

# 요격미사일 배치문제에 대한 하이브리드 유전알고리즘 적용방법 연구

## (An Application of a Hybrid Genetic Algorithm on Missile Interceptor Allocation Problem)

\* 한 현 진(Han, Hyun Jin)\*

### ABSTRACT

A hybrid Genetic Algorithm is applied to military resource allocation problem. Since military uses many resources in order to maximize its ability, optimization technique has been widely used for analysing resource allocation problem. However, most of the military resource allocation problems are too complicate to solve through the traditional operations research solution tools. Recent innovation in computer technology from the academy makes it possible to apply heuristic approach such as Genetic Algorithm(GA), Simulated Annealing(SA) and Tabu Search(TS) to combinatorial problems which were not addressed by previous operations research tools.

In this study, a hybrid Genetic Algorithm which reinforces GA by applying local search algorithm is introduced in order to address military optimization problem. The computational result of hybrid Genetic Algorithm on Missile Interceptor Allocation problem demonstrates its efficiency by comparing its result with that of a simple Genetic Algorithm.

**Keywords : Hybrid Genetic Algorithm, Missile Interceptor, Allocation Problem**

---

논문접수일 : 2009년 10월 16일    논문게재확정일 : 2009년 12월 18일

\* The 3rd Logistics Support Command, Republic of Korea Army

† 교신저자

## 1. Introduction

Determining an optimal solution of a resource allocation problem which requires assigning limited resources to the needed is one of the most important tasks of the operational researchers. A combinatorial perspective is a specific character of the resource allocation problem. Travelers Salesman Problem (TSP) and Quadratic Assignment Problem (QAP) are the well known examples of the combinatorial problem[7]. Reeves [8] defined that a set or a sequence of integers or other discrete objects solution are the characteristics of the combinatorial problem.

Jewell [3] suggested that computational complexity of the combinatorial problem makes it difficult to solve for the researchers in his research which summarizes the analytical methods of the Operations Research (OR) problems. Numerous approaches including Dynamic Programming (DP) and Branch and Bound were performed to address combinatorial optimization problems. Due to the current innovation in computer technology, computer based optimization approaches are more popular than before addressing complex problems such as the combinatorial problem. While Evolutionary Algorithm (EA) such as Genetic Algorithm (GA), Simulated Annealing (SA) and Ant Colony System (ACO) are the techniques borrowed from other scientific fields, they have been proved to address the problems.

Most of the military problems are NP-complicated or NP-hard problems. It is not an easy task to address these problems by traditional optimization solutions approaches. Finding the safest and shortest route for redeploying troops, devel-

oping an effective deployment plan of detection equipment in General Out Post (GOP) and defining a best fire-power allocation strategy are the examples of military OR problems. Typically in air-defense artillery corps, the allocation strategy for missile interceptor is critical in order to maximize air-defense ability for hostile missiles with limited interceptors. Kim [4] introduced Missile-Interceptor Allocation (MIA) problem to minimize the number of destructed cities from the enemy missile attack. With an assumption that the number of interceptors are known and the amount of enemy missiles are unknown, it is required to find the number of interceptors for each city in order to minimize the vulnerability of enemy missile attack. So the structure of the solution set of MIA problem is similar to QAP.

Koopmans and Beckmann formulated QAP in 1957 [5]. They applied an optimization method to solve it. Numerous researchers such as Lawler, Heffley and Bazaara & Elshafei studied QAP using a Linear Programming and a branch and bound algorithm [1][2][6]. Since the traditional optimization approach for a big-size QAP could not be addressed, heuristic methodology has been applied once the computer technology has been developed. Wilhelm and Ward introduced SA in order to address QAP, Taillard applied Tabu Search (TS) approach, and Tate and Smith implanted GA[9][10][11]. All the results of the previous studies showed that the combinatorial problems such as QAP could be addressed by heuristic.

This study introduces hybrid GA approach in order to address military operations research problems. In the hybrid GA framework, TS

based local search method is implemented to the elitist of every generation for complementation the shortcomings of GA which could not explore local area, The MIA problem is used in order to show the usefulness of the hybrid GA approach.

