

Iterated Local Search in Combinatorial Optimization Problem

Mehedi Hasan

Computer Science & Engineering, Stamford University Bangladesh, Dhaka, Bangladesh
mhj.cse@gmail.com

ABSTRACT

Iterated local search (ILS) is a very powerful optimization method for continuous-valued numerical optimization. However, ILS has seldom been used to solve combinatorial integer-valued optimization problems. In this paper, the iterated local search (ILS) with random restarts algorithm is applied to solve combinatorial optimization problems, e.g., the classical weapon-target allocation (WTA) problem which arises from the military operations research. The mathematical model of the WTA problem is explained in detail. Then the idea of ILS with random restarts is explained. A comparison of the algorithm with several existing search approaches shows that the ILS outperforms its competitors on the tested WTA problem.

Keywords: Iterated Local Search, WTA, Heuristic Solution, Combinatorial Optimization, Integer Programming

I. INTRODUCTION

Stochastic resource allocation (SRA) is a typical combinatorial optimization problem in complex systems. The decision-making about these problems depends on probability of stochastic events. For example, the classical weapon-target allocation (WTA) problem in military operations research is a typical example of SRA problems [1]-[3]. WTA has been proved to be a NP-Complete problem [4]. In this paper, based on a recent research on WTA, we applied the iterated local search approach for solving combinatorial SRA problems. The novelty of this paper is reflected by the incorporation of random restarts into ILS in favor of a better trade-off between exploration and exploitation abilities of the optimizer as well as the general SRA problem solving.

The Weapon Target Assignment (WTA) problem is a fundamental problem in the defense-related

applications. The problem is to set proper assignment of weapons to the threats such that overall expected damage of opponents is maximized. These weapon systems may have non-overlapping engagement zones and most importantly different kill probabilities. In addition, the performance of weapon system may vary according to the position and direction of threat as well as its type. The weapon-target assignment problem is an integer programming problem and known to be a typical NP-Complete problem [4]. There are some global optimum solutions to the particular assignment problem such as cutting plane techniques and branch and bound algorithms. Recently, several branch and bound algorithms with various bounding strategies are suggested for the WTA problem [5]. However, the exact solutions face with exponential computational complexities as the problem size is large. Recently, Rosenberger et.al. [6] uses auction algorithm to solve the WTA problem. Yet, this greedy approach is satisfactory only for small scale problems.

Genetic algorithms have been considered as good alternatives for optimization problems. Recently, a genetic solution with eugenic process has been proposed for weapon target assignment problems [7]. But the search efficiency still could be improved by performing reformation considering assignment pair's weight in overall cost instead of only individual cost. In this work, the ILS algorithm is applied for the first time to overcome those problems.

The paper is organized as follows. In Section 2, various literatures related to the WTA and combinatorial optimizations are reviewed. In Section 3, the WTA problem is defined and introduced. The idea and algorithm of Iterated Local Search are described in Section 4. The results of employing the proposed algorithm to solve WTA problem are presented in Section 5. Finally, Section 6 concludes the paper.

II. LITERATURE REVIEW

The study of WTA problem can be traced back to the 1950s and 1960s when Manne [8] and Day [9] built the model of WTA problem. Hosein and Athans [10] classified the WTA problem into two classes: the single-objective WTA problem and the multiple-objective WTA problem. Genetic algorithm [11], ACO algorithm [12], auction algorithm [13], VLSN algorithm [14], Tabu search [15], and other hybrid algorithms [16–18] have been used to optimize single-objective WTA model by many scholars. In contrast to single-objective WTA, multiple-objective optimization can take different criteria into consideration and is more in line with the real combat decision-making.

So, it has aroused wide attention from scholars. Liu et al. [19] proposed an improved multi-objective particle swarm optimization (MOPSO) algorithm to solve multiple-objective WTA problem and apply it to a simple

example including 7 platforms and 10 targets. However, they did not apply the proposed algorithm to different scales WTA problems for contrastive study. Zhang et al. [20] designed a WTA mathematic model and proposed a decomposition based evolutionary multi-objective optimization algorithm. But the algorithm has not been tested on large-scale WTA problem and it has a low convergence speed. Li et al. [21] adopted NSGA-II (domination-based) and MOEA/D (decomposition-based) to solve the multiple-objective WTA problem and carried out tests on three different BOWTA problems. They only applied the proposed adaptive mechanism to the WTA problems, but they did not verify the behaviour of the proposed adaptive mechanism on standard problems.

