 1\}, \forall (T_i, V_v) \in \mathbf{T} \times \mathbf{U} \quad (7)$$

where  $c^v$  is the sequential task order of the  $v$ -th UAV with respect to the following binary decision vector  $x^v$

$$x^v \triangleq [x_1^v, x_2^v, \dots, x_{N_t}^v]^T \quad (8)$$

And  $N_t$  is the number of total tasks, and  $l_i$  is an additional function be satisfied by the target  $i$ , such as timing constraints. Complex subscripted problem formulation is used for multiple UAVs and multiple tasks. The performance index  $c^v$  is a function of the sequential task order  $c^v$  of the  $v$ -th UAV. In this study, a cumulative flight time of all the UAVs was taken as the performance index because the flight time is related to the endurance of the UAV and distances between the UAVs and the targets. Eq. (4) signifies that each task should

be assigned at once. Additional inequality conditions can be composed, as given in Eq. (6). These constraints can be adjusted according to the problem.

### 2.3 Path planning with cost prediction

The expenses incurred from each UAV must be predicted in order to assign multiple tasks to multiple UAVs. A path planning algorithm is required when obstacles and restricted regions exist. The cost is then predicted after evaluating the path planning result. Cost can change according to the task order, even if the assigned tasks are same.

In general, a UAV contains kinematic constraints on the turn radius and velocity. Thus, the motion of a UAV is described by the following 2-dimensional kinematic model:

$$\begin{pmatrix} \dot{x} = V \cos \psi \\ \dot{y} = V \sin \psi \\ \dot{\psi} = \omega_{\max} u \end{pmatrix} \quad (9)$$

where the control input is bounded by  $|u| \leq 1$ . Velocity  $V$  is assumed to be a constant, and the minimum turn radius  $R_{\min}$  is:

$$R_{\min} = \frac{V}{\omega_{\max}} \quad (10)$$

For this case, the distance between nodes depends on the heading angle of the UAV. It is also difficult to applying the model described by Eq. (9) to task assignments is difficult because it requires exhausting computational efforts. In order to reduce the complexity, path planning is separated from task assignment. The distance between node  $i$  and node  $j$  is approximated through a Euclidean distance expression and the turn radius  $R_{\min}$ . Therefore, the flight time between node  $i$  and node  $j$  is approximated as follows.

$$t(n_i, n_j) \approx \frac{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{V} \quad (11)$$

$$t(n_i, n_j) \approx \frac{2\pi R_{\min}}{V} \quad (12)$$

where  $n_i$  is the  $i$ -th node,  $n_j$  is the  $j$ -th node, and  $(x_i, y_i)$  and  $(x_j, y_j)$  are the geometric positions of the corresponding nodes, respectively.

A visibility graph and A\* algorithm are then adopted for path planning and cost prediction. The visibility graph comprises a general visibility graph algorithm (Choset, 2005) based on fixed obstacles with safety margins, as shown in Fig. 1. For computational efficiency, cost look-up tables for (i) the target to target and (ii) the UAV to target are constructed and utilized as shown in Fig. 2. The shortest path between

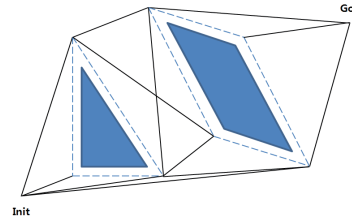


Fig. 1. Visibility graph.

	Target 1	Target 2	...	Target N
Target1	C <sub>11</sub>	C <sub>12</sub>	...	C <sub>1N</sub>
Target2	C <sub>21</sub>	C <sub>22</sub>	...	C <sub>2N</sub>
...	...	...	...	...
Target N	C <sub>N1</sub>	C <sub>N2</sub>	...	C <sub>NN</sub>

	Target 1	Target 2	...	Target N
UAV	C <sub>v1</sub>	C <sub>v2</sub>	...	C <sub>vN</sub>

Fig. 2. Look-up tables of (i) target to target and (ii) unmanned aerial vehicle to target.

```

Procedure PathPlanning( )
If (new target = 1)
    [V, E] = Visibility(targets)
    Cost_table_i = Update_table_i()
    Cost_table_ii= Update_table_ii()
    [Way_set] = A_star(Task_plan)
    
```

Fig. 3. Path planning algorithm.

nodes is required in order to evaluate costs. In this study, we adopted the A\* algorithm, which provides the shortest path that avoids the obstacle. The A\* algorithm is complete and will always find a solution if the given graph structure has the solution. The initial and goal points in Fig. 1 denote the nodes of two targets. Therefore, the A\* algorithm should be performed to all combinations of two targets, and the look-up table as shown in Fig. 2 should be pre-defined. These tables are updated and expanded whenever the costs are changed or new tasks are found. The heuristic term of the A\* algorithm is set as the Euclidean distance from the node to goal node, and convex obstacle regions are only considered when obstacles exist.

Figure 3 shows the path planning function. If a new target is added, table (i) will be updated. Table (ii) is updated whenever the UAV is moving. According to the task plan obtained by the task assignment, the waypoint set of UAV can be obtained. Guidance and control is performed according to the waypoint set.