## 2. Brief of MIA problem

### 2.1 Outline of MIA Problem [4]

Developing a proper defense strategy is one of the most important tasks of ministry of national defense. Among the defense strategies, intercepting missiles is the very core of the air-defense. If the missile attacking plan of enemy is known to us, no computation effort is needed to find out the place to protect and the amount of interceptors to deploy. Unfortunately in the real world, it is not opened to the hostile country. Thus, Military have to find out the optimal allocation combinations.

The summary of MIA problem is as below. Suppose that there is a plan to buy “N” interceptors to repulse an enemy’s long-range missiles in the Army Anti-air Defense Department. The decision maker wants to know the best allocation strategy for the interceptors. From the Army Intelligent Department “M” cities have to be protected from enemy missile attack and all the our cities have different strategic values. The values are predetermined by the Army Intelligent Department. Is is also considered that deploying fewer interceptors than attacking missiles in a certain city means that the whole city is destroyed.

There are four assumptions are applied for this problem in order to make problem simple

and clear the proceeding of hybrid GA approach ;

1. The target hitting rate of enemy missile should be 1. It means if the missile is not intercepted, the city under attack will definitely be destroyed.
2. The interceptor’s single shot kill probability is 1.
3. The interceptors deployed to protect the  $i^{th}$  city only can intercept the missiles which attack the exact city.
4. There will be no time constraints.

### 2.2 Mathematical formulation

Let  $x_i$  be the number of interceptors which deployed in the  $i^{th}$  city, and  $v_i$  be the strategic value of the  $i^{th}$  city. If the  $i^{th}$  city got attacked by  $x_i + 1$  missiles, the city will be destroyed. So, the expected damage value per unit missile for the  $i^{th}$  city is defined as  $\frac{v_i}{x_i + 1}$ . Finally, the objective of this problem is minimizing the expected damage of the unit missile.

The mathematical formulation for this problem is defined as;

$$\text{Minimize} \quad \sum_{i=1}^M \frac{v_i}{x_i + 1}$$

$$\text{St) } \quad \sum_{i=1}^M x_i = N, \quad \text{All } x_i \geq 0$$

Where,

$N$  : Number of interceptors to deploy

$M$  : Number of cities to protect

$v_i$  : Estimated value of the  $i^{th}$  city,  $i = 1, \dots, M$

$x_i$  : Number of Interceptors deployed in the  $i^{th}$  city,  $i = 1, \dots, M$

Let  $f(M, N)$ : methods for assigning  $N$  interceptions to  $M$  cities.

Then,

$$f(1, 1) = f(1, 2) = f(1, 3) = \dots = f(1, j) = \dots = f(1, N) = 1$$

$$f(2, 1) = {}_2C_1 = 2, \quad f(3, 1) = {}_3C_1 = 3, \dots, f(i, 1) = {}_iC_1 = i, \dots, f(M, 1) = {}_MC_1 = M,$$

$$f(2, 2) = 3 = f(1, 2) + f(2, 1)$$

$$f(2, 3) = 4 = f(1, 3) + f(2, 2) = f(1, 3) + f(1, 2) + f(2, 1)$$

$$f(3, 2) = 6 = f(2, 2) + f(3, 1) = f(1, 2) + f(2, 1) + f(3, 1)$$

$$f(3, 3) = 10 = f(2, 3) + f(3, 2) = 10$$

$$f(3, 4) = 15 = f(2, 4) + f(3, 3) = f(1, 4) + f(2, 3) + f(3, 3) = 1 + 4 + 10$$

$$f(4, 3) = 20 = f(3, 3) + f(4, 2) = f(3, 3) + f(3, 2) + f(4, 1) = 10 + 6 + 4$$

•

•

•

$$f(i, j) = f(i-1, j) + f(i, j-1)$$

•

•

•

$$f(M, N) = f(M-1, N) + f(M, N-1)$$

<Figure 1> Algorithm to calculate the size of combination  $(M, N)$

Since the objective function of above problem is NLP and the solution set is combinatorial, heuristic could be a good candidate to address the problem. If it is proved that the size of the solution space for this problem is huge, it might be another remission for heuristic approach. The size of solution space which is composed of “ $M$ ” cities and “ $N$ ” interceptor can be calculated from the algorithm derived in figure 1.