Different from other algorithms, ACO algorithm is a class of reactive search optimization (RSO) methods adopting the principle of “learning while optimizing” [22, 23]. Since it was introduced in 1992, many variant ACO algorithms have been presented, including ant colony system (ACS) [24] and MAXMIN ant system (MMAS) [25]. Meanwhile, ACO algorithms have been intensively investigated and successfully applied to deal with multi-objective problems such as travelling salesmen problem (TSP) [26], scheduling problem [27], vehicle routing problem [28], portfolio selection problem [29], network optimization problem [30], and some others problems [31–33]. In 2004, Doerner et al. proposed the P-ACO algorithm [29] which combined traditional ant system with ant colony optimization. But ILS has the ability to balance local search and global search, it will avoid premature convergence during the solution construction phase. The characteristic is especially suitable for solving portfolio selection problem. Therefore, ILS has been chosen with random restarts as the approach for solving the static WTA problem. However, to the best of my knowledge, there has not been any research conducted about ILS for the static WTA problem.

III. PROBLEM FORMULATION

The WTA problem focuses on how to allocate attack units to targets and can be illustrated in Figure 1. This section describes the integer programming formulation of assignment problem for N threats or targets and M weapons. The basic weapon target assignment problem considers only one to-one optimal assignment of weapons to targets to make the maximum damage or minimize the living probability of opponents. In our work, we extend the particular weapon target assignment problem with the following assumptions: The first one is that the number of weapons and threats are not necessarily equal to each other. Second, it is not mandatory to assign all weapons to threats. This assumption makes search space wide but increases the diversity of the solution. The third assumption is that a weapon system may run out of resource, i.e. missiles or bullets. In this case the inoperable weapon can't be assigned to a threat. The last assumption is that individual kill probability of a weapon for a given threat is somehow known. Kill probability of a weapon, also a measure of effectiveness of the weapon, is expected to include all aspects of engagement such as threat type, threat range, threat sector, weapon state.

Let $PK_{ij} \in [0,1]$ is the known kill probability for each threat $i = 1, \dots, N$ and weapon $j = 1, \dots, M$ pair. Let $V_i \in [0,1]$ is the estimated lethality value for each threat $i = 1, \dots, N$. Now, introduce the decision value $P_{ij} \in \{0,1\}$ which is indicating whether j^{th} weapon is assigned to i^{th} target for $i = 1, \dots, N$ and $j = 1, \dots, M$.

Assume that, the cost of an assignment decision, P , for a given scenario, PK and V , is the sum of weighted cumulative probability of survival of the target set as follows:

$$C(P) = \sum_{i=1}^N V_i \sum_{j=1}^M (1 - PK_{ij})^{P_{ij}}$$

Thus, we may define the combinatorial weapon assignment problem (WTA) as the following integer programming problem with the nonlinear objective (1):

minimize

$$C(P) = \sum_{i=1}^N V_i \sum_{j=1}^M (1 - PK_{ij})^{P_{ij}} \quad (2)$$

such that

$$\begin{aligned} \sum_{i=1}^N P_{ij} &< 1 \\ P_{ij} &\in \{0,1\} \\ i &= 1, \dots, N; j = 1, \dots, M \end{aligned}$$

where the first constraint is the fact that weapon can be allocated at most one target.

