International Journal Of Engineering Research & Management Technology

ISSN: 2348-4039

September- 2014 Volume 1, Issue-5

Email: editor@ijermt.org www.ijermt.org

A Survey of Metaheuristic Algorithms for Travelling Salesman Problem

Sudhanshu Prakash Tiwari
Research Scholar
Mewar University
Rajasthan

Sandeep Kumar
Assistant Professor
Lovely Professional University
Jalandhar Punjab

Dr.Kapil Kumar Bansal
Head Research & Publication
SRM University NCR Campus
Modinagar Ghaziabad

ABSTRACT-
Travelling salesman problem (TSP) is a one of the most popular real world combinatorial optimization problem in which we have to find a shortest possible tour that visits each city exactly once and come back to starting city. It ranges among NP hard problem so it is often used as a benchmark for optimization techniques. The solution to TSP problem has huge applications in various practical fields so it highly raises the need for an efficient solution. The solution of TSP cannot be finding traditionally specially when no. of cities in the problem increases. If one tries to solve, it will take years to find optimal solution of TSP using traditional method by considering all possible tours. In this case metaheuristic algorithm is the feasible and efficient solution. In this paper we consider widely used nature inspired metaheuristic approaches Ant Colony Optimization, Bee Colony, Cuckoo Search, Genetic Algorithm, Particle Swarm Optimization, Simulated Annealing and Tabu Search to solve TSP.

Keywords-Ant Colony optimization; Bee Colony; Cuckoo Search,; Simulated Annealing; Abu Search;

I. INTRODUCTION

The Travelling Salesman Problem (TSP) [1, 2, 6, 11] is widely studied problem in Computer Science in which task is to find a Hamiltonian path with minimum cost. The TSP has held the interests of computer scientists and mathematicians because, even after about half a decade of research, the problem has not been completely solved. The TSP can be applied to solve many practical problems such as logistics, transportation, semiconductor industries etc. Thus, a efficient solution to the TSP would ensure the tasks are carried out effectively and thus increase productivity. Due to its importance in many industries, TSP is still being studied by researchers from various disciplines.

The TSP [1, 2, 6] is known to be NP-hard. It means that no known algorithm is guaranteed to solve all TSP instances optimality within reasonable execution time. So in addition to find exact solution approaches, various heuristics and metaheuristics algorithms have been developed to solve problems approximately. They are used to find good quality solutions with-in reasonable execution times. Metaheuristics are normally improvement algorithms, i.e., they start with one or more feasible solutions to the problem at hand and suggest methods for improving such solutions. So to solve TSP problem researchers have proposed various meta heuristic approaches like Ant Colony Optimization (ACO), particle Swarm Optimization (PSO), Simulated Annealing (SA), Genetic Algorithm (GA). In this paper we'll discuss Ant Colony Optimization, Bee Colony, Cuckoo Search, Genetic Algorithm, Particle Swarm Optimization, Simulated Annealing and Tabu Search to solve TSP.

Travelling Salesman Problem

The problem was first defined in the 1800s by the Irish mathematician W.R. Hamilton and the British mathematician Thomas Kirkman but first formulated as a mathematical problem only in 1930 by Karl Menger.

Suppose given a set of cities and the distance between each possible pair, the Travelling Salesman Problem is to find the best possible way of visiting all the cities exactly once and returning to the starting city.

Generally, TSP can be denoted via graph notations as follows:
• Let G = (V, E) be a graph.
• V is a set of m cities, V = {$v_1 \ldots v_m$}.
• E is a set of edges, E = {(r, s): r, s ∈ V}.
E is normally associated with a distance (or cost) matrix, which is defined as D = (dr,s). If $d_{r,s} = d_{s,r}$ the problem is a symmetric TSP (STSP). Otherwise, it becomes an asymmetric TSP (ATSP). In TSP, the objective is to minimize the total roundtrip distance.

II. ANT COLONY OPTIMIZATION

Ant Colony Optimization (ACO) [3,4] was first introduced and applied to TSP by Marco Dorigo in 1991. ACO algorithm models the behaviour of real ant colonies in establishing the shortest path between food sources and nests. Ants can communicate with one another through chemicals called pheromones in their immediate environment. The ants release pheromone on the ground while walking from their nest to food and then go back to the nest. The ants move according to the amount of pheromones, the richer the pheromone trail on a path is, the more likely it would be followed by other ants. So a shorter path has a higher amount of pheromone in probability, ants will tend to choose a shorter path. Through this mechanism, ants will eventually find the shortest path.

