Moving Target Travelling Salesman Problem using Genetic Algorithm

Nitin S. Choubey
Professor & Head,
Department of Computer Engineering
SVKM's NMIMS, Shirpur, Dhule, Maharashtra, India

ABSTRACT
In this paper, the Moving Target Travelling Salesman Problem (MTTSP) is described. In MTTSP, several sites are required to be visited which are moving with constant velocity in different directions. The distance of the sites from origin, velocity and the angle of movement are known in advance. The goal is to find the fastest tour starting and ending at the origin which intercepts all the sites. The method implemented using genetic algorithm approach on the various data sets and the results are compared with greedy approach.

General Terms
Algorithm, complexity,

Keywords
Genetic Algorithm, Evolutionary Computation, Travelling Salesman Problem, Moving Target Travelling salesman problem, intercept, greedy method.

1. INTRODUCTION
The traveling salesman problem (TSP) was originated by the studies of two mathematicians Sir William Rowam Hamilton from Ireland and Thomas Penyngton Kirkman from Briton in the 18th century. The general form of the TSP is believed to be studied further Kalr Menger in Vienna and promoted by Hassler, Whitney & Merrill at Princeton[2]. A detailed description about the connection between Menger & Whitney, and the development of the TSP can be found in Schrijver, 2005[3].

Given a set of cities and the cost of travel (or distance) between each possible pairs, the objective of the TSP is to find the best possible way of visiting all the cities and returning to the starting point that minimize the travel cost (or travel distance)[1]. Given *n* is the number of cities to be visited, the total number of possible routes covering all cities can be given as a set of feasible solutions of the TSP and is given as $(n-1)!/2$ [4]. In the classical TSP, the sites to be visited are stationary[10]. Motivated by practical application, Helvig et al.[4] introduced a time dependent generalization of TSP, referred as Moving Target TSP, where a pursuer must intercept in minimum time a set of targets which move with constant velocities (e.g., when a supply ship resupplies patrolling boats or when an aircraft must intercept a number of mobile ground units). Helvig et. al.[4] defined the Moving-Target travelling salesperson problem (MTTSP) as

"Given a set $S = \{s_1, \ldots s_n\}$ of targets, each s_i moving at constant velocity v_i from an initial position p_i, and given a pursuer starting at the origin and having maximum speed $v>|v_i|$, find the fastest tour starting (and ending) at the origin, which intercepts all targets."

An instance of MTTSP with four targets to be visited is shown in the fig 1. The start and end position of the pursuer is shown by flag. Helvig et. Al. has shown that the problem is NP-Hard. Several variants of MTTSP are given in [5][6].

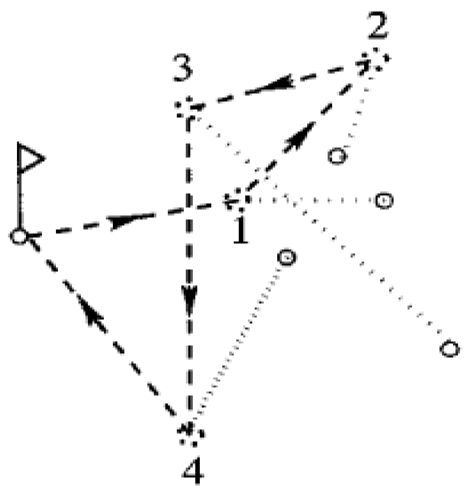

Fig 1 : MTTSP instance with four targets[4]

One of the current and best-known approaches for solving TSP problems is with the application of Evolutionary algorithms. These algorithms are based on naturally occurring phenomenon in nature, which are used to model computer algorithms. Potvin [11] has given the survey of Genetic Algorithm approaches for solving the general TSP. Most of evolutionary algorithm approaches try and find a solution to a particular problem, by recombining and mutating individuals in a society of possible solutions [6]. GA's are invented by John Holland in 1960s [7]. Holland's original goal was not to design algorithms to solve specific problems, but to formally study the phenomenon of adaptation as it occurs in nature and to develop ways in which the mechanisms of natural adaptation might be utilized into computer systems. Holland's 1975 book 'Adaptation in Natural and Artificial Systems' presented the GA as an abstraction of biological evolution and gave a theoretical framework for adaptation under the GA. Many problems in engineering and related areas require the simultaneous genetic optimization for a number of possibly competing objectives. These have been solved by combining the multiple objectives in to single scalar by the approach of linear combination. The combining coefficients, however, usually based on heuristic or guesswork can exert an unknown influence on the outcome of the optimization. A more satisfactory approach is to use the notion of Pareto optimality [8] in which an optimal set of solutions prescribe some surface 'The Pareto front' in the vector space of the objectives. For a solution on the Pareto front no objective can be improved without simultaneously degrading at least one other.

