A Genetic Algorithm Approach to the Artillery Target Assignment Problem

Burçin Sapaz, İsmail Hakkı Toroslu, Göktürk Üçoluk

Middle East Technical University, Ankara, Turkey

Abstract. In this work a new assignment problem with the name "Artillery Target Assignment Problem (ATAP)" is defined. ATAP is about assigning artillery guns to targets at different time instances while some objective functions are to be optimized. Since an assignment made for any time instance effect the value of the shooting ATAP is harder than the classical assignment problem. As far as the objective functions are concerned we define a base case problem. We elaborate on some possible variations of the base case which are exceptionally interesting in the military domain. For two of such problems, genetic algorithm solutions with customized representations and genetic operators are developed and presented.

1 Introduction

In military, assignment of artillery guns for targets is a complex decision process which includes many different parameters to be considered [6, 8, 10–13, 17]. Mainly at each discrete time instance, available guns are assigned to proper targets with the best possible assignment in order to perform shooting to these targets. Although, it is already difficult to determine the best assignment for a single time instance, the main challenge is that the decisions for a time instance usually affects the shooting value.

In this problem, there are three main kinds of elements to be defined:

Definition 1 (Target). *Targets are possible artillery targets and shooting up these targets is the primary goal in artillery assignment. There are m targets;* T_1, T_2, T_3,..., T_i, ..., T_m

Definition 2 (Gun). *Guns are artillery weapons and at each discrete time instance each of them fires a shot. There are n guns;* W_1, W_2, W_3,... W_j,..., W_n

Definition 3 (Time). *Time is expressed by discrete time steps. It is assumed that the decision process for artillery branch of military starts and ends between each consecutive discrete time steps. As a result, at each discrete time step a gun can shoot. Discrete time steps starts at 1 and ends at p;* Z_1, Z_2,..., Z_k,..., Z_p

Artillery target assignment problem (ATAP) is about assigning n guns to m targets in p discrete time instances with best possible cost by means of required optimization parameters. Different military tactics will result in different cost functions. Among most popular cost functions are

- Maximizing the total value of targets which has been shot up by guns.
- Minimizing the total time that is required to shot up all targets.
- Maximizing the total value of shots at a limited time period.
- Minimizing the total displacement of all targets that has been shot up by guns.

Instead of above possible cost definitions, in our model, we will assume that each shot of a gun to each target for each time instance is represented by a cost value, which we will aim to maximize. We will not be interested in whether the target was killed or not, since at any time the purpose of the artillery assignment is to determine the best assignment for the given targets and a given discrete time.

In ATAP Each single gun that belongs to the same battery hierarchy is assumed to be identical. Furthermore, every battery has the same hierarchy structure: So batteries are sectionwise and gunwise identical. Targets though, can differ in their target values, shot values and target sizes.

In the most basic form each gun independently is assigned to a target at a time instance. Thus it looks like three-dimensional matching problem [2, 3, 5, 19], which is a well-known NP-hard problem. On the other hand, in ATAP each gun can fire at each discrete time. Therefore, by repeating each gun for each discrete time we can construct a two-dimensional matching problem, which has a known efficient solution due to Kuhn and Munkres [9, 15].

The base case is defined by:

- Each gun can shoot at any time instance.
- A target that got shot (once) is assumed to be destroyed.
- Each target receives exactly one shot.

This basic form of the problem is not very realistic. In military domain various firing techniques are adopted which corresponds to variations of the basic form. Such as

- treating a group of targets as a single target and firing on it with a single gun,
- firing on one target several shots to cover the whole region of the target,
- firing a group of guns simultaneously to a single target, etc.

In this study we investigate two important variations, namely joining targets to treat as a single target, and firing guns of a battery or a section together for large targets.

ATAP is structurally a complex problem making its GA representation complex too [4, 14, 16]. The main contribution of this work is modeling ATAP as a

GA problem with suitable and efficient representation of its instances and algorithms to handle GA operators. We also show that our modeling produces quite successful results.

The rest of the paper is organized as follows: Section 2 describes two variation of the base case that we have solved by means of GA. The following section 3 exhibits the results of various runs, finally we conclude in section 4.