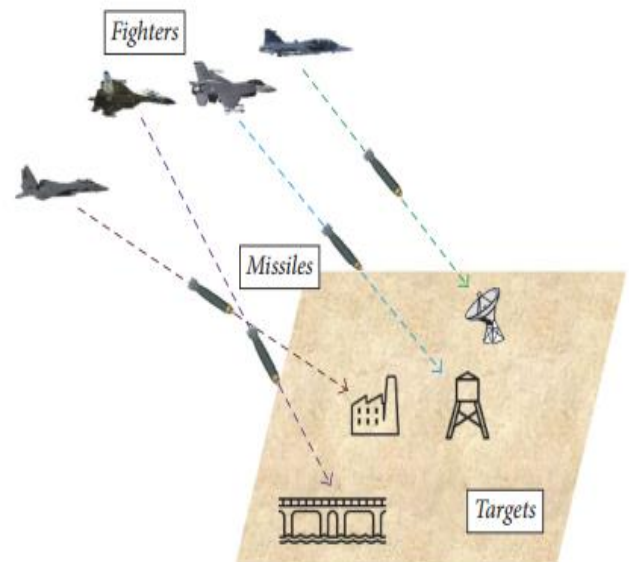


Figure 1: Illustration of the WTA problem

IV. ITERATED LOCAL SEARCH IN COMBINATORIAL OPTIMIZATION

This is the present name for a concept which has been around, in many guises, since at least the 1980s. It's essentially a more clever version of Hill-Climbing with Random Restarts. Each time you do a random restart, the hill-climber then winds up in some (possibly new) local optimum. Thus, we can think of Hill-Climbing with Random Restarts as doing a sort of random search through the space of local optima. We find a random local optimum, then another, then

another, and so on, and eventually return the best optimum we ever discovered. (ideally, It is a global optimum) Iterated Local Search (ILS) tries to search through this space of local optima in a more intelligent fashion: it tries to stochastically hill-climb in the space of local optima. That is, ILS finds a local optimum, then looks for a “nearby” local optimum and possibly adopts that one instead, then finds a new “nearby” local optimum, and so on. The heuristic here is that you can often find better local optima near to the one you’re presently in, and walking from local optimum to local optimum in this way often outperforms just trying new locations entirely at random.

ILS pulls this off with two tricks. First, ILS doesn’t pick new restart locations entirely at random. Rather, it maintains a “home base” local optimum of sorts, and selects new restart locations that are somewhat, though not excessively, in the vicinity of the “home base” local optimum. We want to restart far enough away from our current home base to wind up in a new local optimum, but not so far as to be picking new restart locations essentially at random. We want to be doing a walk rather than a random search.

Second, when ILS discovers a new local optimum, it decides whether to retain the current “home base” local optimum, or to adopt the new local optimum as the “home base”. If we always pick the new local optimum, we’re doing a random walk (a sort of meta-exploration). If we only pick the new local optimum if it’s better than our current one, we’re doing hill-climbing (a sort of meta-exploitation). ILS often picks something in-between the two, as discussed later. If you abstract these two tricks, ILS is very simple. The only complexity lies in determining when a local optimum has been discovered. Since this is often difficult, I will instead employ the same approach here as was used in random restarts: to set a timer. Hill-climb for a while, and then when timer goes off, it’s time to restart. This obviously doesn’t guarantee that we’ve found the local

optimum while hill-climbing, but if the timer is long enough, we’re likely to be in the vicinity.

The algorithm is very straightforward: do hill-climbing for a while; then (when time is up) determine whether to adopt the newly discovered local optimum or to retain the current “home base” one (the NewHomeBase function); then from our new home base, make a very big Tweak (the Perturb function), which is ideally just large enough to likely jump to a new hill. The algorithm looks like this:

ALGORITHM

Algorithm: Iterated Local Search (ILS) with Random Restarts	
1:	T ← distribution of possible time intervals
2:	S ← some initial random candidate solution
3:	H ← S
4:	Best ← S
5:	repeat
6:	time ← random time in the near future, chosen
	from T
7:	repeat
8:	R ← Tweak(Copy(S))
9:	if Quality(R) > Quality(S) then
10:	S ← R
11:	until S is the ideal solution, or time is up, or we
	have run out of total time
12:	if Quality(S) > Quality(Best) then
13:	Best ← S
14:	H ← NewHomeBase(H, S)
15:	S ← Perturb(H)
16:	until Best is the ideal solution or we have run out
	of total time
17:	return Best

Much of the thinking behind the choices of Perturb and NewHomeBase functions is a black art, determined largely by the nature of the particular problem being tackled. Here are some hints. The goal of the Perturb function is to make a very large Tweak, big enough to likely escape the current local optimum, but not so large as to be essentially a randomization. Remember that we'd like to fall onto a nearby hill. The meaning of "big enough" varies wildly from problem to problem. The goal of the NewHomeBase function is to intelligently pick new starting locations.