Jindal et al.[9][16] attempted to solve an variant of MTTSP, with resupply when all the target are moving away from the origin, by using greedy methodology, where authors sorted the target in the order of d_i/v_i to get the sequence of the sites to be visited, where d_i is the distance of the i^{th} target from origin and v_i is the velocity of i^{th} target. The assumption made in this paper is that, although the target are either moving away from the original position of the pursuer or moving towards pursuer, they are also moving with some angle "θ" with respect to the origin[13][14][15]. It is assumed that the angle of movement also plays important role in determining the sequence to be visited for the optimal elapsed time. The MTTSP problem is addressed in this paper with the help of Genetic Algorithm. The solution generated with the help of Genetic algorithm is compared with that of generated with the help of greedy approach. Section 2 of the paper gives details of the suggested approach and the greedy method. Section 3 compares the result followed by concluding remarks in section 4.

2. METHODOLOGY ADOPTED

The suggested approach is divided in to two sub-section namely The generation of the sequence and calculation of Total Elapsed Time (TET) the simulation of the sequence to get total elapsed time for the sequence.

2.1 Generation of sequence

The suggested model uses genetic algorithm for generating the sequence. The answer generated with genetic algorithm is compared with that of greedy method. The methods are discussed in the following subsections.

2.1.1 Genetic Algorithm methodology

The Simple Genetic Algorithm (SGA) is adopted for generating the optimal sequence. A Simple Genetic Algorithm is given in fig. 2.

1. Create initial random population.
2. Calculate fitness of the individuals in the population.
3. Repeat following steps until termination criteria is reached.
   a. Select best fit from the current population and generate offspring.
   b. Evaluate fitness of each offspring.
   c. Replace weak individuals from the population with newly generated ones.

Fig 2 : Simple Genetic Algorithm

The GA produces successive generations of individuals by computing their "fitness" at each step and selecting the best of them, when the termination condition arises. Tournament selection method is used for selection of parent where two parents are chosen randomly and the best amongst them get a chance to be in the mating pool [12]. The termination criteria of user specified maximum number of generations is used. Two point order based crossover and swap single bit mutation is used for reproduction process. The sequence of the targets to be visited by the pursuer is used as chromosome. The working of the crossover and mutation operators is shown in Fig 3. Order based crossover is used to preserve the target in the chromosomes. The TET is used as fitness value for the chromosome sequence.

2.1.2 Greedy Method

The sequence of the sites to be visited is decided based on the ratio of d_i/v_i and are scheduled in ascending and descending order of the ratio d_i/v_i.

Two Point Order Based Crossover (TPOBC)

P1	1	2	3	4	5	6	7	8	9
P2	9	8	7	6	5	4	3	2	1
Random Points 4 & 7				↑			↑		
Ch1	1	2	3	4	7	6	5	8	9
Ch2	9	8	7	6	3	4	5	2	1

Swap Single Bit Mutation (SSBM)

Child	1	2	3	4	5	6	7	8	9
Random Points (4,6)				↑		↑			
Mutated Child	1	2	3	6	5	4	7	8	9

Fig. 3 : Reproduction operators used

2.2 Calculation of Total Elapsed Time

The TET is measure by adding the time required to visit each targets by the pursuer in the provided sequence. The process is shown in figure 4.

1. Initialize the TET to zero, and target to visit as s_i with i=1.
2. Calculate the time required to visit the target s_i from origin of the pursuer, say t_i. Update TET to TET+ t_i, & i to i+1.
3. Calculate the time required to visit the target s_i from the place of the visit target s_{i-1} by the pursuer, say t_i.
4. Update TET to TET+ t_i, & i to i+1.
5. Repeat Steps 3 & 4 for all the targets in sequence.

