Genshe Chen and Jose B. Cruz, Jr., "Genetic Algorithm for Task Allocation in UAV Cooperative Control," *AIAA Conference on Guidance, Navigation, and Control,* CD ROM, August 2003.

Genetic Algorithm for Task Allocation in UAV Cooperative Control

Genshe Chen and Jose B. Cruz, Jr

Department of Electrical Engineering

The Ohio State University

Columbus, OH 43210

Abstract

Task allocation is one of the core steps to effectively exploit the capabilities of cooperative control of multiple Uninhabited Aerospace Vehicle (UAV) teams. Task allocation is an NP-complete problem. In this paper, we present a new task allocation algorithm that is based on the principles of genetic algorithm (GA). GA is a class of adaptive search algorithms derived from biological population genetics. We discuss the adaptation and implementation of the GA search strategy to the task allocation problem in the cooperative control of multiple UAVs. Simulation results indicate that the GA strategy is a feasible approach for the task allocation problem.

1. Introduction

Recently military conflicts have demonstrated the strategic value of unmanned air vehicles (UAVs). The roles of UAV are evolving from reconnaissance purpose to offensive mission as missile-launching platform. The capabilities of UAV will be further improved if multiple UAVs are cooperative. Achievement of cooperation among UAVs requires a method of assigning tasks (searching for (classify), attacking and performing battle damage assessment (BDA) on potential targets) to vehicles that all targets are prosecuted in an optimal manner, and that ensures that constraints are met. This is the task allocation problem. There are two types of task allocation problems: static and dynamic. Static task allocation means that the assignment may be made at time t such that all of the UAVs are committed, while dynamic task allocation is made at any of several discrete points of time.

There is a wide variety of approaches that have been reported for solving the task allocation problem in various applications [1]. Roughly, they can be classified into the following categories: network flow optimization [2], market based approach [3], integer linear programming [4], and fuzzy approach [5]. Because of the intractable nature of the task allocation problem and its importance in cooperative control, it is desirable to explore other avenues for developing good heuristic algorithm for the problem. In this paper, we introduce an approach based on genetic algorithm for the static task allocation problem.

The genetic algorithm (GA) is a stochastic search algorithm that models the process of nature selection and genetics [6]. It is an iterative algorithm that maintains a pool of feasible solutions, for each iteration. The GA starts with a set of randomly selected chromosomes as the initial population that encodes a set of possible solutions. Variables of a problem are represented as genes in a chromosome, and chromosomes are evaluated according to their fitness values, which are obtained by evaluating the considered fitness or cost function. Recombination typically involves two operators: (1) crossover and (2) mutation. Genetic operators alter the composition of genes to create new chromosomes referred to as offspring. The selection operator is an artificial version of nature selection, a Darwinian survival of the fittest among populations, to create populations from generation to generation. Chromosomes with better fitness have higher probabilities of being selected in the next generation. After several generations, GA can converge to the best solution. GA has many advantages over other heuristic techniques. For example, GA can be implemented in a few lines of computer code, it requires only primitive mathematical operators, and it has high probability to escape local optima.

The remainder of this paper is organized as follows. Section 2 provides the mathematical statement of the task allocation problem. Section 3 describes our genetic algorithm for the task allocation problem. Simulation results are reported in section 4. Finally, section 5 concludes the paper.

2. Problem Formation

This task allocation problem is to calculate the optimal UAV allocation subject to resource constraints. The objective function is based on the engagement rules and tactics. It could be either a single objective function or multiple objective functions. For example, an engagement objective could be to determine a UAV allocation that can engage the maximum target value with the highest weighted options. The engagement rules and weapon systems govern other features and constraints of the problem, such as allocation strategies (shoot-look-shoot, salvo attack). In this paper we only consider the static allocation problem. The problem and the constraints are described below.

We assume that team composition has already been performed, and that a set of tasks has been identified which must be performed by the team [7]. Consider that there are M UAVs and N independent targets. Define the decision variable x_{ij} , $i = 1, \cdots, M$, and $j = 1, \cdots, N$,

$$x_{ij} = \begin{cases} 1 : UAV\ i\ \ assigned\ \ to\ target\ \ j \\ 0 : else \end{cases} \quad (1)$$

Our objective is to destroy all targets, while minimizing the risk to each individual UAV. Risk is mitigated by minimizing the UAV flight time to target and by simultaneously attacking targets with multiple UAVs (decoys), such that preference is given to high priority targets.