2 Handled Cases and GA Implementations

In this work we consider two variations to the base case introduced above. These two cases are frequently met in real military domain.

Target joining: Since artillery gun ammunition covers a considerable target area (quite larger than a single point), it is possible to destroy more then one target by a single shot. Therefore, it is common practice to join targets that occupy small area and are in close proximity.

Hierarchal formation of guns: A typical artillery battery contains six guns which are divided into two sections. Depending on the size of the target either the whole battery or only a section or even a single gun shoots. For much larger targets even more then one battery might be assigned to it.

These two cases are solved by GA. As far as the GA data structure is concerned it has to be noted that:

- Linear chromosomal structure for these problems proves to be insufficient and inefficient. Therefore we have adopted a 3-Dimensional representation which matches with the 3-Dimensional structure of the problem.
- Though natural, the representation is handicapped with invalid chromosome production as far as crossover and mutation is concerned. Therefore there is a need for efficient repair algorithms [1].

In the following subsections the 3-Dimensional representations, the crossover and mutation operations and the repair algorithms are introduced.

For both cases handled the chromosome structure is a 3-Dimensional binary array with dimensions in guns, targets and the discrete time. Firing a gun, to a target at a time instance is represented by a `1` in the corresponding entry in the chromosome. Otherwise the entry is `0`.

Also the objective function for both cases is defined as

$$\max \sum_{i \in guns} \sum_{j \in targets} \sum_{k \in times} Chromosome_{ijk} \times ShootingCost(i,j,k) \qquad (1)$$

In equation 1 we assume all $ShootingCosts(i,j,k)$ values are provided for all i,j,k values, namely the targets, guns and time dimensions.

Solution that maximize the objective function is searched by means of GA. The GA employed is of one-point crossover and 2-point gene-swap mutation type and uses elitist selection. The selection replaces the worst 5% of the new population with the best 5% of the previous. Mutation is %1.0 mutation/gene-crossover. Population size is 100.

The crossover produces invalid chromosomes. How this is fixed is explained below. Mutation is implemented as a gun assignment swap among two random chosen genes that belong to the same time dimension. This ensures that no invalid chromosomes are produced due to mutation.

2.1 Target Joining

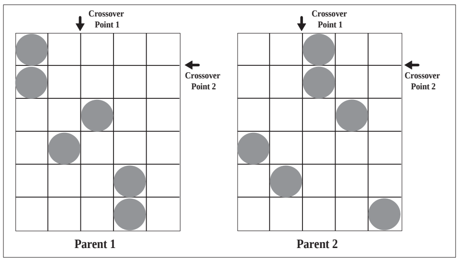

Fig. 1: Sample Parent Individuals for Variation of Target Joining (columns run over guns and rows represent targets)

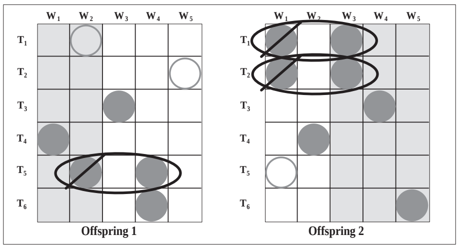

Fig. 2: Offsprings created after a column crossover and related repairs

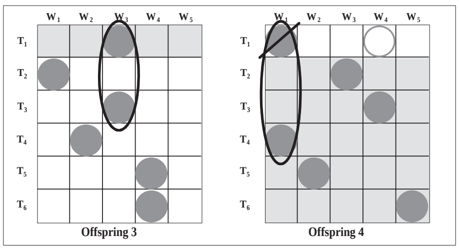

Fig. 3: Offsprings created after a row crossover and related repairs

In order to represent target joins we will allow more than one entries to be `1` in the target dimension. Possible join sets are priorly known. The GA run will determine whether the set will be formed or the targets will be fired upon individually.

We implement the single point crossover to be perfomed in all dimensions. In particular, the crossover in the time dimension is trivial and does not cause any invalid chromosomal formation. On the other hand this is not so for guns and targets dimensions. Here a repairing algorithm is needed for conducting an efficient GA search. Furthermore, the initial pool is formed to contain only valid chromosomes.

Below the repair algorithms for each of the two dimensions, namely guns and targets, are given separately.