Working of ACO for TSP
Initially, each ant is randomly put on a city. During the construction of a feasible solution, ants select the following city to be visited through a probabilistic decision rule. When an ant k states in city i and constructs the partial solution, the probability moving to the next city j neighbouring on city i is given by

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{u \in J_k(i)} [\tau_{iu}(t)]^{\alpha}[\eta_{iu}]^{\beta}} & if \ j \in J_k(i) \\ 0 & otherwise \end{cases}$$

Where τ_{ij} is the intensity of trails between edge (i.j) and ηij is the heuristic visibility of edge (i, j), and$\eta_{ij}=1/d_{ij}$. α Is pheromone influence factor, β is local node influence, and $J_k(i)$ is a set of cities which remain to be visited when the ant is at city i. After each ant completes its tour, the pheromone amount on each path will be adjusted with following equation.

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t)$$

where ρ is pheromone evaporation coefficient and

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t)$$

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} \dfrac{Q}{L_k}, & \text{if (i, j) } \in \text{ tour done by ant k} \\ 0 & \text{otherwise} \end{cases}$$

(1- ρ) is the pheromone decay parameter (0<ρ<1) where it represents the trail evaporation when the ant chooses a city and decide to move. Lk is the length of the tour performed by ant k and m is the number of ants. Q is pheromone deposit factor. Following is procedure of ACO algorithm for TSP:

procedure of ACO algorithm for TSP
Set parameters, initialize pheromone trails
while (termination condition not met) **do**
Construct Solutions
Apply Local Search % optional
Update Trails
end
end ACO algorithm for TSP

III. BEE COLONY OPTIMIZATION

Bee Colony Optimization (BCO) [7, 17] algorithms is defined by Dervis Karaboga in 2005, motivated by the intelligent behaviour of honey bees. The foraging behaviour in a bee colony remains mysterious for many years until von Frisch translated the language embedded in bee waggle dances. Waggle dance operates as a communication tool among bees. Suppose a bee found a rich food source. Upon its return to the hive, it starts to dance in a figure-eight pattern. This figure-eight dance consists of a straight waggle run followed by a turn to the right back to the starting point, and then another straight waggle run followed by a turn to the left and back to the starting point again. The bee usually repeats these for a few times. Remarkably, via this informative dance, the bee has actually informed its hive mates about the direction and distance of the food source. The direction is expressed via the angle of dance relative to the sun position whereas the distance is expressed via the length of the straight waggle run. This coding and decoding process will eventually bring more bees towards the new food discovery.

Working of Bee Colony Optimization for TSP
In our proposed model, a bee is allowed to explore and search for a complete tour path. Before leaving the hive, the bee will randomly observe dances performed by other bees. The bee is then equipped with an ordered set of moves which are observed from the dance. During the foraging process, a bee will travel from one city to another city until it reaches the destination. In the bee model, a heuristic transition rule is employed to aid the bee in its decision making on which city to visit next. This rule consists of two factors: arc fitness and heuristic distance. The arc fitness is computed for all possible paths to cities that can be visited by a bee from a particular city at a particular time. A higher fitness value is assigned to the arc which is part of the preferred path. By doing this, a bee tends to choose the next visiting city based on the preferred path. On the other hand, under the heuristic distance influence, a bee tends to choose the next visiting city which is nearest to its current city. State transition probability gives the likelihood to move from city i to city j after n transitions. It is a function of the distance between cities and of the arc fitness present on connecting edge.

$$P_{ij}(t) = \frac{[\rho_{ij}(t)^{\alpha} . [1/d_{ij}]^{\beta}}{\sum_{j \in Ai(t)} [\rho_{ij}(t)]^{\alpha} . [1/d_{ij}]^{\beta}}$$

Where ρ_{ij} (t) is the arc fitness from city I to city j after n transitions, d_{ij} represents the distance between city i and city j, α represents a binary variable that turns on or off the arc fitness influence in the model, and β is to control the significant level of heuristic distance. The general procedure that is followed by bee for solving TSP is shown below:

