

Tabu Search Method for Solving the Traveling salesman Problem

Isra Natheer Alkallak
israalkallak19@gmail.com
College of Nursing

Ruqaya Zedan Sha'ban
nnabeel2013@gmail.com
College of Medicine

University of Mosul, Iraq

Received on: 10/12/2007

Accepted on: 04/03/2008

ABSTRACT

This research presents the fundamental concepts of tabu search for optimization problem as symmetric traveling salesman problem. The purpose of this research is to solve the traveling salesman problem with tabu search method, to find an optimal solution of small search space and computational requirements.

Keywords: Tabu algorithm, traveling salesman problem, NP problem.

طريقة البحث الممنوع لحل مسألة البائع المتجول

رقية زيدان شعبان

إسراء نذير الكلاك

كلية الطب، جامعة الموصل

كلية التمريض، جامعة الموصل

تاريخ القبول: 2008/03/04

تاريخ الاستلام: 2007/12/10

المخلص

تطُرقت هذه الدراسة إلى أساسيات البحث الممنوع مع مسألة من مسائل الأمثلية-مسألة البائع المتجول- وركزت على حل مسألة البائع المتجول باستخدام طريقة البحث الممنوع، للحصول على حل مثالي للمسألة وبفضاء بحث صغير وحسابات قليلة.

الكلمات المفتاحية: البحث الممنوع، مسألة البائع المتجول، مسألة البرمجة غير الخطية.

1. Introduction

Fred Glover proposed in 1986 a new approach, which he called tabu search. The basic principle of tabu search 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 (short-term memory), that record the recent history of the search[3,5]. Tabu search does not stop at the first local optimum when no improvement is possible the best solution in the neighborhood is always selected, even if it is worse than the current solution. This allows it to explore more solutions from the feasible region. To avoid cycling, a tabu list is used to store solutions that cannot be considered at next iteration. Typically, the tabu list cannot contain all of the previously visited solutions, because of the computational requirements.

Many researchers have attempted to describe or solve the problem with mathematical techniques, but present research solved with tabu search to find an optimal solution of small search space and computational requirements with the shortest distance.

The aim of this research is to present a combination of tabu search with traveling salesman problem to generate feasible solution.

In the figure (1), If a salesman starts at point A, and if the distance between every pair of points are known, what is the shortest route which visits all points and returns to point A ?

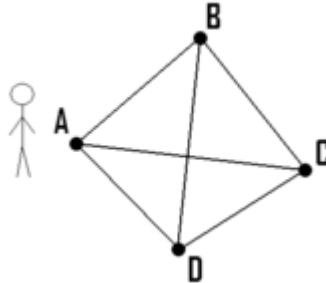


Figure (1)

1.1 Problem Definition

The symmetric traveling salesman problem is a problem in combinatorial optimization [9,12], and is equivalent to a class of mathematical problems called NP (nondeterministic polynomial) complete problems.

Traveling salesman problem (TSP) is the problem of a salesman who wants to find starting from his home town, a shortest trip through a given set of customer cities and to return to his home town. TSP can be represented by a complete weighted graph $G = (V, E)$ with V being vertices, representing the cities, and E is the set of edges fully connecting the vertices. Each assigned a value d_{ij} , which is the length of edge (i,j) is the distance between cities i and j . In the symmetric form of the problem $d(i,j) = d(j,i)$, where the pair (i,j) is the same as the pair (j,i) . In general, for C cities there exist $(C-1)!$ routes to consider [12].

1.2 Objective Function

We express the mathematical optimization problem as follows:

Minimize or Maximize $f(x)$

Subject to:

$$x \in X$$

Where $f(x)$ may be linear or nonlinear, and the set X summarizes constraints on the vector of decision variables x . [6].

The evaluation function of a symmetric traveling salesman problem is :

Given d_{ij} is the distance from city i to city j , the traveling salesman problem model is given as [10] :

$$(1) \text{ Min } z = \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_{ij}$$

Subject to:

$$(2) \sum_{i=1}^n x_{ij} = 1, \quad \text{for } j=1,2,\dots,n$$

$$(3) \sum_{j=1}^n x_{ij} = 1, \quad \text{for } i=1,2,\dots,n$$

$$(4) X_{ij} = \begin{cases} 1 & \text{if salesman travels from city } i \text{ to city } j \\ 0 & \text{otherwise} \end{cases}$$

$$(5) X_{ij} = 0, \quad \text{for } i=j$$

2. The Tabu Approach

The word tabu (or taboo) comes from Tongan, a language of Polynesia, where it was used by the aborigines of Tonga Island to indicate things that cannot be touched because they are sacred. The word now also means "a prohibition imposed by social custom a protective measure " or of something "banned as constituting a risk" [7].