Assume we have two sample individuals shown in figure 1. In order to be readable we explain the repair algorithm for a single time instance. Considering the time dimension as well will not alter the discussion of the repair algorithm.

As it is usual with one point crossover, the crossover line splits the parent chromosomes into two pieces each. Then swapping these pieces two offsprings are generated. It is quite possible that these offsprings are invalid. The type of invalid offspring formation is different for a split in guns dimension (Crossover point 1) with respect to a split in targets dimension (crossover point 2). Hence, different repair algorithms have to be employed. Figure 2 displays the two offsprings generated by a cut point in the guns dimension (Crossover point 1). Similarly, Figure 3 exhibits the case for a cut in the targets dimension (crossover point 2).

In both the Figures 2 and 3 the gray marked genes represent the picture right after the swap. There are three possible kinds of violations that has to be dealt with:

1. A single target get more than one shot (two entry in a row). This can occur in a gun dimension crossover.
2. Two unjoinable targets got joined (a invalid two entry in a column). This can occur in a target dimension crossover.
3. Targets that must got shot in this time instance (due to their parents) might remain unshot after the crossover. (no value in row though parent had it)

For case (1) and (2) the repair phase will cancel randomly one of the two columns or rows. After the cancelation it is possible that additional cases of (3) might get generated. For case (3) the repair is done at last by randomly assigning remaining guns to remaining targets. These are represented by blank circles in the figures.

2.2 Hierarchical Formation of Guns

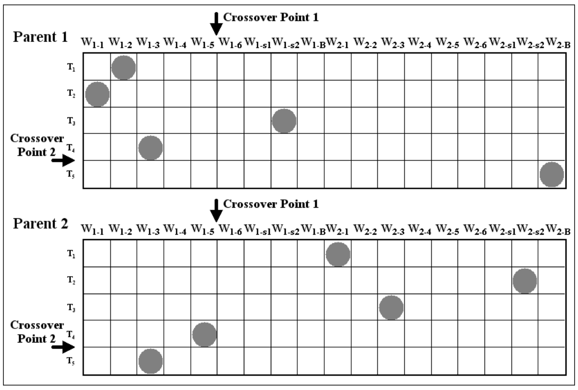

Fig. 4: Sample Parent Individuals for Variation of Hierarchy in guns

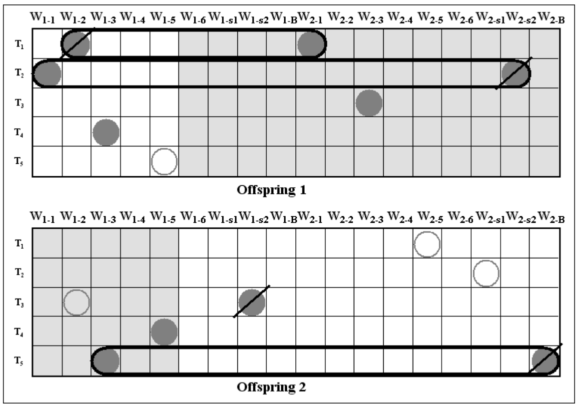

Fig. 5: Offsprings created after a column crossover and related repairs

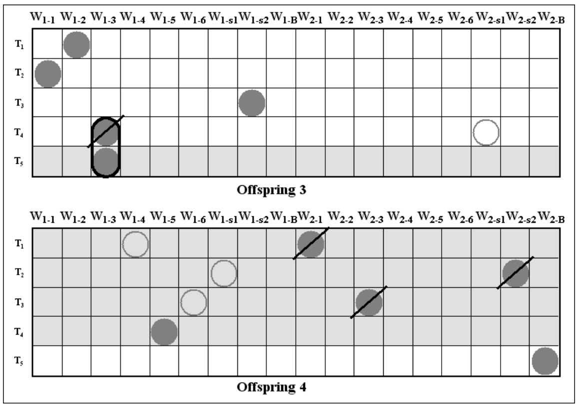

Fig. 6: Offsprings created after a row crossover and related repairs

In order to represent hierarchial formations of guns, each gun, as well as each hierarchial formation is represented by a gene in the guns dimension. Only a single one of these genes will be allowed to be `1`.

