Application of a Genetic Algorithm to improve an existing solution for the General Assignment Problem.

Paul Juell Amal S. Perera Kendall E. Nygard
Department of Computer Science
North Dakota State University
Fargo, ND 58105, USA
{paul.juell, amal.perera, kendall.nygard}@ndsu.nodak.edu

Abstract

In this paper we present a mechanism to improve the solution quality of an existing heuristic based general assignment problem solver by adjusting the heuristic. The heuristic is tweaked using a set of parameters suggested by a genetic algorithm. The genetic algorithm is applied in a way that reduces the amount of involvement required to understand the existing solution. To illustrate the proposed approach we present the design implementation and simulation results for an application. This application addresses the problem of target allocation for a team of unmanned air vehicles that results in an efficient use of available resources. This approach could be easily applied to an existing heuristic based general assignment problem solver to improve the solution quality.

Keywords

General Assignment Problem, Capacitated Transshipment Problem, Genetic Algorithm, Metaheuristics, Unmanned Air Vehicles.

1 Introduction and Background

The generalized assignment problem (GAP) is a well-known, NP-complete combinatorial optimization problem. The GAP consists in assigning a set of tasks to a set of agents with minimum cost. Each agent has a limited amount of a single resource and each task must be assigned to one and only one agent, requiring a certain amount of the resource of the agent. Exact and heuristic algorithms have been suggested to solve the GAP [CAT92], [OSM95],[RAM98],[ROM00]. This paper is an attempt to apply a genetic algorithm to improve the quality of the solution for an existing heuristic based greedy solution for a GAP. Various attempts have been made for the application of a genetic algorithm to solve the GAP. One such approach is to use the GA as the heuristic [CHU97]. In this approach the solution is based on the genetic algorithm. It requires a good representation of the problem space. Special considerations should be made to avoid invalid solutions to the problem. Another attempt is made using a Constructive Genetic Algorithm (CGA), a modified genetic algorithm [LOR02]. In this approach a bi-objective search is used to reduce the computations. In this approach also the CGA must be tailored to be applied to the GAP.

In this paper we are proposing a mechanism to improve an existing GAP solver by adjusting the existing heuristic. The heuristic is tweaked using a set of parameters suggested by the genetic algorithm. This approach reduces the amount of involvement required to understand the existing solution for a GAP. This approach could be easily applied to any existing heuristic based GAP solver to improve the solution quality. Many real life applications can be formulated as a GAP. Examples include resource scheduling, the allocation of memory space in a computer, the design of a communication network with capacity constraints at each network node, assigning software development tasks to programmers, assigning jobs to computers in a network, vehicle routing problems, and others. In this paper we use an existing solver for a capacitated transshipment problem (CTP). Capacitated transshipment problem is a special case of a GAP. This work demonstrates the application of a GA to improve the solution quality provided by the existing solver. The CTP is different from the normal shipment problem. In CTP we can have transshipment points through which the supplier could reach a demand point. The existing CTP solver used is proposed in [NYG01] for a Low Cost Autonomous Attack System (LOCAAS).

LOCAAS weapons systems are small aircrafts, each with a small turbojet engine with sufficient fuel to fly for about 30 minutes. These will be deployed in teams and the objective is to be able to assign a set of targets. This problem has been approached as a linear programming problem [NYG01]. In this approach each LOCAAS gets an assignment for one target. In the event of having multiple targets, it is suggested to use an iterative version of the above solver. In the iterative approach each iteration will allow the solver to freeze one target for a specific Unmanned Air Vehicle (UAV) and then go ahead with the rest of the targets until all the targets are assigned. This will lead to a sequence of tasks for each UAV thus providing it with a plan of attack. In this paper we try to improve the global solution quality of the iterative approach by introducing a genetic algorithm. The GA is used to influence the decision on fixing targets to the respective UAV.

A genetic algorithm is an "intelligent" probabilistic search algorithm. It simulates evolution by taking a population of solutions and applying genetic operators in each reproduction. Each solution is evaluated and better solutions are given more opportunities to reproduce.

Combining existing solutions through the process of reproduction generates new solutions. [GOL89],[CHU97]

We approach the assignment of targets to the UAVs as a generalized assignment problem. We propose the use of a GA to suggest better variations to the existing greedy solver as a novel approach. In this approach changes done to the existing solver will not affect the GA based application. It can run independently and suggest variations to the solver. Iterative Capacitated Transshipment (ICTP) Solver is an existing greedy solver for the GAP. In each iteration the solver will freeze one agent task point, until the solution runs out of supplies or all the tasks are assigned. The greedy heuristic used in this approach will fix the agent task point for the first agent with the ability to finish the task.