Procedure of BCO algorithm for TSP
Initialize Population ( )
while stop criteria are not fulfilled **do**
while all bees have not built a complete path **do**
Observe Dance ( )
Forage By Trans Rule ( )
Perform Waggle Dance ( )
end while
end while
end procedure

III. CUCKOO SEARCH

Cuckoo Search (CS) [8,9,10] is an optimization algorithm developed by Xin-she Yang and Suash Deb in 2009. It was inspired by the egg laying behaviour of cuckoo bird who lays their eggs in the nests of other host birds (of other species). Some host birds can engage direct conflict with the intruding cuckoos. If a host bird discovers the eggs are not their own, it will either throw these alien eggs away or simply abandon its nest and build a new nest elsewhere. Cuckoo search idealized such breeding behaviour, and thus can be applied for various optimization problems. It seems that it can outperform other metaheuristic algorithms in applications.

Each egg in a nest represents a solution, and a cuckoo egg represents a new solution. The aim is to use the new and potentially better solutions (cuckoos) to replace a not-so-good solution in the nests. CS is based on three idealized rules:

1. Each cuckoo lays one egg at a time, and dumps its egg in a randomly chosen nest;
2. The best nests with high quality of eggs will carry over to the next generation;
3. The number of available hosts nests is fixed, and the egg laid by a cuckoo is discovered by the host bird with a probability $p\epsilon(0,1)$. Discovering operate on some set of worst nests, and discovered solutions dumped from farther calculations.

Working of Cuckoo Search for TSP

A cuckoo flies between cities obeying Levy distribution. At the beginning, a cuckoo visited n cities once and saved its traved order in the bulletin. Individual is a nest which represents the first city (the end city) of a route (solution). Population is all n nests which represent n cities. A solution represents an available route. Bulletin board save all n solutions and refresh by offspring's generation by generation. In every generation, n cuckoos start from their nests flying around to get new nests replacing of the olds. By partially reversing of segment of route (start from old nest to the new nest) the algorithm eliminates crossover edges to find shorter route (better solution). Fitness defined as its route length: the smaller, the better.

The general procedure of Cuckoo Search Algorithm to solve Travelling Salesman Problem is shown below:

Procedure of Cuckoo Search Algorithm for TSP
Begin
Objective function f (p), city distances array;
Initial a population of n host nests (cities)) xi,
i = 1;2; · · · ;n;
Cuckoos fly via Levy flight to find an initial route (solution).
Mend initial solutions and saved in the bulletin board.
Evaluate the route length (fitness) of solutions Fi;
while (t < Max Generation) or (Stop Criterion)
Cuckoos start from their nest to search new nests;
if F'i<Fi
Replace old nest by new one,
Reverse the segment between old nest and new one,

Refresh bulletin board.
end if
Host birds abandon pa ∈ (0;1) nests, and search pa
new nests;
Refresh the bulletin board and keeping the best solutions
(and nests).
Rank the solutions, and find the current best route (solution).
t = t +1;
end while
Post process results and visualization.
end

IV. GENETIC ALGORITHMS

Genetic Algorithm (GA) [12,17] is one of the oldest and most successful optimization technique which is inspired by Charles Darwin's theory of evolution and natural selection. It was originally proposed by John Holland ain the 1960s at the University of Michigan. Genetic algorithm uses survival of the fittest approach for selecting the best (fittest) solution from the available solutions. This approach uses the population of Chromosomes as the starting point then each chromosome is tested against fitness using an appropriate fitness function. Then the best chromosomes are selected and they undergo process of crossover and mutation to create new set of chromosome.

Working of GA for TSP
The basic steps of genetic algorithm are given below:

A. Initialization

An initial population is created from a random selection of solutions (which are analogous to chromosomes).

B. Encoding

Encoding is a process of representing individual genes. The process can be performed using bits, numbers, trees, arrays, lists or any other objects. Encoding can be binary, octal, and hexadecimal.

C. Fitness Function

The Purpose of the fitness function is to decide if a chromosome is good then how it is good. In the travelling salesman problem the criteria for good chromosome is its length. Calculation takes place during the creation of the chromosomes as given in equation. Each chromosome is created and then its fitness function is calculated. The length of the chromosome is measured by the scheme of the tour.

$$\mathbf{Fitness_{cromosome}=\sum_{i=1}^{towncount} ti}$$

D. Selection

In Selection operation, the individuals with larger fitness value are chosen to produces Offspring. Selection is the stage of a genetic algorithm in which individual genomes are chosen from a population for later breeding (recombination or crossover).