In a valid structure, at a time instance, no two targets can be shot by two gun formations which are hierarchically related.

In this case as well, we implement the single point crossover in all three dimensions. Similar to the target join case, the crossover in the time dimension is trivial and does not cause any invalid chromosomal formation. On the other hand this is not so for guns and targets dimensions. Here a repairing algorithm is needed for conducting an efficient GA search. Furthermore, the initial pool is formed to contain only valid chromosomes.

Below the repair algorithms for each of the two dimensions, namely guns and targets, are given separately.

Assume we have two sample individuals shown in figure 4. In order to be readable we explain the repair algorithm for a single time instance. Considering the time dimension as well will not alter the discussion of the repair algorithm.

As in the previous case we use a single point crossover, which splits the parent chromosomes into two and the by swapping the pieces coming from two parents forms two new offsprings. Also, as in the previous case possible violations may arise in the new offspring even though the parent chromosomes are valid. The forms of invalidness are different for two different crossovers in two dimensions, namely the splits in the guns (crossover point 1) and in the target dimensions (crossover point 2). Figure 5 depicts the offsprings generated from the former one, and Figure 6 depicts the offsprings generated from the latter one.

The forms of violations are as follows:

1. A single target might receive more than one (two) shots. This can occur for the crossovers in the guns dimension.
2. Two targets might be shot by the same gun system. This case can occur for the crossovers in the target dimension.
3. An invalid case might be generated by assigning two hierarchically related guns to targets in the single time instance. This case may occur in both forms of the crossovers.
4. A target that must be shot in the given time instance might not be shot at all. This case may occur in both forms of the crossovers also.

The first two cases are handled by randomly canceling one of the two shots in in these dimensions. This might generate more cases of (4) also. Case (3) is also handled by canceling out one of the shots of the hierarchically related guns. At this point again notice that since parents are assumed to be valid not more than two guns can be hierarchically related can shoot at a time instance. Therefore, in resolving this violations one of these two shots should be cancelled and this can be done by choosing either higher or lower gun in the hierarchy. This may also generate more cases of (4). The last case is handled by randomly assigning remaining valid guns to remaining targets. Notice that not all idle guns are valid, since no two guns in the same hierarchical path can be utilized at the same time.

3 Experimental Results

As part of the experimental work a an artillery weapon simulator has been implemented [18]. GA solver is based on GAlib [7]. The screenshot of the UI is displayed in Figure 7. This simulator has been used for two purposes:

1. Realistic data generation for experiment.
2. Visual display of events of the experimentation on the timeline.

Inputs to data generation (gun and target specifications and their movements, terrain obstacles) are taken directly from real military domain. The movements of the targets are inputted through an timeline based scenario editor. Using this simulator datasets, properties of which are given in Table 1, are generated.

Furthermore, for each gun-target pair the followings are assigned as a function of time.

- Cost value (value of gained by hitting the target),
- effectiveness (of the gun on that target),
- distance (between the gun and the target),
- availability of the target (hitting as early as possible of the target is desired)