In order to prove that the algorithm is correct, a simple example which has “3-city and 10-interceptor” is considered. The total feasible solutions of the “3-city and 10-interceptor” problem are 66 combinations by counting all possible combination cases. By using the algorithm, the size of solution set is computed to 66.

Computing the solution space using algorithm

$$\begin{aligned} f(3, 10) &= f(2, 10) + f(3, 9) = f(1, 10) + f(2, 9) \\ &\quad + f(2, 9) + f(3, 8) = f(1, 10) + 2f(2, 9) \\ &\quad + f(3, 8) \\ &= f(1, 10) + 2(f(1, 9) + f(2, 8)) + f(2, 8) \\ &\quad + f(3, 7) = f(1, 10) + 2f(1, 9) + 3f(2, 8) + f(3, 7) \\ &= f(1, 10) + 2f(1, 9) + 3f(1, 8) + 4f(2, 7) \\ &\quad + f(3, 6) \\ &= \bullet \bullet \bullet \\ &= f(1, 10) + 2f(1, 9) + 3f(1, 8) + 4f(1, 7) \\ &\quad + 5f(1, 6) + 6f(1, 5) + 7f(1, 4) + 8f(1, 3) + 9f(2, 2) + f(3, 1) \\ &= f(1, 10) + 2f(1, 9) + 3f(1, 8) + 4f(1, 7) \\ &\quad + 5f(1, 6) + 6f(1, 5) + 7f(1, 4) + 8f(1, 3) + 9f(1, 2) + 9f(2, 1) + f(3, 1) \\ &= 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 9(2) + 3 = 66 \end{aligned}$$

<Table 1> Examples of solution combinations for MIA problem

M \ N	50	100	200	300	500
10	1.26E+10	4.26E+12	1.76E+15	6.29E+16	5.89E+18
15	4.79E+13	<b>3.13E+17</b>	3.14E+21	7.74E+23	8.62E+26
20	4.63E+16	4.91E+21	1.08E+27	1.78E+30	2.28E+34
25	1.75E+19	2.6E+25	1.14E+32	<b>1.21E+36</b>	1.73E+41
40	2.66E+25	5.05E+34	1.05E+45	2.4E+51	<b>4.08E+59</b>

```

define    Global parameters ( size of problem ( # of cities, # of interceptors ) )
define    GA parameters ( population size, mutation rate, maximum generations )
define    local search parameters ( Tabu list size, swap times )
initialize population, Best solution
convert    Data structure
compute    fitness
for ( g < maximum generations )
{
    reproduce children
    convert Data structure
    compute fitness
    select elitist in each generation
    // local search //
        initialize Tabu list
    for ( i < swap times )
    {
        swap the numbers assigned to two cities
        compute fitness
        update the elitist
        update Tabu list
    }
    update Best solution
}

```

<Figure 2> Procedure for hybrid GA

Table 1 summarizes the solution space size. Three possible case studies were conducted in this research; 15 city - 100 interceptors, 25 city - 300 interceptors and 40 city - 500 interceptors. The example size of MIA problems is illustrated in order to demonstrate the appropriateness of the heuristic approach. From the figure 1 and the table 1, it is explicit that the solution space increases exponentially by increasing “M”, so the heuristic is good candidate for this problem.