E. Crossover

The crossover recombines two individuals to have new ones which might have a better performance. Following shows the single-point crossover of the binary-coded genetic algorithm:

101 | 101 →101 010
010 | 010 →010 101

IJERMT

F. Mutation

Some individuals are chosen randomly to be mutated and then a mutation point is randomly chosen. The character in the corresponding position of the string is changed. Once this is done, a new generation has been formed. Formula shows the mutation of the binary-coded genetic algorithm as follows.

101110→101010

G. Termination

When the algorithm has run a given number of iterations, it stops and output the best solution. This process is repeated until a termination condition has been reached .

Concluding following is the complete algorithm of GA for TSP:

Precedure of GA for TSP

for i=1 to #pop do [#pop=population]

begin

choose new string town s;

tour[i]=nearest neighbour(starting town=s);

end;

while evaluation **do**

begin

choose parent1 and parent2; [selection]

offspring=crossover(parent1, parent2); [crossover]

activate-edges(offspring, parent1) ;

optimize-local(offspring); [mutation]

if $\neg\exists$ i$\leq$ #pop {length(tour[i])-length(offspring) $\leq$ a} [survival of the fittest]

$\wedge\ \exists$ i$\leq$ #pop {length(tour[i]) $\geq$ length(offspring)}

then replace the longest tour by offspring

end;

V. PARTICLE SWARM OPTIMIZATION

Particle swarm optimization (PSO) [13,14,15] is a population based stochastic optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995, inspired by social behaviour of bird flocking or fish schooling. In PSO, the potential solutions, called particles which are initialized with a population of random solutions and searches for optima by updating generations, fly through the problem space by following the current optimum particles. Each particle keeps track of its coordinates in the problem space which are associated with the best solution (fitness) it has achieved so far. This value is called pBest. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the neighbors of the particle. This location is called lbest. When a particle takes all the population as its topological neighbors, the best value is a global best and is called gBest. The particle swarm optimization concept consists of, at each time step, changing the velocity of (accelerating) each particle toward its pBest and lBest locations (local version of PSO). Acceleration is weighted by a random term, with separate random numbers being generated for acceleration toward pBest and lBest locations.

Working of PSO for TSP

To solve TSP problem we must point two issues which are how to change the position of particle and how to guarantee the position is reasonable. Following is the complete procedure of solving TSP using PSO:

Procedure of PSO for TSP

/* Define initial probabilities for particles' moves:*/

pr1 ← a1 /*to follow its own way*/

pr2 ← a2 /*to go towards pbest*/

pr3 ← a3 /*to go towards gbest*/ /* a1+ a2+ a3=1 */

Initialize the population of particles
do
for each particle p
$value_p \leftarrow Evaluate(x_p)$
if ($value(x_p) < value(pbestp)$) then
$pbestp \leftarrow x_p$
if ($value(x_p) < value(gbest)$) then
$gbest \leftarrow x_p$
end for
for each particle p
$velocity_p \leftarrow$ define velocity(pr1, pr2, pr3)
$x_p \leftarrow update(x_p, velocity_p)$
end_for
/* Update probabilities*/
while (a stop criterion is not satisfied)

VI. SIMULATED ANNEALING

Simulated Annealing (SA) [16,17,18] method was described by Scott Kirkpatrick, C. Daniel Gelatt and Mario P. Vecchi in 1983. Simulated Annealing is the oldest probabilistic meta-heuristic algorithm used to find an approximate solution to global optimization problems. It is inspired by annealing in metallurgy which is a technique of controlled cooling of material to reduce defects. The simulated annealing algorithm begins with a random solution. Each iteration forms a random nearby solution. If this solution is a better solution, it will replace the current solution. If it is a worse solution, it may be chosen to replace the current solution with a probability that depends on the temperature parameter. As the algorithm goes, the temperature parameter decreases, giving worse solutions a lesser chance of replacing the current solution. Allowing worse solutions at the beginning helps to avoid converging to a local minimum rather than the global minimum. There are two variables that affect the result of the algorithm are the initial temperature, the rate at which the temperature falls and the stopping criteria of the algorithm.