### 3. Decentralized Task Assignment Using GA

In general, multiple UAVs will perform its mission given the prior task assignment. However, UAVs require on-line task assignment for changing environments because the UAV must adjust its assigned tasks. Possible on-line task assignment schemes include the decentralized task assignment scheme and the distributed task assignment scheme. This study evaluated a fully decentralized task assignment algorithm.

For decentralized task assignments, communication between UAVs should be dealt with. A UAV group requires a specified communication topology; thus, methods for solving the conventional combinatorial optimization problem are not appropriate for the decentralized task assignment. The decentralized task assignment adopts the GA in order to relieve the computational load and to adjust tasks according to the communication between UAVs. GA is a meta-heuristic algorithm that solves the optimization problem. Communication topology is a one-to-one communication based on negotiation for the fully decentralized task assignment. It is also assumed that there is no base station in this study.

Strategy consists of two stages, the order optimization stage and the communications and negotiation stage. Through these two stages, the UAV is able to adjust its task order and exchange the assigned tasks with other UAVs.

#### 3.1 Order optimization stage

Each UAV follows a task order arrangement procedure during the order optimization stage. In this stage, each UAV adjusts the order of its own tasks. This stage is similar to the TSP, and therefore it can be regarded that each UAV solves the TSP at the order optimization stage. The GA is adopted during this stage. Each UAV has its own solution set, and each set experiences improvements due to the GA's genetic operators.

##### 3.1.1 Modified solution set

The solution set is a set of feasible solution candidates known as a chromosome in the GA. The solution set is expressed as  $C$ , and  $C_n$  is  $n$ -th solution candidate in the set, which is similar to  $C$  in Eq. (3). Figure 4 shows the solution candidates, which are composed of the sequential order of the tasks. Each task is represented as  $T_{j, k}$ , where the first subscript  $j \in N^{N^t}$  represents a target and the second subscript  $k \in N^{N^t}$  represents the sub-task of the target.

According to the representations, feasible combination number of the solution for all UAVs is given by Shima et al. (2006).

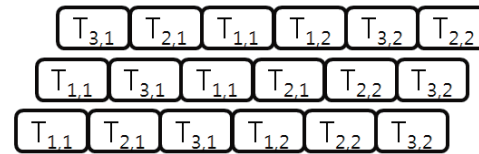


Fig. 4. Solution candidates of the problem.

$$\frac{(N_T N_k)!}{(N_k!)^{N_T}} N_v^{N_T N_k} \quad (13)$$

Each UAV should deal with its own task set at the order optimization stage. For example, if a UAV has  $N_T$  targets and  $N_k$  tasks for each target, the number of combinations will be given by

$$\frac{(N_T N_k)!}{(N_k!)^{N_T}} \quad (14)$$

Therefore, the number of feasible solutions can be reduced by decentralization. However, decentralization could possibly violate the optimality of the task assignment.

Each solution set contains a fitness set with respect to the cost of each candidate. The solution candidate evolves its fitness through the genetic operators. To implement the on-line process, a time extended cost function is defined as follows.

$$c_n(t) \triangleq t + c_p(C_n) \quad (15)$$

where  $t$  is the elapsed time of the UAV,  $C_n$  is the solution of the set, and  $c_p$  is the cost prediction function according to the  $C_n$ .

Then, the fitness of each solution candidate is formulated as follows

$$f_n(t) = (c_w(t) - c_n(t)) + \frac{(c_w(t) - c_b(t))}{(k_s - 1)}, \quad k_s > 1 \quad (16)$$

where  $c_w$  is the cost of the worst case,  $c_b$  is the cost of the best case,  $c_n$  is the cost of  $n$ -th candidate in the set, and  $k_s$  is selection pressure representing the ratio of the best case to the worst case. In this study,  $k_s$  is set as 3.

##### 3.1.2 Genetic operators

The genetic operators consist of selection, crossover, mutation, and substitution operators. To generate a new solution candidate, these four operators are executed.

A proportionate selection with roulette wheel method is adopted as a selection operator. Two solutions are randomly chosen according to the fitness, as shown in Fig. 5. These selected solutions are pruned and attached to each other

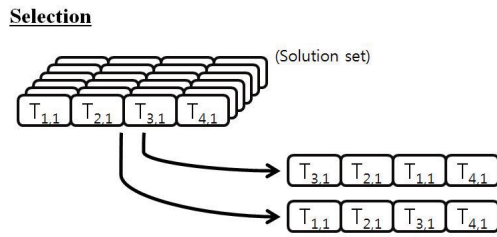


Fig. 5. Selection operator.

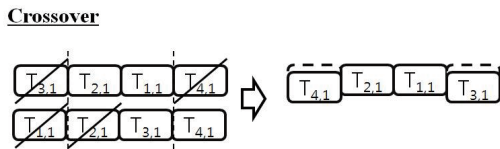


Fig. 6. Crossover operator.

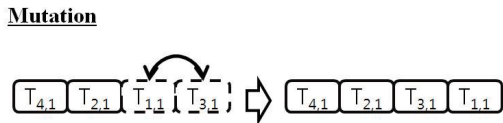


Fig. 7. Mutation operator.