### 3. Heuristic Design for MIA problem

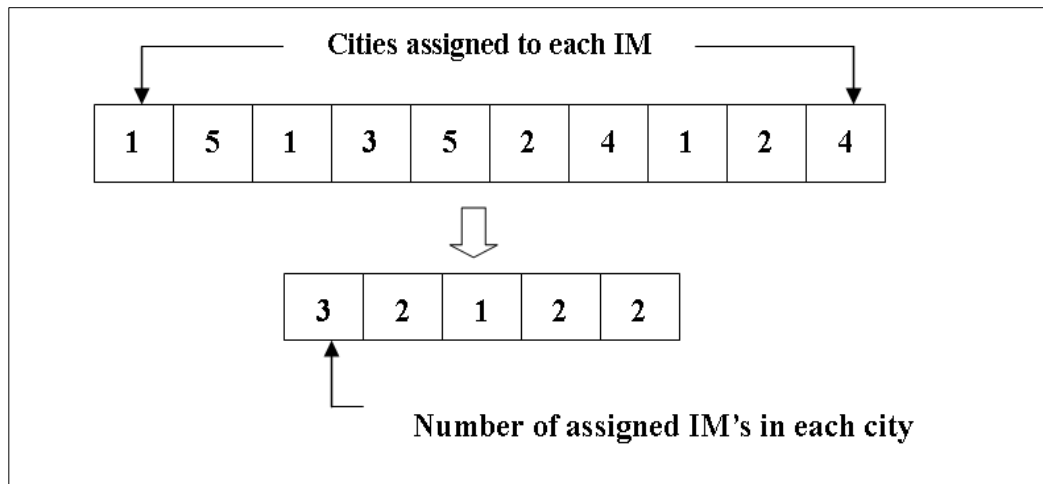
Adding a local search algorithm to GA is a current trend for GA applications in order to improve the fitness of the solution derived by heuristic because the basic GA does not conduct any local search. In theory any local search method could be used. The main idea for hybrid GA applied in this study is implementing a local search to the elitist of each generation: TS based local search method was executed. Figure 2 is the summary of the procedures of the hy-

brid GA approach applied in this research.

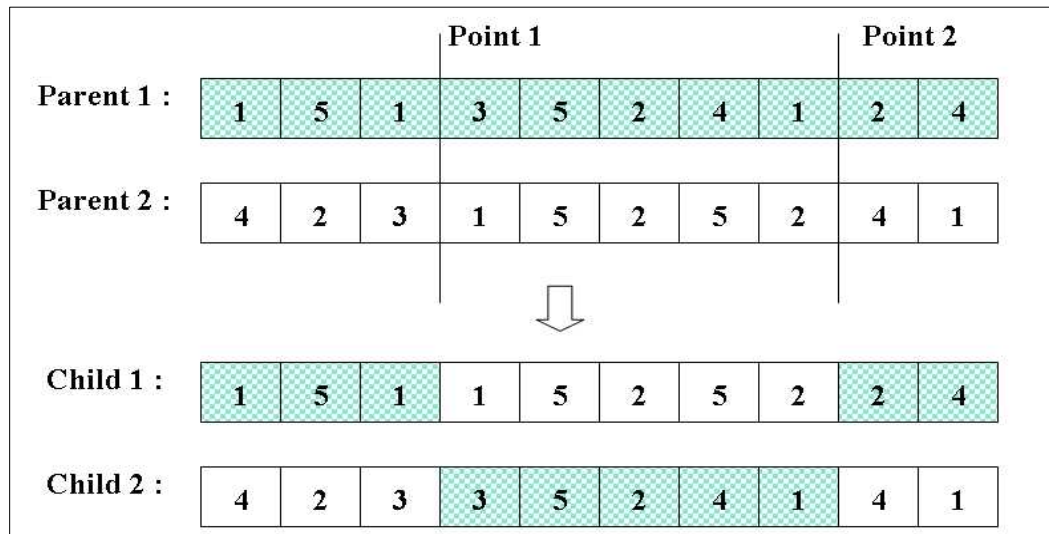
#### 3.1 Population encoding and data structure

The MIA problem addresses the question that “how many interceptors are deployed in the specific cities in order to minimize the threat of enemy missile attack?” Each chromosome has to imply all the information for assigning  $i^{th}$  interceptor to a randomly generated city. In the GA design for the MIA problem, each element of a population represents an interceptor allocated into a certain city. Another vital procedure is converting acquired chromosome into reasonable data structure in order to compute the fitness of each chromosome. By computing the fitness of the chromosome, it will be possible to figure out the elitist of each generation.

Figure 3 is an example of the structure of a chromosome. The example demonstrates the result of the data converting process in case of “5 city - 10 interceptor.” problem. The information of each chromosome is as below. The first line