2 Proposed Design

In this section we give a high level description of the proposed solution. The solution consists of three components, namely the capacitated transshipment problem solver (CTP), Iterative CTP, and the GA based ICTP.

2.1 Solution with the Capacitated Transshipment Problem Solver.

The existing CTP solver will assign the best set of targets for the available set of UAVs. Each UAV will be assigned only one target. Given n UAVs and m targets the CTP will assign n targets to n available UAVs.

2.2 Extension using the Iterative CTP to plan a path for the UAVs.

The ICTP solver will assign a set of targets for each UAV. In this approach the UAV with the earliest completion time will be permanently assigned that target and the CTP solver will be executed again with the remaining targets. Primary objective is to be able to assign multiple targets to each UAV. Targets will be assigned to each UAV until it runs out of resources or there are no more targets to be assigned.

2.3 Improving the ICTP solution using a GA

ICTP approach is a greedy approach to arrive at a quick solution. It is clear that the naïve approach of fixing the target for the UAV with the earliest completion time will not arrive at a near global optimum for all cases. This observation leads us to consider other options to fix the targets. In this paper we have tried three different target fixing schemes by assigning weights to the UAVs and targets. The weighting scheme allows the solution to fix the targets with some influence from the respective weights. The GA supplies the weight for each item. The greedy algorithm then uses this weight as part of the metric that determines the order of selection. This allows the GA to reorder the greedy order suggested by the naïve approach. Quality of each solution is measured by observing the target assignment for total coverage and resource utilization. The GA will use feedback from the quality of solution to guide the search for the best set of weights.

3.0 Implementation

In this section we give a detailed description of the implementation. An existing implementation of the ICTP was used with a publicly available GA implementation. Most of the implementation effort was spent on writing glue code and application specific routines.

3.1 Genetic Algorithm

GENESIS is a system for function optimization based on genetic search techniques [GRE87]. This is implemented in C and the code is publicly available. The existing ICTP is implemented in C++. The interface requirement for the GA is to be able to supply the set of weights and get the evaluation from the ICTP. Figure 1 shows an outline of the proposed setup for the solution. The solver will take the available UAVs and the identified targets as an input. It will output a list of targets assigned for each UAV. The existing implementation of ICTP was modified to allow multiple target fixing schemes to be used. Several approaches were tried and we finally decided on having the GA call the ICTP through the main program interface. This approach allowed to setup this application with the least amount of modifications to the existing solution.

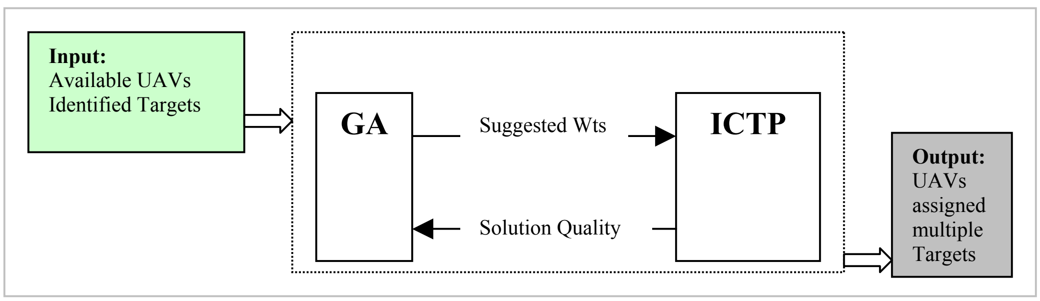

Figure 1

3.2 Weight Schemes for the ICTP.

The weighting scheme allows us to influence the target assignment for each iteration of the CTP. This is achieved by suggesting an importance to each assignment. Each assignment suggested by the CTP is ranked with a function that takes time to complete, and suggested weight into account. Top ranked assignment is fixed for that iteration. Different weight values selected by the GA will cause items to be assigned in order based on the weight value. Three different schemes were evaluated in assigning weights. They are UAVs only, Targets only and weights for both UAVs and Targets together.

3.3 GA Evaluation Function

The evaluation function is an important aspect of any GA based search optimization [GOL89]. In this application the final objective is to assign the maximum possible number of targets for the available UAVs. Desired qualities of the expected solution are, maximum coverage and the ability to maximize on the average number missions per UAV. Several variations were tried and the following function was used as the evaluation function.

$$\text{QualityOfSolution} = \frac{\sum_{t \in targets} t.assigned}{\sum_{u \in UAVs} u.used} \times \frac{1}{(1 + \sum_{t \in targets} \neg t.assigned}$$

Where: t.assigned: indicates a target that is assigned in this solution; t.used: indicates a UAV used for this solution; t.assigned and t.used are boolean functions; targets : all targets in test bed, UAVs: all UAVs available for the task.