The task allocation problem can be formulated as

$$\min \sum_{i,j} c_{ij} x_{ij} \quad (2)$$

$$S.t. \quad \sum_{j=1}^{M} x_{ij} = 1,\ for\ i = 1, 2, \cdots, N : \text{All targets must be assigned to one } UAV$$
$$x_{ij} \subset \{0,1\}\ for\ all\ \ i = 1, 2, \cdots, N,\ j = 1, 2, \cdots, M \quad (3)$$

If we assume the number of UAV and the number of targets is the same, M=N, then the constraints become

$$\sum_{j=1}^{N} x_{ij} = 1,\ for\ i = 1,2,\cdots,N$$
$$\sum_{i=1}^{N} x_{ij} = 1,\ for\ j = 1,2,\cdots,N \quad (4)$$

One of the critical questions in UAV-target assignment problem is how to choose the value of weights C_{ij}. Different expressions will yield different situations. For attacking, the main objective is to assign the highest value possible to kill a target of the highest-valued type, with other tasks (search for, BDA) generating less benefits. The value for attacking is calculated as follows

$$C_{ij} = P_{id} * (1 - P_{kij}) * V_i * \min_i (TMatrix) / TMatrix(i, j) \quad (5)$$

where P_{id} is target identification certainty, $1 - P_{kij}$ is the probability of target i not damaged by UAV j, V_i is the value of target i, and TMatrix contains the required flight times for every UAV j to fly to every target i.

Equation (2) can be further extended to weapon level as follows

$$\min_{x_{ij}} J = \sum_{i=1}^{N} P_{id} * V_i * \min_i (TMatrix) / TMatrix(i, j) \sum_{j=1}^{M} (1 - P_{kij}) x_{ij} \quad (6)$$

A naïve solution to the above problem would be to find the optimal x_{ij} such that objective function (2) is optimized subject to constraints (3) or (4). However, this will require solving a problem with a large search space. For example, for one target and M UAVs, where each UAV is a potential assignment to attack this target, the number of possible target UAV pairings is M . For N targets and M UAVs, number of possible pairing is M^N . Thus, as the number of target increases, the number of such combinations or possible target-UAV pairing increases exponentially.

3. Genetic Algorithm for Task Allocation Problem

In this paper, we propose a GA based heuristic method to solve the task allocation problem. GA is a search method that heuristically "mimics" biological evolution. The concept of a genetic algorithm was first conceived by John Holland of the University of Michigan, Ann Arbor. Over the last decade, GA has been extensively used as search and optimization tools in various problem domains, including the sciences, commerce and engineering. The primary reasons for their success are their broad applicability, ease of use and global perspective [6]. In this section, we describe the formulation of a GA for the task allocation problem. The chromosomes and genetic operators used for our task allocation problem are briefly introduced in the following.

The encoding of task allocation solutions is straightforward. The permutation π is coded as a vector of targets and the value of the component indicates to which target the UAV is assigned. Take UAV=4 and Target=4 as an example. The chromosome $(4\ 1\ 3\ 2)$ represents an assignment list where UAV 1 is assigned to target 4, UAV 2 is assigned to target 1, UAV 3 is assigned to target 3, and UAV 4 is assigned to target 2. It is noted that there are exact M genes (component) in chromosome (individual) and their corresponding values are integers between 1 and N.

One-cut-point crossover, inversion mutation, and roulette selection operators are employed in GA. The crossover operator randomly generates a cut position and swaps the cut parts of two parents so as to generate offspring. Mutation operations randomly change a gene in the chromosome M_n times, where $M_n < M$ is randomly generated. After off-springs are generated, chromosomes need to be chosen for the next generation population. A set of chromosomes $P(t+1)$ is selected from the pool of parents and offspring to reduce the population to its original size. After those new chromosomes are selected, the process is repeated until a stop criterion is reached.

The main steps in our task allocation algorithm are shown in Figure 1.