Fig 4 : Calculation of TET

3. Experimental Setup and Result Analysis

The experiment is conducted with JDK 1.6 on an Intel Core™2 CPU with 2.66 GHZ and 2 GB RAM. The Population size =40, Maximum number of generation = 500, Crossover Rate = 0.8 and mutation rate = 0.05 is used for the purpose of simulation. Table 1, Table 2, Table 3 and Table 4 shows the four data sets used with four, six, eight and ten targets respectively for the experiment. The same data sets are used for Genetic Algorithm as well as for the greedy method. The results obtained after simulation are shown in Table 5, Table 6, Table 7 and Table 8 for the datasets with four, six, eight and ten targets respectively. The sample result for data set 10D3 is shown in fig 5. The initial position of the targets are shown by the red boxes. The path followed by the pursuer is shown by pink color whereas the track followed by the targets is shown by green color. The comparative chart for the total elapsed time taken by each method on various data sets is shown in fig 6. It is found that the time taken by the sequence generated by GA is much less than that of other methods. The result shows that the difference in the time taken for few targets is less as compared to that of more number of targets on various methods. The greedy methods does not take account of the of the target's direction of movement while deciding the sequence of the targets to be visited, where as in GA methods the final time taken by the pursuer is considered as a fitness function. It shows that there is need to incorporate the direction of targets movement apart from merely its direction and velocity only.

Table 1. Data Sets for Four targets

Target	Distance	Set 1, $4D_1$		Set 2, $4D_2$		Set 3, $4D_3$		Set 4, $4D_4$	
S_i	d_i	v_i	θ_i	v_i	θ_i	v_i	θ_i	v_i	θ_i
1	303	27	80	17	80	17	80	9	80
2	515	8	150	23	150	23	150	3	150
3	596	20	30	10	30	10	30	16	30
4	497	28	45	17	45	17	45	1	45

Table 2. Data Sets for Six targets

Target	Distance	Set 1, $6D_1$		Set 2, $6D_2$		Set 3, $6D_3$		Set 4, $6D_4$	
S_i	d_i	v_i	θ_i	v_i	θ_i	v_i	θ_i	v_i	θ_i
1	303	2	80	6	80	26	80	27	80
2	515	25	150	22	150	12	150	7	150
3	596	14	30	25	30	23	30	19	30
4	497	29	45	6	45	6	45	20	45
5	527	22	60	4	60	5	60	27	60
6	289	36	294	3	197	26	307	17	267

Table 3. Data Sets for Eight targets

Target	Distance	Set 1, $8D_1$		Set 2, $8D_2$		Set 3, $8D_3$		Set 4, $8D_4$	
S_i	d_i	v_i	θ_i	v_i	θ_i	v_i	θ_i	v_i	θ_i
1	303	16	80	4	80	8	80	14	80
2	515	26	150	3	150	17	150	28	150
3	596	27	30	4	30	9	30	17	30
4	497	1	45	11	45	7	45	22	45
5	527	23	60	29	60	3	60	18	60
6	326	37	296	15	85	39	30	19	122
7	419	26	136	30	168	4	32	23	14
8	281	10	104	13	334	7	42	17	28

Table 4. Data Sets for Ten targets

Target	Distance	Set 1, $10D_1$		Set 2, $10D_2$		Set 3, $10D_3$		Set 4, $10D_4$	
S_i	d_i	v_i	θ_i	v_i	θ_i	v_i	θ_i	v_i	θ_i
1	303	15	80	3	80	1	80	20	80
2	515	3	150	26	150	5	150	19	150
3	596	10	30	28	30	14	30	26	30
4	497	11	45	16	45	3	45	18	45
5	527	13	60	10	60	26	60	5	60
6	425	36	227	18	278	1	240	12	128
7	288	30	283	26	356	39	15	23	313
8	311	3	37	20	309	29	284	22	15
9	203	36	179	28	281	25	171	33	11
10	462	16	86	4	15	15	145	30	95

Table 5. Results for set with four targets

Data Sets	GA Method		Greedy Method			
			Di/vi, Ascending		Di/vi, Descending	
	Solution	TET	Solution	TET	Solution	TET
$4D_1$	1-2-3-4	24911	2-3-4-1	25821	1-4-3-2	26534
$4D_2$	1-4-2-3	24453	3-4-2-1	24860	1-2-4-3	24621
$4D_3$	1-4-2-3	26056	4-2-3-1	28349	1-3-2-4	27651
$4D_4$	1-2-3-4	24713	1-3-2-4	25824	4-2-3-1	26293

Table 6. Results for set with Six targets

Data Sets	GA Method		Greedy Method			
			Di/vi, Ascending		Di/vi, Descending	
	Solution	TET	Solution	TET	Solution	TET
$6D_1$	6-1-2-4-3-5	26458	6-4-2-5-3-1	30035	1-3-5-2-4-6	33685
$6D_2$	6-5-3-4-2-1	27945	2-3-1-4-6-5	43354	5-6-4-1-3-2	49389
$6D_3$	6-5-3-2-4-1	28024	1-6-3-2-4-5	36692	5-4-2-3-6-1	49461
$6D_4$	6-1-2-5-3-4	27919	1-5-4-6-3-2	48437	2-3-6-4-5-1	48527