Working of SA for TSP

In the Travelling Salesman Problem simulated annealing starts with an random initial tour. The objective function i.e. total length of tour) is analogous to the current energy state of the system. It changes from one state to another which produces a reduced tour length. This is analogous to slow cooling. Changes that increase the tour length are only accepted with probability $p(d,T)=exp(-d/T)$, where d is the change in the tour length and T is the temperature of the system. Temperature parameter controls the annealing process.
The algorithm for TSP using SA has two parts. Part 1 concern with the distribution of the tour among the processors and interacting among the cities with every processor. In part 2 some interconnection scheme is used to move or redistribute the cities of the tour among the processor. Complete procedure of SA for solving TSP is given as follows:

Procedure of SA for TSP
begin procedure SA
generate initial solutions
set temperature, and cooling rate
while(termination criteria not meet)
generate new solutions
access new solutions
if (accept new solution)
update storage
adjust temperature
end if

end while
post-process results and output
end

VII. TABU SEARCH

Tabu search (TS) [19,20,21] is developed by Fred W. Glover in 1986 and formalized in 1989 as a metaheuristic search method. It is a algorithm that can be used for solving combinatorial optimization problems (problems where an optimal ordering and selection of options is desired).. Local searches take a potential solution to a problem and check its immediate neighbours (that is, solutions that are similar except for one or two minor details) in the hope of finding an improved solution. Local search methods have a tendency to become stuck in suboptimal regions where many solutions are equally fit.

Woking of Tabu-Search for TSP

In this technique, we start with an arbitrary node and make a succession of moves to transform this permutation into an optimal one (or as close to the optimum as possible). In this particular setting, it is equivalent to starting with a randomly generated tour and making a succession of edge swaps trying to reduce the cost of the tour until we can find the minimum cost. Tabu search enhances the performance of these techniques by using memory structures that describe the visited solutions or user-provided sets of rules. If a potential solution has been previously visited within a certain short-term period or if it has violated a rule, it is marked as "tabu" (forbidden) so that the algorithm does not consider that possibility repeatedly. The procedure of TS for solving TSP is given as follows:

Procedure of TS for TSP
Generate a random dataset
Initialize population and clear the tabu list;
while (termination criteria not met)
Generate neighbours of the current seed solution by a neighbourhood structure.
If aspiration criteria is not satisfied.
Select the “best” neighbour which is not tabu is as new seed
Update the tabu list
end if.
else
Store the aspiration solution as the new seed and the best solution.
end else
end while
output optimization result

VIII. CONCLUSION

This paper presents a view of most widely used metaheuristic optimization algorithm techniques namely ACO, BCO, CS, GA, PSO, SA and TS optimization to solve the travelling salesman problem. In sections II to VIII we have discussed these metaheuristic approaches as they are described in the literature. ACO is the process used by ants to forage food source. They use pheromone trail deposition and evaporation technique to find the shortest path among all paths. BCO algorithm solves TSP based on the bees’ collective foraging and waggle dance behaviour. Cuckoo Search is a new metaheuristic algorithm that solves the TSP problem in combination with Levy flights which is based on the breeding strategy of some cuckoo species. CS is more generic and robust for many optimization problems, comparing with other metaheuristic algorithms. GA is an optimization technique, inspired by the law of biological reproduction and survival of fittest theory, where mutation and crossover operations used to find by the local optimal solution. PSO is a robust stochastic optimization technique based on the movement and intelligence of swarms. PSO solves TSP by accelerating each particle towards its pbest and the gbest locations, with a random weighted acceleration at each time. SA uses annealing process in metallurgy which is a technique of controlled cooling of materials to

reduce defects. The key to SA's ability to find global optimums in TSP is that at each iteration it has a decreasing probability of moving to less fit states which reduces total cost. The basic principle of TS to solve TSP is to pursue local search whenever it encounters a local optimum by allowing non-improving moves; cycling back to previously visited solutions is prevented by the use of memories, called tabu lists that record the recent history of the search. Tabu Search can work with a short time but sometime it can't found the best path and if it work with a large problem it just have low performance. All these techniques have immense potential and scope of application ranging from engineering to software engineering, and real world optimization problems. These techniques need to be further explored to find their suitability to certain applications. Also, there is a need to combine two or more techniques so that they complement each other and nullify their respective limitations.