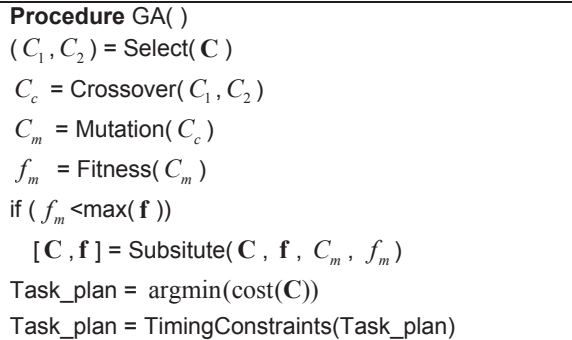


Fig. 8. Genetic algorithm (GA).

by the crossover operator as shown in Fig. 6. Thus, the order crossover is adopted for the array rearrangement of the solution candidate. Mutation is executed by switching the order randomly as shown in Fig. 7. It is used for the local perturbation of the solution. Finally, substitution is performed. Through the recursion of these procedures, the task order would be rearranged. Figure 8 shows the procedures of the GA.

### 3.2 Communications and negotiation stage

The following assumptions were made for the process of

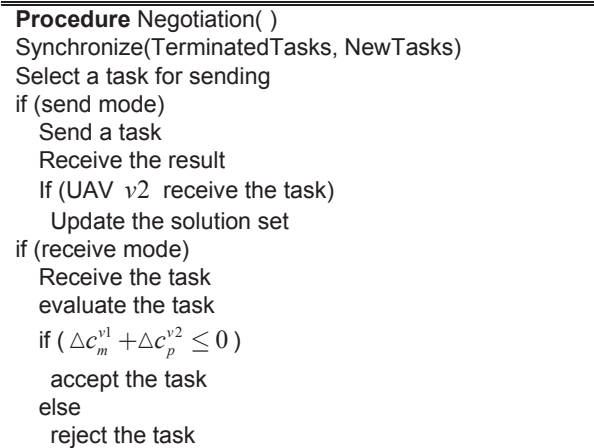


Fig. 9. Communication algorithm. UAV: unmanned aerial vehicle.

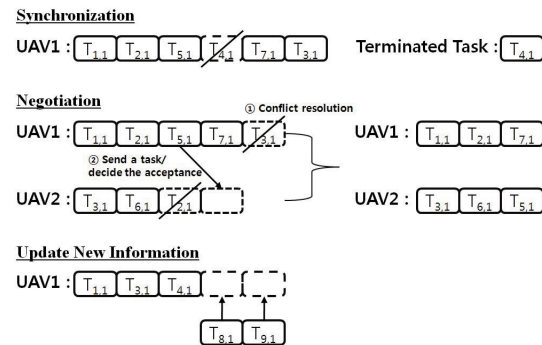


Fig. 10. Communication procedures. UAV: unmanned aerial vehicle.

communication between UAVs.

**Assumption 1.** There are no base stations, and there exists no centralized computers in the UAV groups.

**Assumption 2.** Each communication is restricted to one-to-one communication based on the negotiation.

**Assumption 3.** Tasks are only exchanged between two UAVs based on the communication.

**Assumption 4.** Communication limit due to a range limit is only considered.

The communication strategy is described as follows based on the assumptions above.

1. Synchronize the terminated tasks as well as the additional tasks, and take into account the changes in the task plan.
2. (UAV  $v_1$ ) Choose and send a task to UAV  $v_2$  with the reduced cost  $\Delta c_m^{v_1}$  of UAV  $v_1$ .
3. (UAV  $v_2$ ) Evaluate the received task and compute the increased cost  $\Delta c_p^{v_2}$  of UAV  $v_2$
4. If the sum of the reduced cost of UAV  $v_1$  and the increased cost of UAV  $v_2$  is less than zero, then UAV  $v_2$  accepts the task.

Otherwise, UAV  $v_2$  rejects the task.

5. UAV  $v_2$  executes the same process to UAV  $v_1$ .

Figure 9 shows the procedure of the communication algorithm, and Fig. 10 shows the procedures for GA solutions. In this stage, the terminated tasks and the newly added tasks are required for synchronization. And, the tasks sending reduced cost information are only required.

If all UAVs are connected, tasks will be distributed without duplication. However, rules are required for the situation when the communication is cut off. If UAV detects new targets, the UAV should behave according to the following rules.

**Rule 1.** If the new target information does not exist in the synchronized data, UAV should take all tasks of the new target and update the information.

**Rule 2.** If the new target information exists in the synchronized data, UAV does not include the tasks of the target, because other UAV already found the target and updated the information into the synchronized data.

If the UAV detects new targets when communication is cut off, the UAV should follow the above two rules. However, a situation may occur in which each UAV assumes the same tasks as the new target when communication is cut off. In this case, then UAV should follow the following rules.

**Rule 3.** If a UAV receives the same information of a new target as another UAV, but it does not have the tasks of the new target, the UAV discards the tasks of the new target.

**Rule 4.** If a UAV has the same information of new target as another UAV, and it has the tasks of the target, the UAV will distribute the tasks to another UAV.

And, UAVs should obey the following rule during the negotiation step.

**Rule 5.** When UAVs exchange tasks, the UAV should only send its own tasks to another UAV.

The above rules provide non-conflicting assignments to UAVs in which unbalanced information is available. However, duplication is inevitable when each UAV takes and performs the new tasks without communication. To avoid the duplication, the following condition should be satisfied.