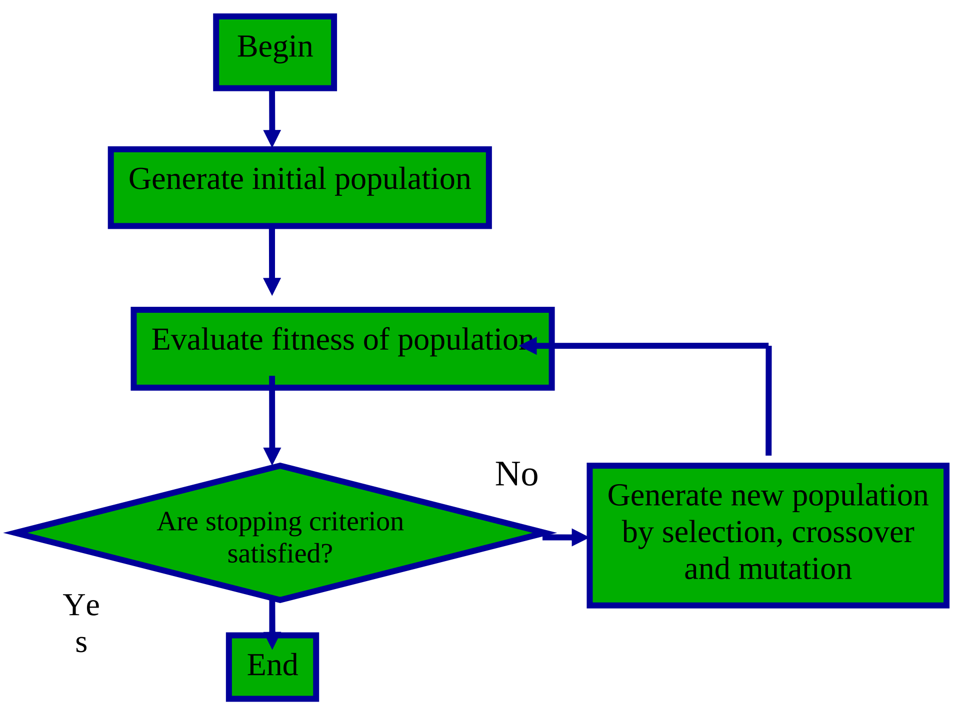

Figure1. GA algorithm for task allocation problem

4. Simulation

In order to demonstrate the performance of this algorithm, simulation was performed. For this purpose a Matlab computer simulation program was developed. The tests were based on the following scenario: four UAV attack four different target types in a rectangular area. In our allocation problem, the following assumptions were made: (1) The individual probability of killing P_{kij} by assigning the j^{th} UAV to destroy the i^{th} target is known in advance (see Table 1); (2) The value of target i is V_i, which is also known in advance and determines the preference of which target to attack (see Table 1); (3) All UAVs in one team are not far away each other. The same situation holds for targets. In this case, we do not need to consider the flight time factor. (4) Target identification certainty is 1.

Table 1 Target value and kill probability

	$P_k(U_1)$	$P_k(U_2)$	$P_k(U_3)$	$P_k(U_4)$	V_i
T_1	0.1	0.8	0.3	0.4	0.9
T_2	0.1	0.2	0.4	0.9	0.9
T_3	0.2	0.2	0.8	0.3	0.9
T_4	0.9	0.1	0.2	0.3	0.9

The parameters of GA are: (1) Crossover probability =0.9. (2) Mutation probability =0.4. (3) The size of the initial population for GA =20.

Table 2 shows the simulation results and Figure 2 illustrates the change of fitness with generation. Table 2 shows that the best assignment in the first generation is (4 3 3 2). Obviously this is not the solution because UAV 2 and 3 are assigned to the same target 3, while there is no UAV assigned to target 1. The algorithm converged with the fitness value as 3.06 after the fifth generation, and the best allocation is: UAV 1 is assigned to target 4, UAV 2 is assigned to target 1, UAV 3 is assigned to target 3, and UAV 4 is assigned to target 2. This result is very reasonable. This example demonstrates that the GA approach results in a satisfactory solution for task allocation problem.