Table 7. Results for set with eight targets

Data Sets	GA Method		Greedy Method			
			Di/vi, Ascending		Di/vi, Descending	
	Solution	TET	Solution	TET	Solution	TET
$8D_1$	6-5-3-2-4-7-1-8	29789	6-7-1-2-3-5-8-4	42326	4-8-5-3-2-1-7-6	61764
$8D_2$	8-7-4-3-5-2-6-1	28526	7-5-6-8-4-1-3-2	47136	2-3-1-4-8-6-5-7	60615
$8D_3$	8-6-5-3-2-4-7-1	26880	6-2-8-1-3-4-7-5	39875	5-7-4-3-1-8-2-6	46494
$8D_4$	8-6-5-3-4-2-7-1	29726	7-2-6-1-4-8-5-3	52419	3-5-8-4-1-6-2-7	64171

Table 8. Results for set with ten targets

Data Sets	GA Method		Greedy Method			
			Di/vi, Ascending		Di/vi, Descending	
	Solution	TET	Solution	TET	Solution	TET
$10D_1$	9-7-6-10-2-5-3-4-1-8	29862	9-7-6-1-10-5-4-3-8-2	45654	2-8-3-4-5-10-1-6-7-9	67522
$10D_2$	9-8-7-6-10-5-3-4-2-1	30878	9-7-8-2-3-6-4-5-10-1	42623	1-10-5-4-6-3-2-8-7-9	77921
$10D_3$	8-7-6-5-3-2-4-10-1-9	30076	7-9-8-10-5-3-2-4-1-6	44276	6-1-4-2-3-5-10-9-8-7	66117
$10D_4$	9-7-1-6-2-4-10-3-5-8	30929	9-8-1-10-7-3-2-4-6-5	54279	5-6-4-2-3-7-10-1-8-9	66543