Just as global optimization algorithms in general lie between the extremes of exploration (random search and random walks) and exploitation (hill-climbing), the NewHomeBase should lie somewhere between these extremes when considering among local optima. At one extreme, the algorithm could always adopt the new local optimum, that is,

$$\text{NewHomeBase}(H, S) = S$$

This results in essentially a random walk from local optimum to local optimum. At the other extreme, the algorithm could only use the new local optimum if it's of equal or higher quality than the old one, that is,

$$\begin{aligned} \text{NewHomeBase}(H, S) &= S \\ &\text{if } \text{Quality}(S) \geq \text{Quality}(H) \\ \text{NewHomeBase}(H, S) &= H; \\ &\text{Otherwise} \end{aligned}$$

Tweak is adding a small amount of random noise to each number in the candidate solution vector. Tweak is to make a small, bounded, but random change. Tweak wouldn't ever make big changes. If we're stuck in a sufficiently broad local optimum, Tweak may not be strong enough to get us out of it. Thus, the algorithms so far have been local optimization algorithms. There are many ways to construct a global optimization algorithm instead. That's why we are changed the local search with random restart to overcome the stuck in local optima.

V. COMPUTATIONAL EXPERIMENTS

In this section, a numerical example that illustrates the merits of solving combinatorial WTA problem with ILS algorithm is presented. To validate the validity of the algorithm, simulation results are compared between the best solutions gained with standard Genetic Algorithm (GE) and improved Differential Evolution (DE) of the literature [34]. Basic data was provided in the literature [34], where there were 14 weapons and 12 targets. The killing probability of weapons was given in table II. The maximal update times of simulation with ILS, standard GE and improved DE for solving WTA problem was 200, and the swarm and initial random candidate solution scale were 200 too. The distribution of possible time intervals, T of ILS was 1 to 3. Simulation programs run 50 times to search the best solution. Simulation results is given in the table I. Table I shows that the ILS with random restarts is superior over standard GE & improved DE algorithm in the case of best & average value of convergent solution, average convergent time and times achieving the best solution when solving the WTA problem.

VI. CONCLUSION

In this article, the Iterated Local Search (ILS) algorithm is first time applied to solve the combinatorial stochastic WTA problem. A combinatorial WTA model satisfying expected damage probabilities is also formulated in this article. With this WTA model, the target with greater threat value can be intercepted first, and the armament can be saved. Perturb function is used to make a big enough tweak to escape the local optimum. The ILS with Random Restarts algorithm gives an effective way to solve this WTA problem. Finally, an initial simulation is demonstrated to solve the combinatorial problem. The

result of the simulation is compared with the Genetic Algorithm and improved Differential Evolution. The comparison shows that ILS is more effective and efficient than those algorithms in solving the combinatorial WTA problem.

TABLE I. Result Contrast

Algorithm	Best MinE	Avg. MinE	Avg. Convergence Time	Times of getting best solution
Standard GE	2.10	2.21	67	7
Improved DE	2.07	2.12	48	14
ILS with Random Restarts	2.01	2.06	39	5

TABLE III. Killing Probability

<i>j</i>	<i>i</i>											
	1	2	3	4	5	6	7	8	9	10	11	12
1	0.10	0.15	0.20	0.56	0.32	0.10	0.00	0.80	0.50	0.40	0.55	0.20
2	0.20	0.75	0.45	0.56	0.00	0.20	0.74	0.25	0.20	0.00	0.30	0.20
3	0.40	0.86	0.40	0.00	0.40	0.45	0.65	0.40	0.75	0.23	0.40	0.30
4	0.70	0.80	0.70	0.06	0.54	0.15	0.10	0.20	0.35	0.80	0.60	0.40
5	0.90	0.00	0.65	0.40	0.90	0.95	0.08	0.45	0.87	0.32	0.75	0.00
6	0.10	0.20	0.00	0.20	0.00	0.00	0.60	0.60	0.50	0.00	0.00	0.00
7	0.08	0.35	0.10	0.50	0.70	0.90	0.80	0.30	0.50	0.80	0.50	0.40
8	0.70	0.45	0.10	0.40	0.60	0.70	0.43	0.50	0.65	0.30	0.56	0.70
9	0.60	0.50	0.65	0.50	0.20	0.30	0.50	0.60	0.70	0.60	0.43	0.60
10	0.50	0.23	0.45	0.67	0.65	0.00	0.30	0.40	0.30	0.10	0.43	0.87
11	0.20	0.12	0.34	0.87	0.10	0.50	0.45	0.50	0.10	0.45	0.54	0.67
12	0.34	0.43	0.54	0.95	0.08	0.40	0.67	0.70	0.70	0.54	0.65	0.90
13	0.78	0.56	0.76	0.45	0.65	0.50	0.79	0.80	0.56	0.65	0.76	0.45
14	0.65	0.74	0.34	0.63	0.005	0.04	0.40	0.40	0.34	0.76	0.34	0.34