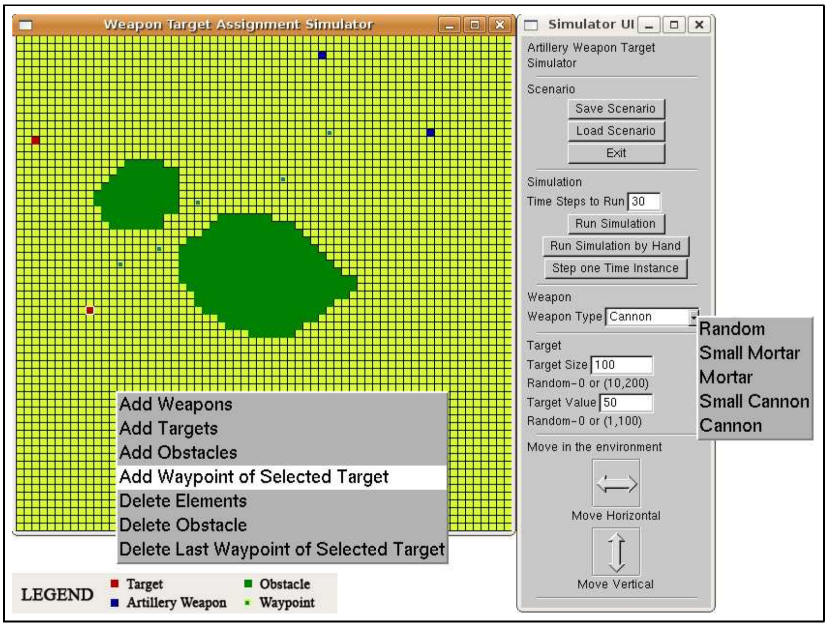

Fig. 7: Artillery Weapon Target Simulator

The improvement of score of the best individual in the GA pool is displayed for all datasets of both problem cases, in the Figures 8,9,10,11,12,13 as observed, they display a very characteristic pattern of GA. The convergence speed is very satisfying, which exhibits a linear relation with the size of the dataset.

The manual verification of the solutions proved that the results are very close to the real optimal.

Table 1: Number of Elements in Datasets

	Dataset 1	Dataset 2	Dataset 3
Number of Weapons for target joining	5	10	10
Number of Weapons for hierarchial formation	45	90	90
Number of Targets	10	100	100
Number of Time Instances	50	50	100
Number of Total Joins for target joining	32	396	943

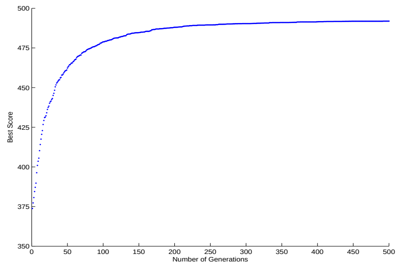

Fig. 8: Results of dataset 1 for variation of target joining

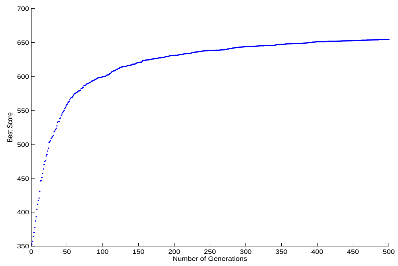

Fig. 9: Results of dataset 1 for variation of hierarchy in weapons

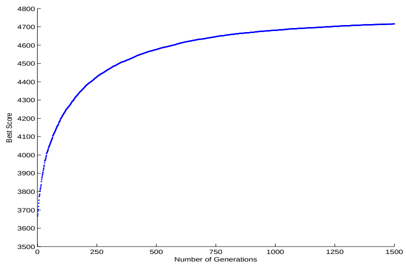

Fig. 10: Results of dataset 2 for variation of target joining

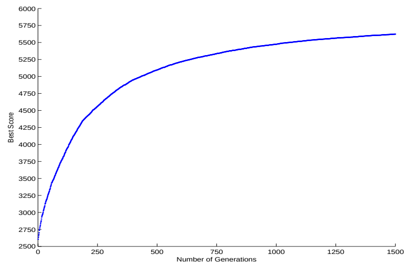

Fig. 11: Results of dataset 2 for variation of hierarchy in weapons

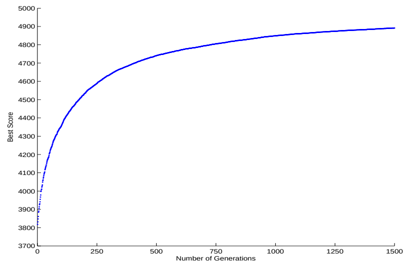

Fig. 12: Results of dataset 3 for variation of target joining

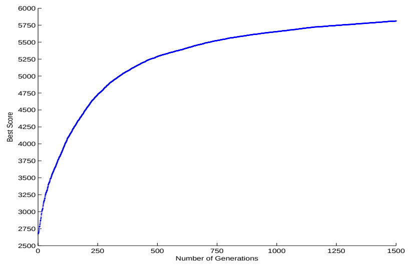

Fig. 13: Results of dataset 3 for variation of hierarchy in weapons

4 Conclusion

Artillery target assignment problem can be considered as a novel area of study in the field of assignment problems. These problems are complex in nature and known to be NP hard. GA proves to be feasible in finding sub-optimal solutions. The complex nature of these type of problems require suitable chromosomal representation and task specific GA operators.