Table 2 The simulation result

Generation 1: f(4, 3, 3, 2)=2.520000

Generation 2: f(4, 3, 3, 2)=2.520000

Generation 3: f(4, 1, 1, 2)=2.610000

Generation 4: f(4, 1, 1, 2)=2.610000

Generation 5: f(4, 1, 3, 2)=3.060000

Generation 6: f(4, 1, 3, 2)=3.060000

Generation 7: f(4, 1, 3, 2)=3.060000

Generation 8: f(4, 1, 3, 2)=3.060000

Generation 9: f(4, 1, 3, 2)=3.060000

Generation 10: f(4, 1, 3, 2)=3.060000

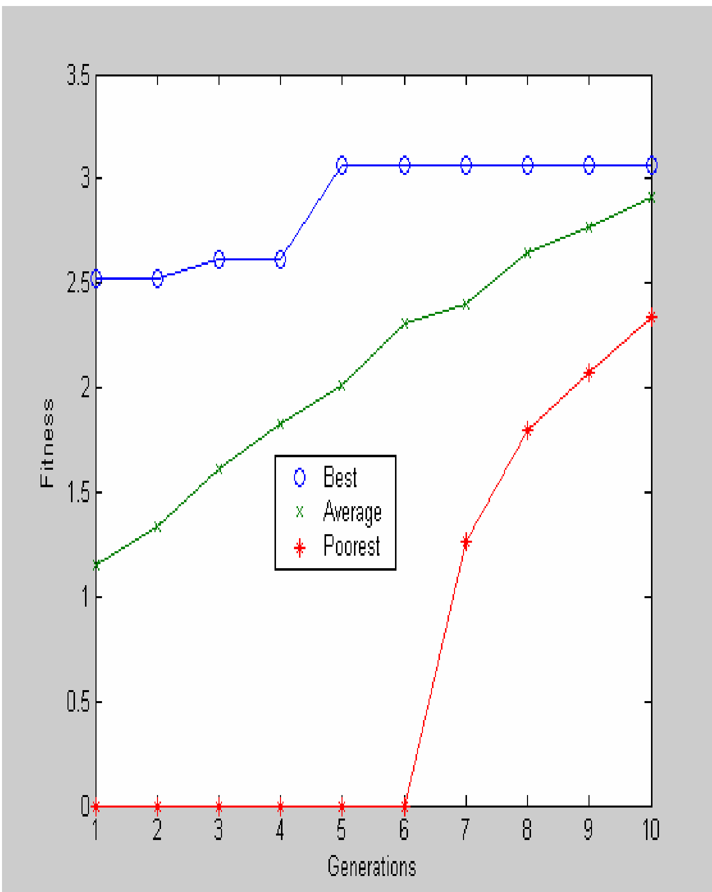

Fig. 2. The change of fitness with generation

5. Conclusion

In UAV cooperative control, qualified allocation of tasks among UAVs is an important step for efficient utilization of resources. In this paper, a genetic based algorithm was presented for the problem of task allocation for teams of UAV. The results showed that the GA algorithm solution quality is satisfied. These results indicated that the proposed GA is an attractive alternative for solving the task allocation problem in UAV cooperative control. There are several possible extensions to this work: (1) Numerous simple improvements may be made to the current algorithm to improve its overall performance: using domain specific knowledge in the genetic operator, initial population generation and including some local search. (2) improvements are needed to make the cost function yield other tasks (search, BDA), and other constraints, such as timing constraints, e.g. a target can not receive a BDA before it is attacked, nor can a target be attacked before it is searched (classified). (3) Another possible extension is to study dynamic task allocation.

Acknowledgment

This research was sponsored by the Defense Advanced Research Project Agency (DARPA) under Contract F33615-01-C3151 issued by the AFRL/VAK. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the DARPA or AFRL.

References

[1] Ibarraki T. and Katoh N., Resource Allocation problem: algorithmic approaches. Cambridge, Mass, MIT Press, 1988.

[2] Chandler, P. R., Pachter, M., Nygard, K.E. and Swaroop, D. “Cooperative Control for Target Classification,” Cooperative Control and Optimization (R. Murphey and P. M. Pardalos editor), Kluwer Academic Publishers, 2002.

[3] Tierno, J. E., “Distributed autonomous control of concurrent combat tasks,” Proceeding of the American Control Conference, Arlington, VA, June 25-27, 2001, PP37-42.

[4] Griggs B. J., Parnell G. S., and Lehmkuhl, “An Air Mission Planning Algorithm Using Decision Analysis and Mixed Integer Programming, ” Operations Research, 45(5), 1997, PP662-676.

[5] Garagic, D. and Cruz, J. B., “Target Allocation Using a Binary Integer Programming with Fuzzy Objective,” OSU Workshop for DARPA MICA Program, November 08, 2002

[6] Goldberg, D.E., Genetic algorithm in search optimization and machine learning, Addison Wesley, New York, 1988.

[7] Heise, S. A., Mixed Initiative Control of Automa-teams (MICA), http://dtsn.darpa.mil/ixo/mica.asp