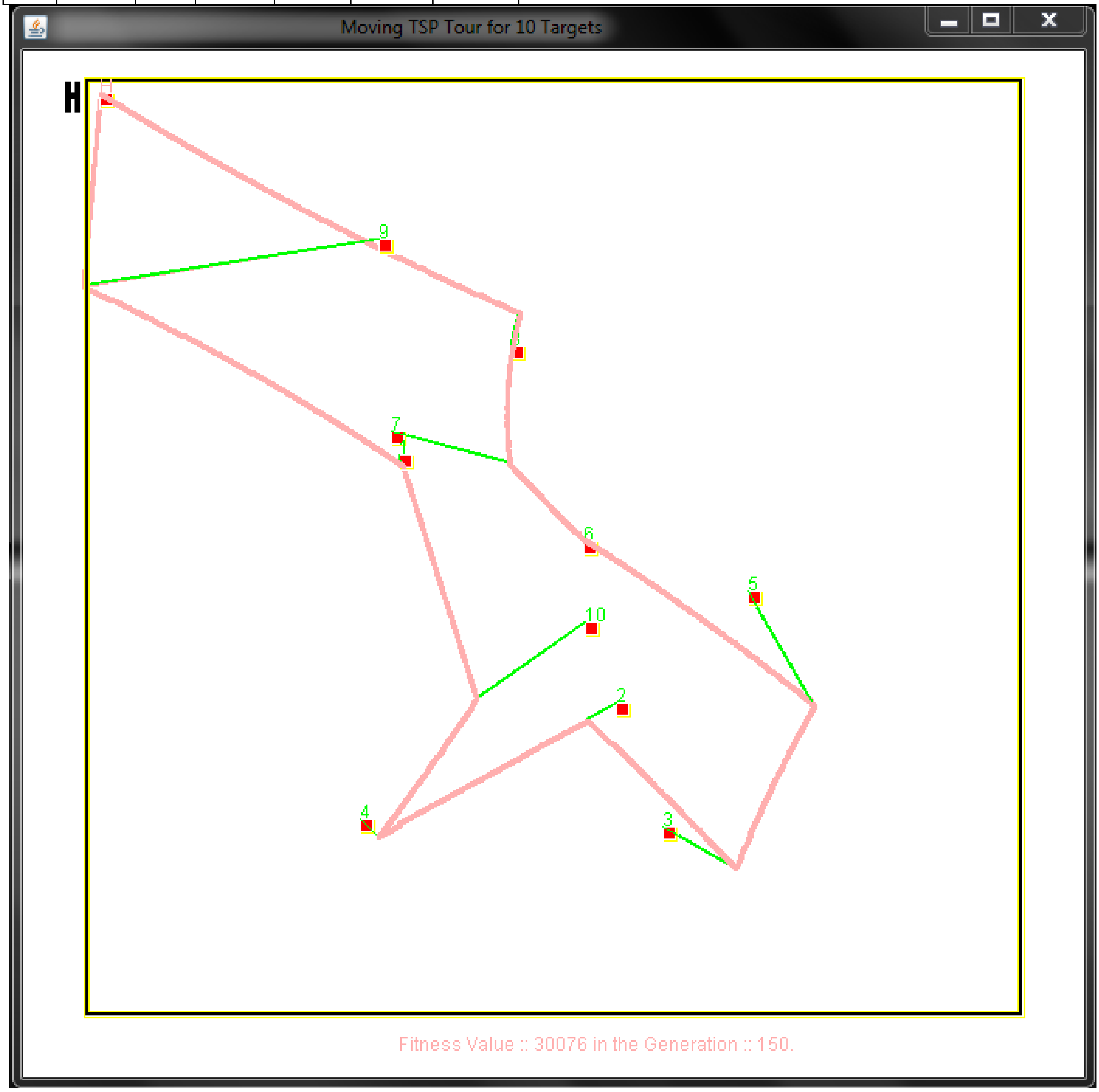

Fig 5 : Simulation Result for data set 10D3

4. CONCLUSION

In this paper, Moving Target TSP is studied. The method is implemented with Genetic Algorithm methods. The result of the experiment tested on the various datasets. The results generated by Genetic Algorithm method found to be more effective than generated by greedy method. There is need to find the effective solution by using analytical method for incorporating the target's direction of movement.

5. ACKNOWLEDGMENTS

Author thanks Mrs. Susheela Choubey and Prof. P. S. Gawande for their valuable time and contribution given while the drafting this paper. Author also thank to the reviewer for their expertise and keen review.