VII. REFERENCES

[1] A. S. Manne, A target-assignment problem, *Operations Research*, 6(1): 346-351, 1958.

[2] J. Chen, B. Xin, Z. H. Peng, et al, Evolutionary decision-makings for dynamic weapon-target assignment problem, *Science in China Series F: Information Sciences*, 52(11): 2006–2018, 2009.

[3] B. Xin, J. Chen, and Z. H. Peng, et al. An efficient rule-based constructive heuristic to solve dynamic weapon-target assignment problem, *IEEE Trans. on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 41(3): 598-606, 2011.

[4] S. P. Lloyd and H. S. Witsenhausen, “Weapon allocation is NP-Complete”, in *Proceedings of the IEEE Summer Simulation Conference*, 1986: 1054-1058.

[5] R. K. Ahuja, A. Kumar, K. Jha and J. B. Orlin, Exact and heuristic methods for the weapon

- target assignment problem: MIT Sloan Working Paper No: 4464-03, 2003.
- [6] J. M. Rosenberger and A. Yucel, "The generalized weapon target assignment problem," 10th Int. Conf. Com. and Cont. Res. and Techn. Symp, 2005.
- [7] Z.-J. Lee, S.-F. Su and C.-Y. Lee, "Efficiently solving general weapon-target assignment problem by genetic algorithms with genetic eugenics," *IEEE Trans. Syst. Man. Cybern.*, vol. 33, 2003.
- [8] A. S. Manne, "A target-assignment problem," *Operations Research*, vol. 6, pp. 346–351, 1958.
- [9] R. H. Day, "Allocating weapons to target complexes by means of nonlinear programming," *Operations Research*, vol. 14, no. 6, pp. 992–1013, 1966.
- [10] P. A. Hosen and M. Athans, *Preferential Defense Strategies. Part I: The Static Case*, MIT Laboratory for Information and Decision Systems, Cambridge, Mass, USA, 1990.
- [11] Z. R. Bogdanowicz, A. Tolano, K. Patel, and N. P. Coleman, "Optimization of weapon-target pairings based on kill probabilities," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1835–1844, 2013.
- [12] S. Chen, J. He, and H. Liu, "Realization and simulation of parallel ant colony algorithm to solve WTA problem," in *Proceedings of the International Conference on Systems and Informatics (ICSAI '12)*, May 2012.
- [13] A.-G. Fei, L.-Y. Zhang, and Q.-J. Ding, "Multi-aircraft cooperative free assignment based on auction algorithm," *Systems Engineering & Electronics*, vol. 34, no. 9, pp. 1829–1833, 2012.
- [14] M.-Z. Lee, "Constrained weapon-target assignment: enhanced very large-scale neighborhood search algorithm," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 40, no. 1, pp. 198–204, 2010.
- [15] D. G. Galati and M. A. Simaan, "Effectiveness of the Nash strategies in competitive multi-team target assignment problems," *IEEE Transactions on Aerospace & Electronics System*, vol. 43, no. 1, pp. 126–134, 2007.
- [16] Z.-J. Lee and W.-L. Lee, "A Hybrid search algorithm of ant colony optimization and genetic algorithm applied to weapon-target assignment problems," in *Intelligent Data Engineering and Automated Learning*, vol. 2690 of *Lecture Notes in Computer Science*, pp. 278–285, Springer, Berlin, Germany, 2003.
- [17] Y. Li and Y. Dong, "Weapon-target assignment based on simulated annealing and discrete particle swarm optimization in cooperative air combat," *Acta Aeronautica et Astronautica Sinica*, vol. 31, no. 3, pp. 626–631, 2010.
- [18] H.-D. Chen, S.-Z. Wang, and H.-Y. Wang, "Research of firepower assignment with multi-launcher and multi-weapon based on a hybrid particle swarm optimization," *Systems Engineering & Electronics*, vol. 30, no. 5, pp. 880–883, 2008.
- [19] X. Liu, Z. Liu, W.-S. Hou, and J.-H. Xu, "Improved MOPSO algorithm for multi-objective programming model of weapon-target assignment," *Systems Engineering & Electronics*, vol. 35, no. 2, pp. 326–330, 2013.
- [20] Y. Zhang, R.-N. Yang, J.-L. Zuo, and X.-N. Jing, "Weapon-target assignment based on decomposition-based evolutionary multi-objective optimization algorithms," *Systems Engineering & Electronics*, vol. 36, no. 12, pp. 2435–2441, 2014.
- [21] J. Li, J. Chen, B. Xin, and L. Dou, "Solving multi-objective multistage weapon target assignment problem via adaptive NSGAI and adaptive MOEA/D: A Comparison Study," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '15)*, pp. 3132–3139, May 2015.