<Figure 3> A structure of a chromosome

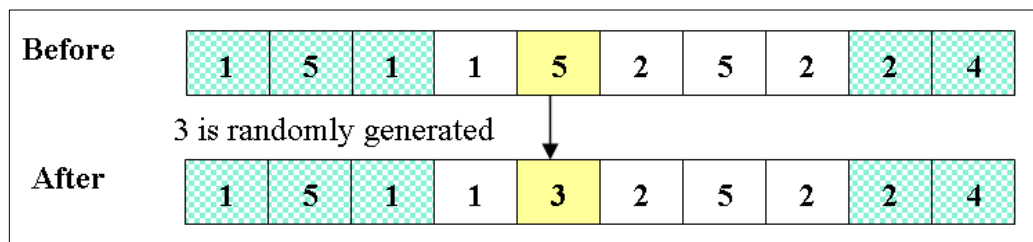


<Figure 4> Two point crossover

on the figure 3 demonstrates the interceptor allocation information; the 1<sup>st</sup> interceptor is assigned to the 1<sup>st</sup> city, the 2<sup>nd</sup> interceptor is assigned to the 5<sup>th</sup> city, the 3<sup>rd</sup> interceptor is assigned to the 1<sup>st</sup> city, the 4<sup>th</sup> interceptor is assigned to the 3<sup>rd</sup> city and so forth. Then all the information is converted to the 2<sup>nd</sup> line; three interceptors are assigned to the city 1, two interceptors are assigned to the city 2, one interceptor is allocated to the city 3 and the city 4 and 5 both will be protected with two interceptors.

### 3.2 Initial population and reproduction algorithm

Once the chromosome is defined, it is required to generate an initial population. The initial population is randomly generated. It is checked for its fitness. Then the roulette wheel method based on individual fitness is applied for selecting parents. The two point crossover, shown in figure 4, is executed as a reproduction process. For the mutation of the gene, the random solution change operation is applied in order to select the gene which will be mutated. The city for assigning the selected interceptor in the gene is also randomly generated. The mutation process is demonstrated in figure 5; the 5<sup>th</sup> gene of the child 1 is randomly selected, and it



<Figure 5> Random mutation algorithm

is changed to 3 by random number generation algorithm. Data in table 2 summarize the result of offspring reproduction operations.

<Table 2> Solutions generated by crossover and mutation

City	Parent 1	Parent 2	Child 1	Child 2
1	3	2	3	2
2	2	3	3	2
3	1	1	1	2
4	2	2	1	3
5	2	2	2	1

### 3.3 Local search based on TS algorithm

After picking up the elitist in each generation, TS is implemented for local search. Since the order of gene in the elite chromosome does not make any difference to the solution in this problem, TS algorithm is executed after converting the solution sets. In order to execute TS, the whole solution set was defined as a tabu. Movement operator for this algorithm is swapping interceptors assigned in the two cities. In that case, 105 swaps are performed in 15 city problem, 300 swaps for 25 city and 780 swaps for 40 city problem are required. After completing the local search, all the solutions that were

generated during TS procedures are immediately killed except the one with the best fitness. It is recorded for the best solution of the generation.

## 4. Computational result

In this chapter, three case studies are introduced in order to demonstrate the use of suggested hybrid GA; 15 city - 100 interceptors, 25 city - 300 interceptors and 40 city - 500 interceptors. The strategic value of each city is randomly assigned in this problem shown in appendix 1; it will come from Intelligence Department. A GA and Hybrid GA are separately run to check the efficiency of the suggested method. Then, the best solutions of two cases are observed.

Populations of 200 individuals in every generation are applied in order to solve the problem. The program is terminated after producing 1000 generations for all the three cases. Previous solution set is defined as the Tabu list; only one Tabu list is set for the local search. The local search is conducted 10 iterations for each trial, and the best solution is recorded. The proposed algorithms were implemented in Visual C++ and executed on a Pentium 4, 2.2GHz computer.