The tabu search is a higher level method for solving optimization problems. The tabu search approach can be imposed on any procedure whose operation produce a sequence of moves that lead from one trial solution to another.

The tabu search algorithm starts at some initial solution and then move to a neighboring solution. A neighboring solution is generated by a set of admissible moves. At each iteration the method moves to the best solution in the neighborhood of the current solution [2].

Tabu search allows the search to explore solutions that do not decrease the objective function value only in those cases where these solutions are not forbidden. This is usually obtained by keeping track of the last solutions in terms of the action used to transform one solution to the

next. When an action is performed it is considered tabu for the next T iterations, where T is the tabu status length [4].

Tabu search is a procedure for solving combinatorial optimization problems to obtain optimal and sub optimal solutions. Tabu search starts from initial solution and attempts to determine a better solution in manner like Greedy method, the basic idea of the method is to explore the search space of all feasible solutions by a sequence of moves. A move is a change which transforms the current solution into one of its neighboring solutions. Associated with each move is a move value, which represents the change in the objective function value as a result of the move. At every iteration of the algorithm an admissible best move is applied to the current solution to obtain a new solution to be used in the next iteration. A move is applied even if it is a non-improving one that it does not lead to a solution better than the current solution. To escape from local optimal and to prevent cycling, a subset of moves is classified as tabu (or forbidden) for a number of iterations the classification depends on the history of the search.

Particularly as manifested in the recency or frequency that certain moves of solution components, called attributes have participated in generating past solutions. Might classify a move as tabu if the reverse move has been made recently. These restrictions are based on the maintenance of a short-term memory function (the tabu list), which determines how long a tabu restriction will be enforced, or which moves are admissible at each iteration.

The tabu tenure (the duration for which a move will be kept tabu) is an important feature of tabu search because it determines how restrictive the neighborhood search is. The tabu tenure t was selected depending on the size n of the problem by the algorithm $t = (n^{1/2}) [1]$.

The tabu restrictions are not inviolable under all circumstances and a tabu move can overridden under some conditions. A condition that allows such an override is called an aspiration criterion. An aspiration criterion might allow overriding a tabu move if the move leads to a solution, which is the best obtained so far [7,8].

2.1 Methodology of Tabu Search

Tabu search uses a local or neighborhood search procedure to iteratively move from a solution to another solution until some stopping criterion has been satisfied. To explore regions of the search space that would be left unexplored by the local search procedure and by doing this escape local optimality, tabu search modifies the neighborhood structure of each solution as the search progresses [11]. The most basic form of the tabu search algorithm consists of the following:

- 1- A methodology for generating an initial solution.
- 2- A mechanism for generating a neighboring solution of the current solution.
- 3- A function that measures each neighboring solution.
- 4- A tabu list in order to prevent cycling and leads the search to unexplored regions of the solution space.
- 5- An aspiration criterion (if a tabu move satisfies the aspiration criterion, it is considered admissible) [2].

A general framework of tabu search is given below [3,12]:

Begin

Step1 : Initialization:

- Input the number of cities n , and their distances that are between them.
- Input the stop criterion it; {stopping after a number of iterations}.
- Set tabu tenure t ; {is number of iteration for move will be kept tabu where $t = \sqrt{n}$ }.
- Initialize tabu list is empty; {tabu list = [position of city1 position of city2 position of city n], therefore tabu list becomes = [0 0 0 0.... 0] }.

Step2 :Use Greedy procedure to find best solution x^* with objective function z^* at

the same time is consider as current solution x as follows:

Let x , current solution.

Let $x^* = x$, the best solution.

Let $z^* = z(x)$, where $z^* = \{\text{summation of distance for } x \text{ solution}\}$.

Step3: Iteration:

While stop criterion is not satisfied Do

Begin

- Identify Neighborhood set by creating many solutions with used Move

operation. The move operation in the TSP is making swap between any two cities randomly of the current solution x :

each $x \in X$ has an associated Neighborhood $N(x) \subset X$.