In this work we proposed two GA representations and their associated GA operators for two main variants of ASAP. Experimental study proved that the proposed GA structure is efficient in producing admissible solutions.

Future work may concentrate on solving other sub-problems and mixtures of sub-problems by means of GA in which new representations and operators will probably be required.

References

1. Thomas Baeck, David B. Fogel, Z. Michalewicz, Handbook of Evolutionary Computation, C5.4 Repair Algorithms. Published by CRC Press, 1997.
2. Egon Balas, Matthew J. Saltzman. An Algorithm for the Three-Index Assignment Problem. Operations Research, 39(1):150-161, 1991.
3. François Bourgeois, Jean-Claude Lassalle. An Extension of the Munkres Algorithm for the Assignment Problem to Rectangular Matrices. Communications of the ACM, 14(12):802-804, 1971.
4. Carlos A. Coello Coello. Theoretical and Numerical Constraint-Handling Techniques Used with Evolutionary Algorithms: a Survey of the State of the Art. Computer Methods in Applied Mechanics and Engineering, 191(11-12):1245-1287, 2002.
5. Reuven Cohen, Liran Katzir, Danny Raz. An efficient approximation for the generalized assignment problem. Information Processing Letters, 100(4):162-166, 2006.
6. David G. Galati. Game theoretic target assignment strategies in competitive multi-team systems. Phd Thesis, University of Pittsburgh, 2004.
7. GAlib: Matthew's Genetic Algorithms Library. http://lancet.mit.edu/ga/
8. Cai Huaiping, Liu Jingxu, Chen Yingwu, Wang Hao. Survey of the research on dynamic weapon-target assignment problem. Journal of Systems Engineering and Electronics, 17(3):559-565, 2006
9. H. W. Kuhn. The Hungarian Method for the Assignment problem. Naval Research Logistic Quarterly, 2:83-97, 1955.
10. Ojeong Kwon, Kyungsik Lee, Sungsoo Park. Targeting and Scheduling Problem for Field Artillery. Computers and Industrial Engineering. Vol. 33, Nos 3–4, pp. 693-696. 1997.
11. Zne-Jung Lee, Wen-Li Lee. A Hybrid Search Algorithm of Ant Colony Optimization and Genetic Algorithm Applied to Weapon-Target Assignment Problems. Lecture Notes in Computer Science 2690, pages 278-285, 2003.
12. Zne-Jung Lee, Shun-Feng Su, Chou-Yuan Lee. A Genetic Algorithm with Domain Knowledge for Weapon-Target Assignment Problems. Journal of the Chinese Institute of Engineers, 25(3):287-295, 2002.
13. Zne-Jung Lee, Shun-Feng Su, Chou-Yuan Lee. Efficiently Solving General Weapon-Target Assignment Problem by Genetic Algorithms With Greedy Eugenics. IEEE Transcations on Systems Man and Cybernetics, Part B Cybernetics, 33(1):113-121, 2003.

14. Zbigniew Michalewicz. A Survey of Constraint Handling Techniques in Evolutionary Computation Methods. Proceedings of the Fourth Annual Conference on Evolutionary Programming, MIT Press, Cambridge, MA (1995), pp. 135-155.
15. James Munkres. Algorithms for the assignment and transportation problems. Journal of the Society for Industrial and Applied Mathematics, 5(1):32-38, 1957.
16. David Orvosh, Lawrence Davis. Using a Genetic Algorithm to Optimize Problems with Feasibility Constraints. Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE Press, New York (1994), pp. 548-553.
17. Jay M. Rosenberger, Hee Su Hwang, Ratna P. Pallerla, Adnan Yucel, Ron L. Wilson, Ed G. Brungardt. The Generalized Weapon Target Assignment Problem. 10th International Command and Control Research and Technology Symposium, The Future of C2, McLean, VA. 2005.
18. Burin Sapaz. Artilllery Target Assignment Problem with Time Dimension. M.S. Thesis, Middle East Technical University. 2008.
19. Frits C. R. Spieksma. Multi index assignment problems: complexity, approximation, applications. Nonlinear Assignment Problems, Chapter 1, Springer, First Edition, 2000.