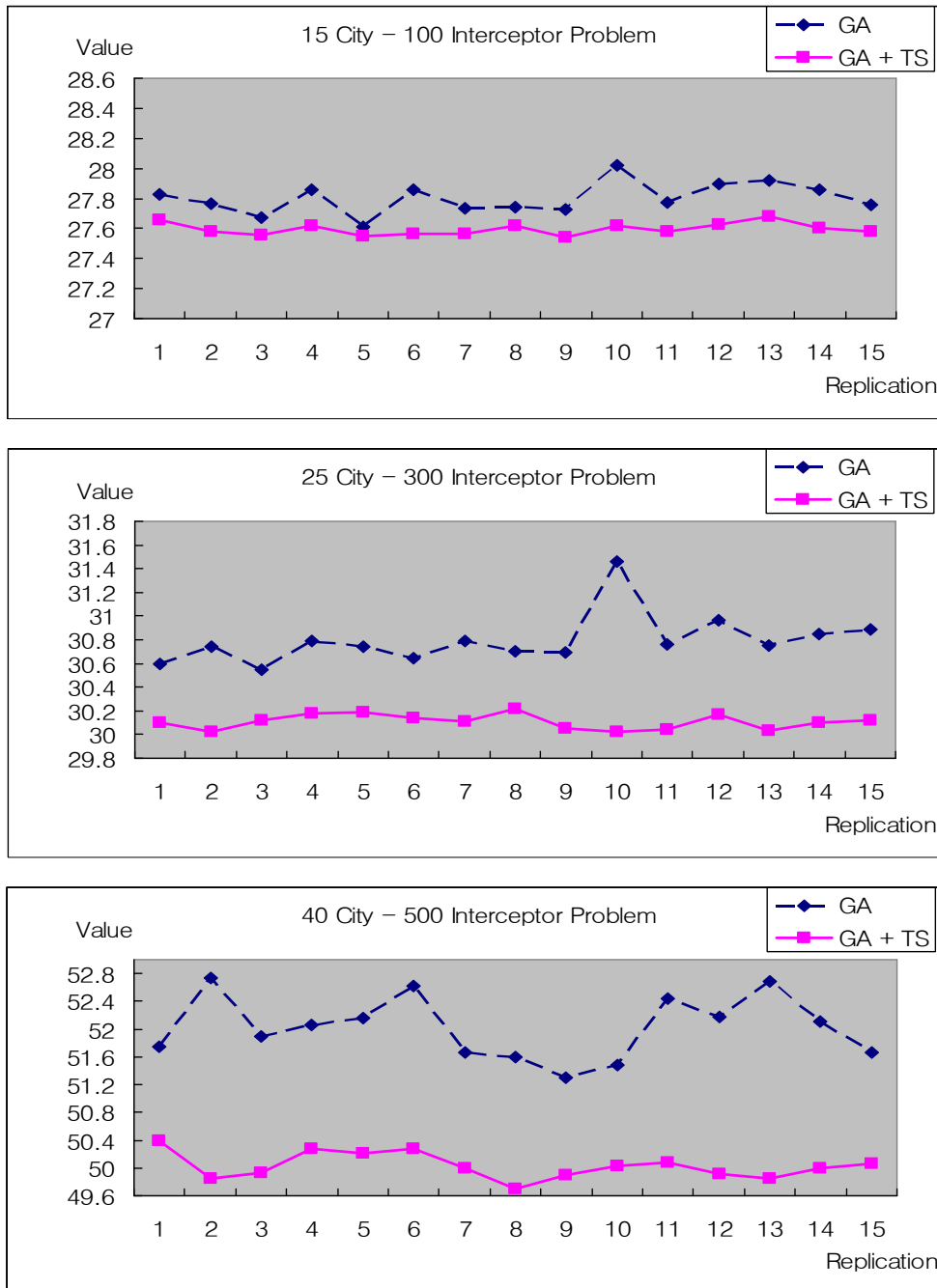
<Table 3> Computational results of GA & Hybrid GA

Problem Size	Heuristic	Best	Worst	Average	St. dev.
15 -100	GA	27.6135	28.0217	27.8015	0.104
	GA + TS	27.5444	27.6786	27.5966	0.04
25 -300	GA	30.5477	31.4582	30.7947	0.212
	GA + TS	30.0215	30.2219	30.1067	0.064
40 -500	GA	51.3007	52.7348	52.0205	0.453
	GA + TS	49.701	50.391	50.029	0.19