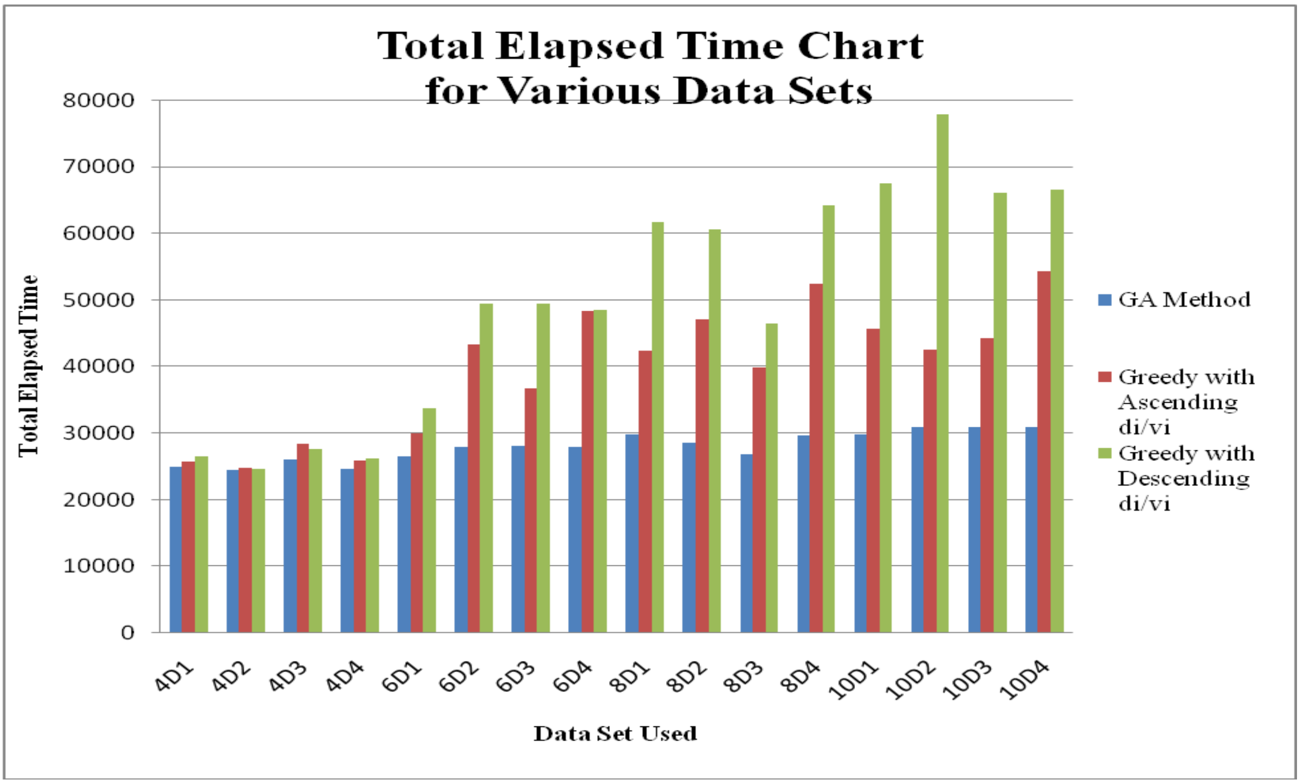

Fig 6 : Comparison chart

6. REFERENCES

[1] Horowitz E., Sahani S, and Rajasekaran S, 2007. Fundamentals of Computer Algorithm, University Press.

[2] Rajesh Matai, Surya Singh and Murari Lal Mittal (2010). Traveling Salesman Problem: an Overview of Applications, Formulations, and Solution Approaches, Traveling Salesman Problem, Theory and Applications, Prof. Donald Davendra (Ed.), ISBN: 978-953-307-426-9, InTech, Available from: http://www.intechopen.com/books/traveling-salesman-problem-theory-and-applications/traveling-salesmanproblem an-overview-of-applications-formulations-and-solution-approaches.

[3] Schrijver, A., 2005. On the History of Combinatorial Optimization (Till 1960). In K. Aardal, G. L. Nemhauser, & R. Weismantel (Eds.), Handbooks in operations research and management science, 12, Amsterdam: Elsevier.

[4] Helvig C. S., Gabriel Robins, and Alex Zelikovsky, 2003. The Moving-target travelling salseman problem, Journal of Algorithms 49 (2003), 153-174.

[5] Helvig C. S. Gabriel Robins, and Alex Zelikovsky, C. H. Helvig, G. Robins, and A. Zelikovsky. “Moving Target TSP and Related Problems”, G. Bilardi et Al. (Eds.): ESA’98, LNCS 1461, pp 435-1998.

[6] D. P. Guy, Evolutionary Computing as a Tool for Grammar Development, CNTS – Language Technology Group, UIA – University of Antwerp, Antwerp – Belgium, GECCO 2003 LNCS 2723, Springer-Verlag, Berlin, Heidelberg, 2003, pp. 549–560.

[7] J. H. Holland, Adoption in Natural and Artificial System, University of Michigan Press, Ann Arbor, MI, 1975.

[8] D. E. Goldberg, Genetic Algorithms in Search, Optimizations and Machine Learn-ing, Addison-Wesley, 1989.

[9] Jindal Pawan, Amit Kumar, and Shishir Kumar, 2011, “Dynamic Version of Travelling Salseperson Problem”, International Journal of Computer Applications (0975 – 8887), Volume 19– No.1, April 2011.

[10] Choubey Nitin S., 2012, “Solving TSP using DARO”, International Journal of Computer Applications (0975-8887), Volume 58-No.9, November 2012.

[11] Potvin J. Y., 1996, “Geneti Algorithms for the Travelling Salseman Problem”, Annals of Operation Research, 339-370, 1996.

[12] Sivanandam, Deepa, 2008, “Introduction to Genetic Algorithm”, Springer, 2008.

[13] Bartley Cristopher, 2004, “Closed Form Guidance Laws for Intercepting Moving Targets” Master’s thesis, Rice University, Houstons, Texas, April 2004.

[14] Gabrial Jacob Diaz, Flip Phillips and Brett R. Fajen, 2009, “Intercepting Moving Targets: a Little Foresight Helps a Lot” Springer-Verlag 2009.

[15] Jason R. Looker, 2008, “Minimum Paths to Interception of a Moving Targets when Constrained by Turning Radius”, DSTO Defence Science and Technology Organisation, Australia, December 2008.

[16] Jindal Pawan, Amit Kumar, and Shishir Kumar, 2011, “Multiple Target Intercepting Travelling Salseperson Problem”, International Journal of Computer Science and Technology, (0976 – 8491), Volume 2– Issue 2, June 2011.