**Necessary condition.** New target information should be transferred and synchronized to all UAVs until another UAV performs the task of the new target.

According to the above condition, if a UAV recovers the communication, it can perform the mission without the duplicating tasks. Therefore, synchronization is the most important step for autonomous UAVs. To synchronize the unbalanced information, at least  $v_{C_2}$  times communications are required.

Through the order optimization stage and the

---

```

Procedure Main( )
C = Initialize the feasible solution set
f = Fitness( C )
while(operation)
    Receive the result of GA
    Receive the result of Path Planning
    Guidance and Control
    if (  $T_{j,k}$  is done )
        Transfer  $T_{j,k}$  from the set C to the terminated
        tasks
    if (communication =1)
        Receive the result of Negotiation
        Modify the solution set
    
```

---

Fig. 11. Main loop of the task assignment, path planning, and communication of unmanned aerial vehicle. GA: genetic algorithm.

communication and negotiation stage, the decentralized task assignment can be realized. Note that each UAV should be capable of mission planning and communicating for autonomous operation.

Figure 11 shows the main process loop of each UAV. Task assignment, path planning, and communication procedures can be divided into several modularized functions.

### 3.3 Timing constraints operators

The solution's task order may be rearranged after the communication and optimization stages. However, this rearrangement may violate timing constraints, such as the individual order of each target. Therefore, internal constraints check logic is required for GA, because conventional crossover and mutation cannot reflect these conditions. In this study, additional operators were incorporated into the study as the timing constraints. The operation time of the  $i$ -th task ( $i \in N^{N \times N^k}$ ) can be bounded as the upper bound and lower bound as

$$t_{i, LB} \leq t_i \leq t_{i, UB} \quad (17)$$

where  $t_{i, LB}$  is the lower bound of the  $i$ -th task, and  $t_{i, UB}$  is the upper bound of the  $i$ -th task. Note that  $t_{i, LB}$  and  $t_{i, UB}$  are decreased with time.

Subsequently, modified mutation operators, which comprise the forward mutation and the backward mutation, are composed with respect to the constraints, as shown in Figs. 12 and 13. The forward mutation is related to the upper bound. If the operation time exceeds the upper bound, the task is shifted forward. Similarly, if the operation time is less than the lower bound, then the task is shifted backward. Each UAV should have the upper bound and the lower bound

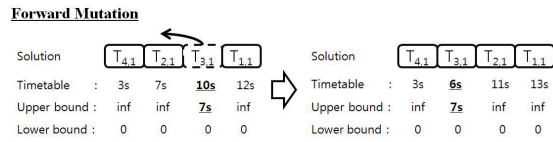


Fig. 12. Forward mutation operator.

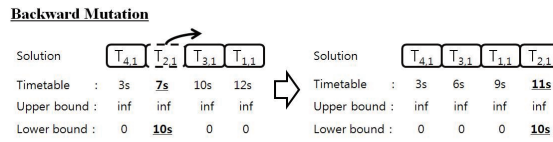


Fig. 13. Backward mutation operator.

```

Procedure TimingConstraints( )
while(terminate=0)
  for i=1:length(task plan)
    if all  $t_i \leq t_{i,UB}$  and  $t_i \geq t_{i,UB}$ 
      Terminate = 1
    for i=1:length(task plan)
      if  $\tau_i > t_{i,UB}$ 
        Forward mutation()
      if  $\tau_i < t_{i,LB}$ 
        Backward mutation()

```

Fig. 14. Algorithm of the timing constraints.

information of each task. Initial upper bound values are set as infinity, and the initial lower bound values are set as zero. For the timing constraint operators, the timetables for the solution set are required.

Figure 14 shows the algorithm of the timing constraints operators.  $\tau_i$  is the timetable value of the  $i$ -th task.

All of the additional constraints are dealt with in the GA loop, and therefore a non-conflicting solution can be obtained. At the negotiation stage, each UAV requires the time bound information of other UAVs, and the

additional information about the bounds is required for communication.

## 4. Numerical Simulations

### 4.1 Simulation conditions

Numerical simulations were performed for the multiple UAVs, multiple tasks which consist of multiple targets and their sub-tasks. Table 1 summarizes the conditions of the numerical simulations. Standard setting includes 3 UAVs, 6 known targets, and 2 sub-tasks for each target, and therefore the UAV group should perform 12 tasks under the timing constraints. Case 1 is the centralized task assignment case using MILP (Schumacher et al., 2004), which provided a solution for comparison to the other cases. Cases 2-6 are the decentralized task assignment cases using GA and the negotiation process, which were evaluated for on-line applications. The simulations were performed on a desktop computer which has a dual-core 2.53 GHz CPU and 2 GB RAM, and MATLAB software. The process time for each function was tuned by adjusting the number of generations in the GA. The process times of the GA, communication, and path planning were 0.03, 0.05, and 0.1 seconds, respectively. The probability of the mutation was set to 0.8, and the number of the solution candidates was set to 10 in the GA. The number of generations per each step was set to 15.

Unknown targets were included in simulations concerning the dynamic environment. If a UAV approached the unknown targets, the UAV detected the unknown targets, and the information will be updated. The sensing range was set to 800 meters. Additional timing constraints were set to the deadlines of the specified targets independent of the task order of each target.