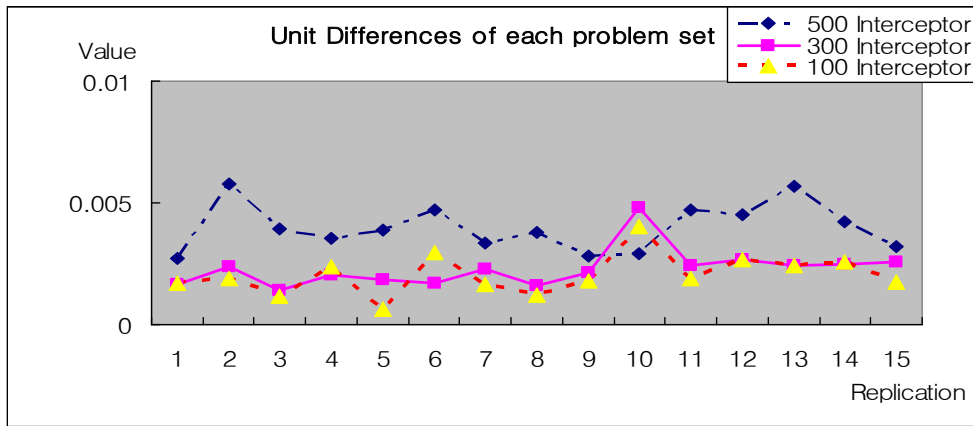


Table 3 summarizes the computational results of the three case studies. Table 3 contains the best, worse, average solutions and standard deviation

of 15 replications of GA and Hybrid GA. The results in table 3 demonstrates that the Hybrid GA could have a better solution than



<Figure 6> Effects of local search on each problem set



<Figure 7> Unit increase performed by local search for each problem set

canonical GA in all three demo problems.

Figure 6 illustrates the computational results of the three case studies. Through the figures, it is possible to compare the computational result of each pair of GA and Hybrid GA which have the same random seeds for each problem. The result demonstrates that Hybrid GA have better solution than canonical GA at all replications as well. It shows that the fitness is increased to all three problems by plugging TS in GA for the local search framework.

Figure 7 is the summary of unit increases by local search in each replication for three different problem sets. The important aspect from the figure is that the effect of the local search is more efficient to the large size problem. Since military has more interceptors than the example problems, the idea of implanting local search into canonical GA is justified by the computational result.

## 5. Conclusion

Heuristic, the present computer technology, has been a strong candidate for NP-hard or

NP-complete problems. Military operations research problems are too large to solve through traditional OR methodologies as well. This research illustrates a computational method using heuristic for the resource allocation problems in military. This study briefly demonstrates that a heuristic is a proper suit for the complex resource allocation problem. This study introduces an application of Hybrid GA to a real military resource allocation problem. In the algorithm, TS based local search method is performed to promote fitness for the elitist of each generation.

For the future study suggestion, other heuristic methods such as Ant Colony Optimization(ACO) and Particle Swarm Optimization(PSO) can be applied to get the better solution. To make this problem more realistic, it is needed to consider successful missile target hitting rate or adding up the different types of interceptors which has different intercept rates.

## Reference

- [1] Bazaara, M.S. and Elshafei, A.N., 1979, An Exact Branch-and-Bound Procedure for

- Quadratic Assignment Problem, *Naval Research Logistics Quarterly* 27 Vol.1, pp.109-120.
- [2] Heffley, D.R., 1972. The Quadratic Assignment Problem: A Note. *Econometrica*, Vol.40, No.6, pp.1155-1163.
- [3] Jewell, W.S., 1977, The Analytic Methods of Operations Research, *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, Vol. 287, No.1346, pp.373-404.
- [4] Kim, C. Y., 2000, *Nonlinear Programming*, Du-Nam Publications, Seoul, Korea.
- [5] Koopmans, T.C. and Beckmann, M.J., 1957, Assignment Problems and the Location of Economic Activities, *Econometrica* 25, pp.53-76.
- [6] Lawler, E.L., 1963, The Quadratic Assignment Problem, *Management Science* 9, pp.586-599.
- [7] Michalewicz, Zbigniew, 1996, *Genetic Algorithms + Data Structures = Evolution Programs*, 3<sup>rd</sup> Edition, Springer - Verlag Berlin Heidelberg, New York.
- [8] Reeves, C.R., 1993, *Modern Heuristic Techniques for Combinatorial Problems*, Blackwell Scientific Publications, London, UK.
- [9] Taillard, E., 1991, Robust taboo search for the quadratic assignment problem, *Parallel Computing* 19, pp.443-455.
- [10] Tate, D.M. and Smith, A.E., 1995, A Genetic Approach to The Quadratic Assignment Problem, *Computers Operations Research*, Vol.22, No.1, pp.73-83.
- [11] Wilhelm, M.R. and Ward, T.L., 1987, Solving Quadratic Assignment Problems by 'Simulated Annealing', *IIE Transactions*, Vol.19, March, pp.107-119.