4 Experiments and Results

In this section we present experimental results to show the improvement in quality of solution with the proposed approach. Initially we present 2 selected graphical examples of test cases where the proposed solution performs better than the existing solution. Finally we present the results of simulations on randomly generated test cases.

4.1 Examples of solutions.

Figure 2 shows two example scenarios of how the GA based solution performs better than the existing ICTP solution. The simulations were done for 2 UAVs with 6 targets and minimal amount of combined range (radius of maximum possible flight distance) for the UAVs to cover all targets. In Figure 2 left, two targets are not assigned where the GA based solution is able to assign all targets. In figure 2 right the GA based solution is able to capture all the targets where the ICTP based solution will miss 2 targets. It can be clearly observed that the GA is able to avoid getting stuck on local optimums.

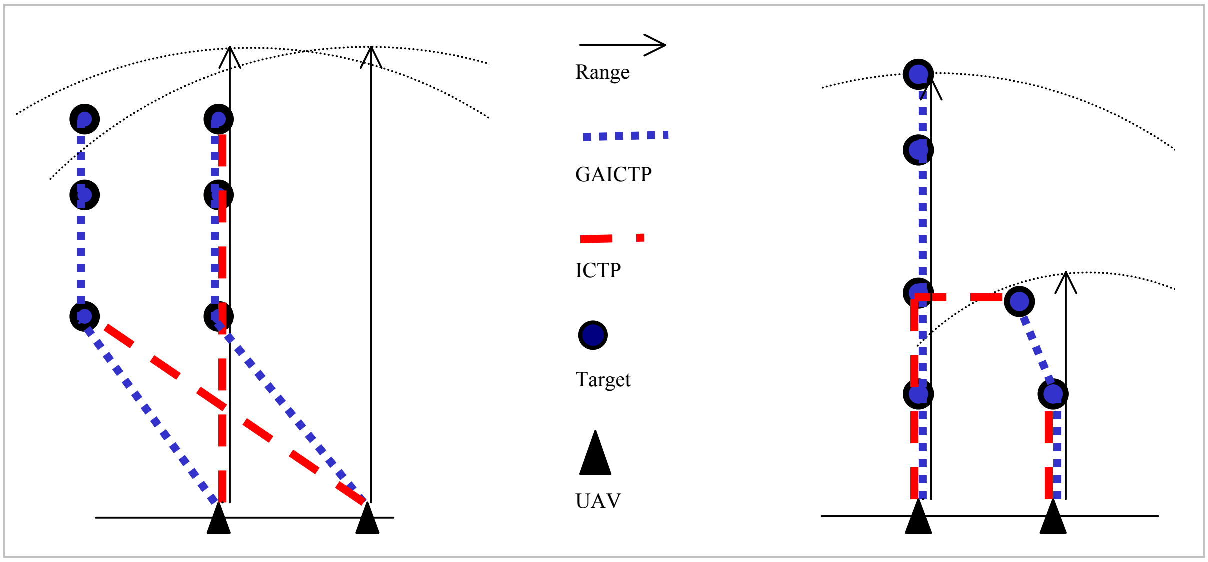

Figure 2

4.2 Simulation Results for Randomly Generated Test Cases.

Simulations were conducted with randomly generated sets of test cases. Each target was assumed to be equally valuable. Targets were spread around an area that can be covered by the combined range available on all the UAVs. Results of the experiments are given on Table A in the Appendix. The solution is compared against the existing ICTP. Summarized results are given in Table 1. Measure of comparison used is the ratio of targets covered by the solution. It is clear from the values that the GA based solution is better than ICTP. There is also a slight improvement shown with an increase in the number of weights in the weight scheme. The three weight schemes discussed in section 3.2 are compared. Weight usage is given as a ratio against the scheme that uses the maximum number of weights.

	ICTP	GA		
Weight Scheme	**-**	**UAVs**	**Targets**	**UAVs & Targets**
Coverage	0.896	0.944	0.950	0.958
Weight usage	-	0.2	0.8	1.0

Table 1

Table A (in Appendix) also shows number of trials and the generation number to achieve 10 percent convergence for each solution. This can be used as an indication of the amount of computation required for each solution. Each trial requires the execution of the ICTP. Most of the computational overhead is spent on the ICTP. Computational cost can be estimated by observing the number of trials required.

5 Conclusions and Future Work.

We have presented a clear-cut application of a GA to improve an existing solver for a CTP. We have clearly shown that the GA based solution is better than the existing ICTP. Thus we have demonstrated the capacity of a GA to improve an existing solution for a General Assignment Problem. This approach reduces the amount of involvement required to understand the existing solution for a GAP. This approach could be easily applied to any existing heuristic based GAP solver to improve the solution quality. As described in this paper the GA can be applied independent to the existing solver.