### 4.2 The centralized task assignment

Case 1 is the centralized task assignment case. The result of

Table 1. Conditions of the numerical simulations

Case 1 (MILP)	Case 2 (GA)	Case 3 (GA)	Case 4 (GA)	Case 5 (GA)	Case 6 (GA)
3 UAVs	3 UAVs	3 UAVs	3 UAVs	3 UAVs	3 UAVs
(homogeneous, $V = 25$ m/s, $R_{min} = 100$ m)	(homogeneous, $V = 25$ m/s, $R_{min} = 100$ m)	(homogeneous, $V = 25$ m/s, $R_{min} = 100$ m)	(homogeneous, $V = 25$ m/s, $R_{min} = 100$ m)	(heterogeneous, $V = \{25, 15, 35\}$ m/s, $R_{min} = \{75, 50, 100\}$ m)	(homogeneous, $V = 25$ m/s, $R_{min} = 100$ m)
6 known targets	6 known targets	6 known targets	6 known targets	6 known targets	6 known targets
2 tasks per target	2 tasks per target	3 unknown targets (sensing range 800 m) 2 tasks per target	3 unknown targets (sensing range 800 m) 2 tasks per target	3 unknown targets (sensing range 800 m) 2 tasks per target	3 unknown targets (sensing range 800 m) 2 tasks per target
			Additional timing constraints	Additional timing constraints	Communication range limit 500 m

MILP: mixed integer linear programming, GA: genetic algorithm, UAV: unmanned aerial vehicle.

Table 2. Result of Case 1–tasks with time

UAV 1	(Target, Task)	(1,1)-(2,1)-(6,1)-(4,2)
	Time (s)	27.2-62.8-99-120
UAV 2	(Target, Task)	(3,1)-(5,1)-(5,2)-(4,1)-(6,2)
	Time (s)	23.2-51.6-77-118-140
UAV 3	(Target, Task)	(3,2)-(1,2)-(2,2)
	Time (s)	26.4-49-85.8

UAV: unmanned aerial vehicle.

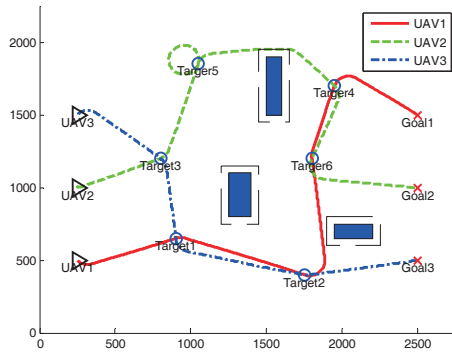


Fig. 15. Case 1 results–the centralized task assignment using mixed integer linear programming.

Table 3. Result of Case 2–tasks with time

UAV 1	(Target, Task)	(1,1)-(3,2)-(5,2)-(6,2)-(4,2)
	Time (s)	27.2-51.8-80-125-150.4
UAV 2	(Target, Task)	(3,1)-(5,1)-(4,1)-(6,1)
	Time (s)	23.4-51.6-93-114.8
UAV 3	(Target, Task)	(1,2)-(2,1)-(2,2)
	Time (s)	46-81.8-107

UAV: unmanned aerial vehicle.

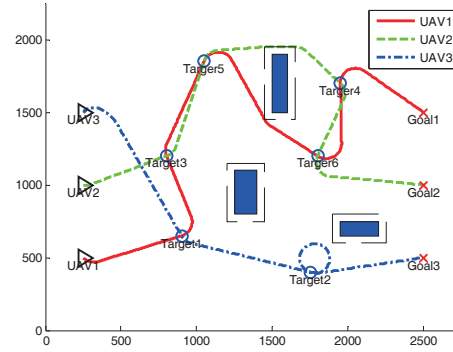


Fig. 18. Case 2 results–the decentralized task assignment using genetic algorithm and the negotiation.

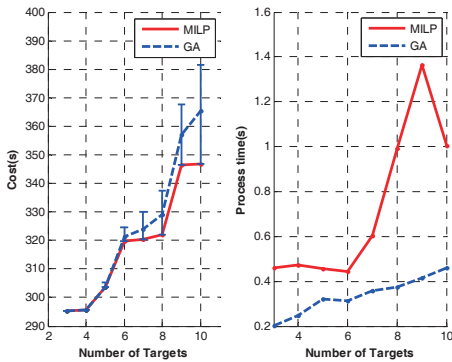


Fig. 16. Costs and process times with respect to the number of target contains 1 sub-task.

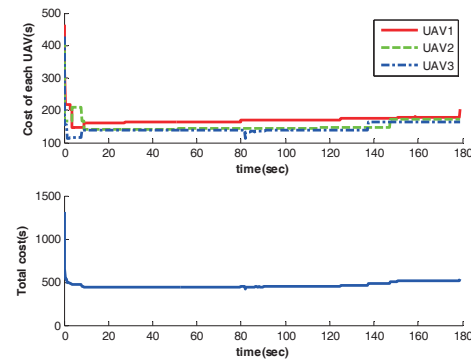


Fig. 19. Cost of each unmanned aerial vehicle and total cost with respect to the time–Case 2.