## 저 자 소 개

한 현 진(E-mail: hanhj416@gmail.com)

1992 육군사관학교 졸업(학사)  
 2001국방대학교 운영분석학과 졸업(석사)  
 2007미 Auburn Univ. 산업시스템공학과 졸업(박사)  
 현재육군 제3군수지원사령부 제1급양대장  
 관심분야경제성공학, 비용분석, 의사결정, OR/SA

## Appendix 1. Acquired values for cities in problem sets

### 1. 15 city problem

City	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Value	8	5	15	7	16	15	8	9	6	32	30	25	21	16	14

### 2. 25 city problem

City	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Value	7	9	21	3	8	11	12	9	20	18	17	15	11	14	10
City	16	17	18	19	20	21	22	23	24	25					
Value	17	32	31	29	13	12	25	24	19	26					

### 3. 40 city problem

City	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Value	7	9	21	3	8	11	12	9	20	18	17	15	11	14	10	17	32	31	29	13
City	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
Value	23	22	21	18	14	11	29	9	4	17	28	25	15	33	6	20	10	24	32	12

## Appendix 2. Best interceptor allocating strategies for each problem set.

### 1. 15 city problem

City	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Interceptors	5	3	7	4	8	7	5	5	4	11	10	10	8	7	6

### 2. 25 city problem

City	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Interceptors	8	9	15	6	8	10	11	9	15	13	12	11	9	11	9
City	16	17	18	19	20	21	22	23	24	25					
Interceptors	12	18	17	17	11	10	15	15	14	15					

### 3. 40 city problem

City	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Interceptors	8	9	15	6	8	10	11	9	15	13	12	11	9	11	9	12	18	17	17	11
City	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
Interceptors	14	14	14	14	11	10	17	9	5	12	16	16	12	18	8	14	9	15	18	10

### Appendix 3. GA and Hybrid GA running solutions for each problem set.

Run	15 - 100			25 - 300			40 - 500		
	GA	GA+TS	Unit Difference	GA	GA+TS	Unit Difference	GA	GA+TS	Unit Difference
1	27.8293	27.6571	0.0017	30.5985	30.0965	0.0017	51.7435	50.3912	0.0027
2	27.7686	27.5813	0.0019	30.7391	30.0254	0.0024	52.7348	49.8483	0.0058
3	27.6717	27.5574	0.0011	30.5477	30.1219	0.0014	51.8943	49.9274	0.0039
4	27.8562	27.617	0.0024	30.7919	30.1756	0.0021	52.0574	50.2838	0.0035
5	27.6135	27.5499	0.0006	30.7385	30.1874	0.0018	52.1509	50.2032	0.0039
6	27.8595	27.5642	0.0030	30.6455	30.1402	0.0017	52.6226	50.2719	0.0047
7	27.7309	27.5665	0.0016	30.7899	30.1071	0.0023	51.6582	49.9914	0.0033
8	27.7441	27.621	0.0012	30.7021	30.2219	0.0016	51.5932	49.7012	0.0038
9	27.7231	27.5444	0.0018	30.691	30.0489	0.0021	51.3007	49.9035	0.0028
10	28.0217	27.621	0.0040	31.4582	30.0215	0.0048	51.4879	50.0209	0.0029
11	27.7707	27.5816	0.0019	30.766	30.0389	0.0024	52.4325	50.0732	0.0047
12	27.8932	27.6241	0.0027	30.9647	30.1684	0.0027	52.1799	49.9111	2.0045
13	27.9202	27.6786	0.0024	30.7528	30.029	0.0024	52.6863	49.8418	0.0057
14	27.8607	27.6031	0.0026	30.8451	30.1009	0.0025	52.1007	49.9977	0.0042
15	27.7587	27.5816	0.0018	30.8898	30.1167	0.0026	51.6639	50.0669	0.0032
Average	27.8015	27.5966	0.0021	30.7947	30.1067	0.0023	52.0204	50.0289	0.0040