Further work can be done to evaluate a reliable stopping condition for the GA for this application. Computational cost should also be evaluated for feasibility purposes. Attempts should be made on other application areas to substantiate the broader claim of using a GA to improve an existing heuristic based GAP solver.

Appendix (Simulation Results)

	ICTP		GA											
Wt Sch.	-	-	UAVs				Targets				UAVs & Targets			
					Convergence				Convergence				Convergence	
Exp #	AsnTgs	QOS	AsnTgs	QOS	Gen #	Trial #	AsnTgs	QOS	Gen #	Trial #	AsnTgs	QOS	Gen #	Trial #
1	22	43.5	23	66.7	81	1267	23	66.7	20	363	24	100.0	29	545
2	22	45.5	24	100.0	29	462	24	100.0	20	378	24	100.0	19	373
3	24	100.0	24	100.0	29	462	25	250.0	14	275	25	200.0	19	370
4	23	76.9	24	111.1	25	372	24	111.1	20	374	24	111.1	14	278
5	22	47.6	25	200.0	98	1536	25	200.0	24	455	25	200.0	14	284
6	25	200.0	25	250.0	81	1267	25	250.0	20	374	25	250.0	48	901
7	23	66.7	24	111.1	39	633	24	111.1	20	367	25	250.0	29	550
8	25	250.0	25	250.0	58	902	25	250.0	25	467	25	250.0	24	461
9	24	111.1	25	250.0	58	902	24	111.1	35	643	24	111.1	28	541
10	23	71.4	24	111.1	70	1088	25	250.0	39	728	24	111.1	43	810
11	22	45.5	25	250.0	58	902	24	111.1	20	374	25	200.0	20	379
12	22	38.5	22	40.0	70	1087	23	62.5	25	463	23	58.8	24	464
13	19	18.9	21	27.8	73	118	22	38.5	15	284	22	40.0	72	1276
14	22	45.5	24	111.1	40	642	25	250.0	15	281	25	250.0	14	276
15	20	23.8	22	41.7	48	730	22	41.7	15	280	23	55.6	14	273
16	22	45.5	23	58.8	53	825	23	58.8	25	466	23	62.5	24	460
17	24	100.0	24	100.0	46	734	24	100.0	19	366	24	100.0	14	279
18	21	31.3	22	41.7	61	907	22	43.5	18	360	22	41.7	14	275
19	23	66.7	24	90.9	17	278	24	111.1	9	182	24	111.1	14	271
20	20	25.6	22	43.5	59	904	22	41.7	20	368	23	58.8	51	906
Avg	22.4	47.6	23.6	76.9	54.65	800.9	23.75	83.3	20.9	392.4	23.95	90.9	26.4	498.6
Succ	**0.896**		**0.944**				**0.950**				**0.958**			

TABLE A

Key **AsnTgs** : Number of targets assigned by the solution (total number of targets = 25) **QOS:** Quality of solution (using Evaluation Function)
Convergence Gen# | Trial# : number of generation / trial when 10 % convergence was evident on solution **Wt Sch:** Weighting Scheme

References

[NYG01] Dynamic Network Flow Optimization Models for LOCAAS Resource Allocation, Kendall E. Nygard, Program on Unmanned Combat Air Vehicles, Office of Naval Research, 2001.

[RAM98] Adaptive Approach Heuristics For The Generalized Assignment Problem, Helena Ramalhinha, Laurena Daniel Serra, 1998.

[CHU97] A genetic algorithm for the generalized assignment problem, P.C. Chu and J.E. Beasley, Camp. Opns. Res 24(1), 17-23, 1997.

[LOR02] A Constructive Genetic Algorithm For The Generalized Assignment Problem, Luiz A.N. Lorena, Marcelo G. Narciso, J.E. Beasley, working paper, 2002.

[ROM00] A Class of Generalized Greedy Algorithms for the Generalized Assignment Problem, H.E. Romeijn, D. Romero Morales, Discrete Applied Mathematics 103, 209-235, 2000.

[CAT92] A Survey of algorithms for the generalized assignment problem, D. Cattrysse, L.N. Van Wassenhove, Eur. J. Oper. Res. , 1992

[OSM95] Heuristics for the generalized assignment problem, simulated annealing and tabu search approaches, I. H. Osman, OR Spektrum, 1995.

[GRE87] A User's Guide to GENESIS, John J. Grefenstette, Navy Center for Applied Research in Artificial Intelligence, Naval Research Laboratory, Washington, D.C., August 1987.

[GOL89] Genetic Algorithms in Search Optimization, and Machine Learning, D.E. Goldberg, Addison Wesley, 1989.