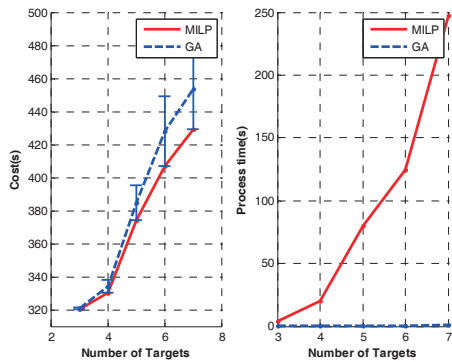


Fig. 17. Costs and process times with respect to the number of target contains 2 sub-tasks.

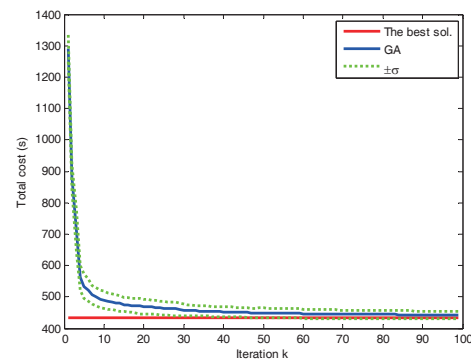


Fig. 20. Mean cost variation of Monte Carlo runs (n = 100).



the assignment is shown in Fig. 15. This result is the optimal assignment solution under linear cost approximation. Table 2 summarizes the assignment results with respect to the time.

Timing constraints cause difficulties during problem solving. Figure 16 shows the cost and process time results of the centralized task assignment using MILP and GA when each target had one sub-task. In this figure, the process times of MILP and GA are relatively short. Figure 17 is the case of two sub-tasks for each target. In this case, the timing constraints were required, and the constraints sizes were expanded. As shown in Fig. 17, the process time of MILP remarkably increased.

### 4.3 The decentralized task assignment

Cases 2-6 are the decentralized task assignment cases using GA and communication. Figure 18 shows the result of the decentralized task assignment of Case 2 and it is summarized in Table 3. Figure 19 shows the cost variation with respect to the time. The costs between UAVs were balanced during the early stages. Figure 20 shows the results obtained from the Monte Carlo simulation. The task assignment simulations were performed 100 times, and the mean cost and standard deviation with respect to the iteration numbers were plotted. Achieving a converging solution is difficult. However, the total cost approached the minimum cost.

Figure 21 shows Case 3 results, and the assigned tasks with respect to time are summarized in Table 4. This case includes the dynamic environments. In this Case, 3 unknown targets were incorporated into the dynamic environment. In the Fig. 21, the triangles denote the unknown targets. During the early stages, each UAV does not know the unknown targets. When the UAV approached the unknown targets, it detected the unknown objects and updated the targets information.

Case 4 is the case with the timing constraints. The results are shown in Fig. 22. The task order of the same target can be locally adjusted for each UAV. However, if the UAVs exchanged tasks with each other, the task order may become disorganized. Modified mutation was used in order to prevent disorganization. In this case, additional constraints, such as the internal task order, were configured. For target 1, sub-task 1 was set to execute between 30 seconds and 100 seconds, and sub-task 2 was set to execute after 100 seconds. Table 5 shows that the constraints were satisfied.

Case 5 involves heterogeneous UAVs. UAV 1 velocity was assumed to be 25 m/s, UAV 2 was 15 m/s, and UAV 3 was 35 m/s. The turn radius of UAV 1 was 75 m, UAV 2 was 50 m, and UAV 3 was 100 m. The additional timing constraint was set to the same value as that in Case 3. Because of the different velocities and turn radii, the flight time of each UAV was also

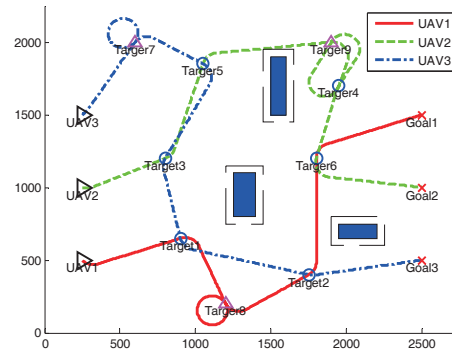


Fig. 21. Case 3 results—the decentralized task assignment using genetic algorithm and the negotiation with new targets condition.

Table 4. Result of Case 3—tasks with time

UAV 1	(Target, Task)	(1,1)-(8,1)-(8,2)-(2,1)-(6,1)
	Time (s)	27.2-50.6-75.6-101-133.8
UAV 2	(Target, Task)	(3,1)-(5,1)-(9,1)-(4,1)-(9,2)-(4,2)-(6,2)
	Time (s)	23.6-51.6-87.2-102.4-126.6-148.8-169.6
UAV 3	(Target, Task)	(7,1)-(7,2)-(5,2)-(3,2)-(1,2)-(2,2)
	Time (s)	24.2-49.4-69.8-99.8-122.4-158.8

UAV: unmanned aerial vehicle.