Step3.1: Select best admissible move from $N(x)$ by choosing the solution has minimum $z(x)$ from this is transforms x into $x' \in N(X)$ and objective function value transforms to $z(x')$ and add its attributes to the tabu list.

Check x' is Tabu:

IF No Goto step 3.2

Else step 3.3

Step3.2: Perform exchange:

$x = x'$, $z(x) = z(x')$.

If $z(x) < z^*$ then

$z^* = z(x)$, $x^* = x$.

End if

Go to step 4

Step 3.3 Check x' is Aspiration

IF Yes Goto step 3.2

Else

Check Neighbor set.

Step 4 : Record tabu for the current move in tabu list, i.e. update the tabu list. by

change the contents of tabu list as following:

- If solution not tabu add solution to tabu list by change the position of city from zero to t For example if the problem consist of 5 cities and solution that resulted from (swap city 4 &city 3) is not tabu the tabu list must be [0 0 0 t 0].in the next iteration the tabu tenure decrees by one.

End while

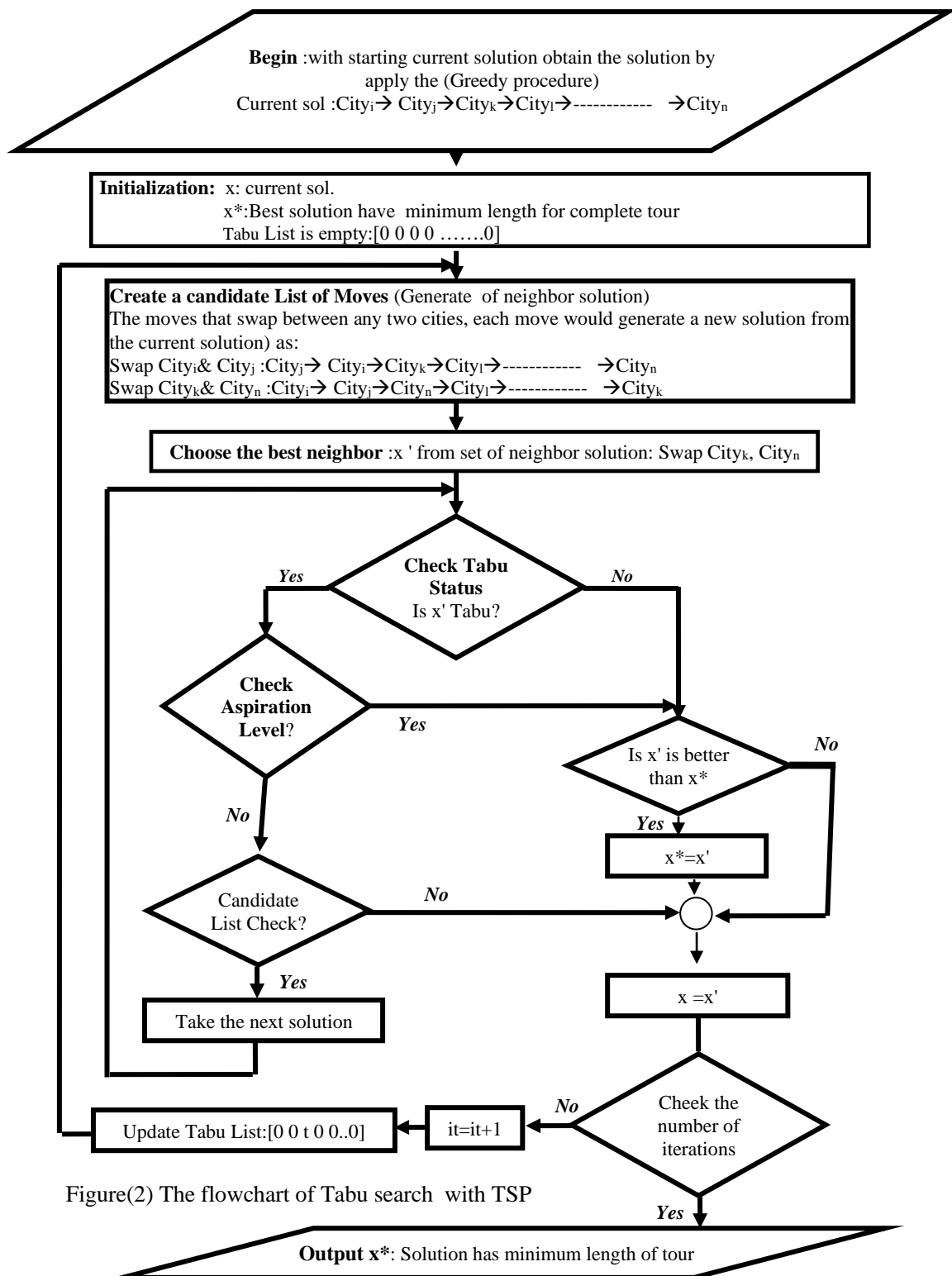
Step 5: Result: solution x^* has length of tour must be minimum of all determined solutions, with objective function value z^* .