- [22] R. Battiti, M. Brunato, and F. Mascia, "Reactive search and intelligent optimization," *Operations Research*, vol. 45, no. 1, pp. 74–89, 2008.
- [23] R. Battiti and G. Tecchiolli, "The reactive tabu search," *ORSA Journal on Computing*, vol. 6, no. 2, pp. 126–140, 1994.
- [24] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53–66, 1997.
- [25] T. Stutzle and H. H. Hoos, "Max-min ant system," *Future Generation Computer Systems*, vol. 16, no. 9, pp. 889–914, 1999.
- [26] S. Iredi, D. Merkle, and M. Middendorf, "Bi-criterion optimization with multi colony ant algorithms," in *Evolutionary Multi-criterion Optimization: First International Conference, EMO 2001 Zurich, Switzerland, March 7–9, 2001 Proceedings*, vol. 1993 of *Lecture Notes in Computer Science*, pp. 359–372, Springer, Berlin, Germany, 2001.
- [27] D. Merkle, M. Middendorf, and H. Schneck, "Ant colony optimization for resource-constrained project scheduling," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 4, pp. 333–346, 2002.
- [28] J. E. Bell and P. R. McMullen, "Ant colony optimization techniques for the vehicle routing problem," *Advanced Engineering Informatics*, vol. 18, no. 1, pp. 41–48, 2004.
- [29] K. Doerner, W. J. Gutjahr, R. F. Hartl, C. Strauss, and C. Stummer, "Pareto ant colony optimization: a metaheuristic approach to multi-objective portfolio selection," *Annals of Operations Research*, vol. 131, pp. 79–99, 2004.
- [30] P. Cardoso, M. Jesus, and A. Marquez, "MONACO-multi-objective network optimization based on an ACO," in *Proceedings of the X Encuentros de Geometria Computacional*, vol. 1, no. 1, pp. 1–10, Seville, Spain, 2013.
- [31] P. R. McMullen, "An ant colony optimization approach to addressing a JIT sequencing problem with multiple objectives," *Artificial Intelligence in Engineering*, vol. 15, no. 3, pp. 309–317, 2001.
- [32] T. Stutzle and M. Dorigo, "ACO algorithms for the quadratic assignment problem," *New Ideas in Optimization*, vol. 3, no. 1, pp. 33–50, 2000.
- [33] A. Baykasoglu, T. Dereli, and I. Sabuncu, "A multiple objective and colony ant colony optimization approach to assembly line balancing problems," in *Proceedings of the 35th International Conference on Computers and Industrial Engineering*, pp. 263–268, IEEE, Istanbul, Turkey, June 2005.
- [34] S. Wei-dong, Z. Cheng-wang and H. Jun-xiu, "Improved Differential Evolution Algorithm for Solving WTA Problem," *International Conference on Computer and Automation Engineering (ICCAE 2011), IPCSIT vol. 44*, 2012.

Cite this Article : Mehedi Hasan, "Iterated Local Search in Combinatorial Optimization Problem", *International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET)*, Online ISSN : 2394-4099, Print ISSN : 2395-1990, Volume 6 Issue 4, pp. 405-412, July-August 2019. Available at doi : <https://doi.org/10.32628/IJSRSET196455>
Journal URL : <http://ijsrset.com/IJSRSET196455>