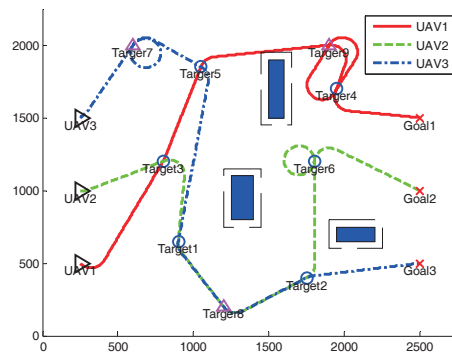


Fig. 22. Case 4 results—the decentralized task assignment using genetic algorithm and the negotiation with new targets and additional timing constraints conditions

Table 5. Result of Case 4—tasks with time

UAV 1	(Target, Task)	(3,2)-(5,1)-(9,1)-(4,1)-(9,2)-(4,2)
	Time (s)	38.6-66.6-101.8-117-141.2-163.2
UAV 2	(Target, Task)	(3,1)-(1,1)-(8,1)-(2,1)-(6,1)-(6,2)
	Time (s)	23.4-49.8-71.6-96.6-129.4-154.6
UAV 3	(Target, Task)	(7,1)-(7,2)-(5,2)-(1,2)-(8,2)-(2,2)
	Time (s)	24.2-49.4-70.2-119.6-141.6-166.6

UAV: unmanned aerial vehicle.

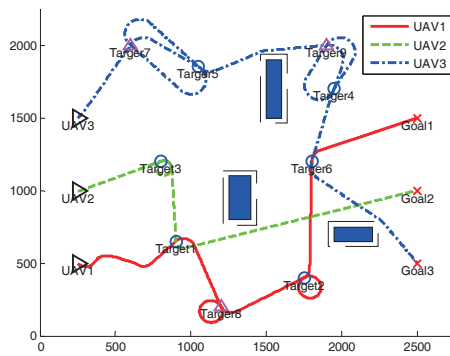


Fig. 23. Case 5 results—the decentralized task assignment using genetic algorithm and the negotiation with new targets, additional timing constraints, and heterogeneous unmanned aerial vehicles conditions.

Table 6. Result of Case 5—tasks with time

UAV 1	(Target, Task)	(1,1)-(8,1)-(8,2)-(2,1)-(2,2)-(6,1)
	Time (s)	30.2-54.6-73.4-98.4-117.2-149.8
UAV 2	(Target, Task)	(3,1)-(3,2)-(1,2)
	Time (s)	39-59.8-100.4
UAV 3	(Target, Task)	(7,1)-(5,1)-(7,2)-(5,2)-(9,1)-(4,1)-(9,2)-(4,2)-(6,2)
	Time (s)	17.2-32.2-55-76.6-102-112.8-130.2-146-161

UAV: unmanned aerial vehicle.

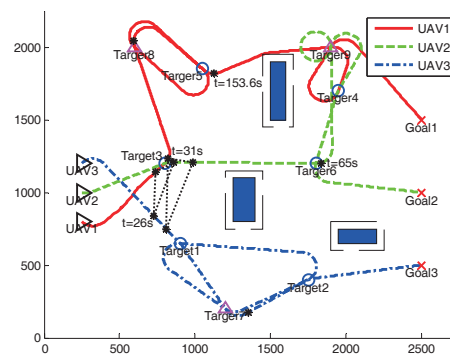


Fig. 24. Case 6 results—the decentralized task assignment using genetic algorithm and the negotiation with new targets, additional timing constraints, and communication limit conditions.

different. Each UAV adjusted the timing constraints as shown in Fig. 23 and Table 6.

Case 6 involved the simulation with respect to the communication range limit. The limit was set to 500 m. In Case 6, repeated tasks were experienced, as shown in Table 7. Figure 24 shows that known tasks were distributed in the early stages, but unknown tasks were not distributed. As

Table 7. Result of Case 6—tasks with time

UAV 1	(Target, Task)	(3,2)-(8,1)-(5,1)-(8,2)-(9,1)-(4,2)-(9,2)
	Time (s)	22.4-63-89.4-120-150-185.6-200.8
UAV 2	(Target, Task)	(3,1)-(6,1)-(9,1)-(9,2)-(4,1)-(6,2)
	Time (s)	23.4-63.4-98-123.2-148.8-169.6
UAV 3	(Target, Task)	(1,1)-(7,1)-(2,1)-(1,2)-(7,2)-(2,2)
	Time (s)	36.2-57.8-83-125.8-154.6-178.6

UAV: unmanned aerial vehicle.

Table 8. Cost and process time of each case

	Total cost (s)	Cost of UAV 1 (s)	Cost of UAV 2 (s)	Cost of UAV 3 (s)	Process time (s/step)
Case 1	435.2	146.4	172.6	116.2	130
Case 2	464.2	179	147.6	137.6	0.07
Case 3	556.4	165.6	201.6	189.2	0.08
Case 4	575.2	190.6	187.6	197	0.08
Case 5	584	181.2	212.4	190.4	0.08
Case 6	670.4	259.6	201.6	209.2	0.08

UAV: unmanned aerial vehicle.

UAVs performed the mission, each UAV recognized new tasks or received synchronized information from the other UAVs.