ACKNOWLEDGEMENT

I thank all who have supported for this research.

REFERENCES

1. Rajesh Matai1. Traveling Salesman Problem: An Overview of Applications, Formulations, and Solution Approaches.Traveling Salesman Problem, Theory and Applications, 1-24 (2011).
2. M. Dorigo and L. M. Gambardella, The colony system: A cooperative learning approach to the traveling salesman problem, IEEE Transactions on Evolutionary Computation, Vol.1, No.1, April, 1997.
3. M. Dorigo, V. Maniezzo, and A. Colorni, The ant system: Optimization by a colony of cooperating agents, IEEE Transactions on System, Man, and Cybernetics, Part B, Vol.26, pp. 29-41, 1996.
4. M. Dorigo, M. Birattari, T. Stuzle, Ant Colony Optimization, Artificial Ants as a Computational Intelligence Technique, IEEE Computational Intelligence Magazine, November 2006
5. S. Nakrani and C. Tovey, "On honey bees and dynamic server allocation in Internet hosting centers," Adaptive Behavior, vol. 12, no. 3-4, pp. 223-240, 2004.
6. R. E. S. D. J. Rosenkrantz and P. M. Lewis, "An analysis of several heuristics for the traveling salesman problem," SIAM Journal on Computing, vol. 6, no. 3, pp. 563-581, 1977.
7. J. C. Biesmeijer and T. D. Seeley, "The use of waggle dance information by honey bees throughout their foraging careers," Behavioral Ecology and Sociobiology, vol. 59, no. 1, pp. 133-142, 2005.
8. Xin-She Yang, Suash Deb. Cuckoo Search via Lvy Flights,
9. Proceedings of World Congress on Nature & Biologically Inspired Computing (NaBIC 2009, India), IEEE Publications,210-214(2009).
10. Xin-She Yang, Suash Deb. Engineering Optimisation by Cuckoo Search. International Journal of Mathematical Modelling and Numerical Optimisation. 3, 330-343 (2010)
11. Xin-SheYang, Nature-Inspired Metaheuristic Algorithms,Luniver Press. 2010.
12. Johnson D.S. McGeoch L.A. The Traveling Salesman Problem: A Case Study in Local Optimization. Local Search in Combinatorial Optimization, 215-310 (1997).
13. Melanie Mitchell.,"An Introduction to Genetic Algorithms". MIT Press, Cambridge,MA, 1998.
14. J. Kennedy and R. C. Eberhart. Particle Swarm Optimization. In Proceedings of the IEEE International Conference on Neural Networks, pages 1942-1948,1995.
15. http://clerc.maurice.free.fr/pso/pso_tsp/Discrete_PSO_TSP.h
16. K.P. Wang, L. Huang, C.G. Zhou, W. Pang. Particle swarm optimization for traveling salesman problem. In Proceedings of the International Conference on Machine Learning and Cybernetics ,pages 1583–1585,2003
17. S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi. "Optimization bysimulated annealing". Science, Number 4598, vol. 220, 4598, pp. 671-680, 13 May 1983.
18. Binitha, S.S. Siva sathya, "A survey of bio inspired optimization algorithm", IJSCE, Vol-2, issue-2, May 2012
19. N. Sureja, B. Chawda, "Random Travelling Salesman Problem usingSA," International Journal of Emerging Technology and AdvancedEngineering, Volume 2, Issue 4, April 2012
20. Yonghui Fang, Guangyuan Liu, Yi He, and Yuhui Qiu, "Tabu search algorithm based on insertion method", in Proceedings of the 2003 International Conference on Neural Networks and Signal Processing, 2003, 2003, vol. 1, pp.
21. Tsubakitani and J. Evans, "Optimizing Tabu List Size for the Traveling Salesman Problem," Computers & Op-erations Research, Vol. 25, No. 2, 1998, pp. 91-97.
22. G. Barbarosoglu and D. Ozgur, "A Tabu Search Algo-rithm for the Vehicle Routing Problem," Computers and Operations Research, Vol. 26, No. 3, 1999, pp. 255-270

IJERMT