In Fig. 24, the small dotted points denote the position at which the UAVs identified a new target, and the dotted lines denote the communication connection. Therefore, information of target 7 was updated when UAV3 moved to target 1, which was located nearby target 7 ( $t = 26$  s). Information from target 7 was transferred to UAV 1 and UAV 2. Therefore UAV 1 performed the tasks of the new target as target 8, which was located at the top left-hand side. UAV 1 located target 8 when it moved from target 3 to target 5 ( $t = 31$  s), the sensing point was located on a slightly upward location from the target 3, and therefore UAV 1 promptly changed its task order to perform the tasks of target 8. While UAV 1 performed the tasks of target 8, UAV 2 moved to the new target and performed the new target as target 9 ( $t = 65$  s). During that time, communication of UAV 1 and UAV 2 was disconnected.

Therefore, UAV 1 performed the tasks provided by target 9, and treated it as a new target ( $t = 153.6$  s). This phenomenon is natural under the limited communication range condition, and it can be resolved by adopting the large range communication equipments.

Table 8 summarizes the cost and process time of each case. Case 1 is the off-line assignment case, whereas Cases 2-6 were on-line assignment cases considering the process time and the communications. In comparison to Case 1,

Case 2 incurred more costs in order to perform the tasks. However, the process time obtaining the solution was shorter than that of Case 1. Because of the timing constraints, the process time of MILP method increased. Therefore, GA was suitable to the sequential process. Moreover, the assignment result can be improved with respect to the time. Cases 3-6 considered the additional conditions such as unknown targets, additional timing constraints, heterogeneous model, and communication range limit. As shown in the figures and tables, task assignments were properly performed to the given conditions. Though the cost increased according to the conditions, the process time did not change.

## 5. Conclusions

In this study, a decentralized task assignment algorithm was evaluated as a means for implementing autonomous multiple UAV operations. The results are described as follows.

First, a modified algorithm for applying on-line task assignments to multiple UAVs was proposed. The multiple targets containing multiple tasks scenarios were dealt with for the problem. The considered scenarios have more computational complexity rather than the single waypoint assignment scenario. A genetic algorithm was adopted in order to facilitate problem solving. It is implemented sequentially by adjusting the recursion steps for the on-line application.

Second, the fully decentralized task assignment process was suggested. For the decentralization, the following assumptions were made: a base station did not exist, and each had autonomous function for the mission planning. The problem size can be reduced by the decentralization compared with the centralized task assignment.

Third, the communication and task distribution strategy were dealt with for the dynamic environments and the communication limit. Each UAV only distributed its own tasks through communication, and each UAV performed tasks according to the specified strategy. The proposed method provided non-conflicting assignments communication was limited and new target information was available.

Although the solution is not optimal, it still serves as a feasible solution guaranteeing good performance. The simulation results validate the proposed approach. The proposed method is appropriate for moderate sized problems. However, process time increases are inevitable as the problem size increases. In this case, hard constraints, such as the fully decentralized process and the communication limit, must be relieved. In other words, if the problem size

increases, partially decentralized process having the base station and guaranteeing the communication ranges are required.

## Acknowledgments

This work is supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 20100018871).

## References

- Alighanbari, M. and How, J. P. (2005). Decentralized task assignment for unmanned aerial vehicles. *44th IEEE Conference on Decision and Control and the European Control Conference*, Seville, Spain. pp. 5668-5673.
- Chandler, P. R., Pachter, M., Rasmussen, S., and Schumacher, C. (2002). Multiple task assignment for a UAV team. *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Monterey, CA.
- Choset, H. M. (2005). *Principles of Robot Motion: Theory, Algorithms, and Implementation*. Cambridge, MA: MIT Press.
- Cruz, J. B., Chen, G., Li, D., and Wang, X. (2004). Particle swarm optimization for resource allocation in UAV cooperative control. *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Rhode Island.
- Eun, Y. and Bang, H. (2009). Cooperative task assignment/path planning of multiple unmanned aerial vehicles using genetic algorithms. *Journal of Aircraft*, 46, 338-343.
- Murray, R. M. (2007). Recent research in cooperative control of multivehicle systems. *Journal of Dynamic Systems, Measurement, and Control*, 129, 571-583.
- Papadimitriou, C. H. and Steiglitz, K. (1982). *Combinatorial Optimization: Algorithms and Complexity*. Englewood Cliffs, NJ: Prentice Hall.
- Potvin, J.-Y. (1996). Genetic algorithms for the traveling salesman problem. *Annals of Operations Research*, 63, 337-370.
- Richards, A., Bellingham, J., Tillerson, M., and How, J. (2002). Coordination and control of multiple UAVs. *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Monterey, CA.
- Schumacher, C., Chandler, P., Pachter, M., and Pachter, L. (2004). Constrained optimization for UAV task assignment. *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Rhode Island.

Shima, T., Rasmussen, S. J., Sparks, A. G., and Passino, K. M. (2006). Multiple task assignments for cooperating uninhabited aerial vehicles using genetic algorithms. *Computers & Operations Research*, 33, 3252-3269.

Sujit, P., Sinha, A., and Ghose, D. (2007). Team, game,

and negotiation based intelligent autonomous UAV task allocation for wide area applications. In J. Chahl, L. Jain, A. Mizutani, and M. Sato-Ilic, eds. *Innovations in Intelligent Machines 1 Studies in Computational Intelligence Vol 70*. Heidelberg: Springer Berlin. pp. 39-75.