End

3. Solving the Travel Salesman Problem with Tabu Search

Figure (2) illustrated the flowchart of the Tabu search algorithm with travel Salesman problem:

Tabu Search Method for Solving the Travel salesman Problem



Figure(2) The flowchart of Tabu search with TSP

3.1 Search Space (Configuration) of Problem

The Search space is the space of all possible solutions that can be considered (visited) during the search.[3] The search space size is $(C-1)!$. As represented in table (1).

	C1	C2	C3	Cn
C1	0	D C 1 C 2				
C2	D C 2 C 1	0				
:			0			
:				0		D C n-1 C n
Cn					D C n C n-1	0

Table (1)

3.2 Initial Solution (Configuration)

The initial configuration specifies where the search begins in the search space .In this research to generate the initial solution we used Greedy algorithm, below Greedy algorithm [7,9]:

- 1- Sort all edges.
- 2- Select the shortest edge and add it to our tour if it doesn't create a cycle with less than E edges, or increases the degree of any node to more than 2.
- 3- Do we have E edges in our tour ? If no, repeat step2.

3.3 Neighborhood Structure

The neighborhoods they define contain all the feasible route configurations that can be obtained from the current solution [3]. In neighborhood search, each solution $x \in X$ has an associated set of neighbors, $N(x) \subset X$, called the neighborhood of x , each solution $x' \in N(x)$ can be reached directly from x by an operation called Move. And x is said to move to x' when such an operation is performed.[6]

The neighborhood generally consists of solutions obtained by making small changes in the previous solution. In present research a Neighborhood configuration by making a single swap of current values of two randomly cities. There are many possible ways of selecting the best neighbor, including: first, choosing the first neighboring solution that improves objective function. Second, considering a subset of the neighbors and selecting the best of the set. Third, evaluating the whole set of neighboring solutions and selecting the best in terms of the objective functions [2]. The neighborhood is not a static set, but rather a set that can change according to the history of the search [7].

In our implementation in present research, second method is used for selecting the best neighbor.

3.4 Swapping

The swap move mechanism, which is used from this point onward, replaces a selected distance between two cities by another distance [7]. Or the swapping operation neighborhood considers each possible exchange of two (cities) in the problem.

3.5 Aspiration Criterion

In order to override the tabu list when there is a good tabu move, aspiration criterion is used, the tabu move is accepted if it produces better solution than the best obtained so far.[2,5]

3.6 Termination criterion

The most commonly used stopping criteria in tabu search are [3]:

- 1- Terminate the search after number of iterations.
 - 2- After some number of iterations without an improvement in the objective function value .
 - 3- When the objective reaches a pre-specified threshold value.
- In this research the first criteria is used for stopping criteria.

4. Experiments and Result

The experiments consist of running the tabu search methods with Travel salesman problem instance . In general decreasing the number of routes in search space leads to access the optimal solution. Matlab software was selected as the programming language.

4.1 Problem Instance

We will deal with the following constraints and definitions:

There are C1, C2, C3, C4, C5 the set of all cities .

The distance between each city to another in table (2).

The solution can be written as $2 \rightarrow 5 \rightarrow 1 \rightarrow 4 \rightarrow 3 \rightarrow 2$. An itinerary that begins and ends at the same city and visits each city once is called a tour.

	C1	C2	C3	C4	C5
C1	0	132	217	164	158
C2	132	0	290	201	79
C3	217	290	0	113	303
C4	164	201	113	0	196
C5	158	79	303	196	0

Table (2)

Suppose the initial solution to our problem is the shown in figure (3):

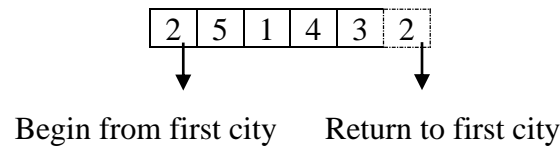
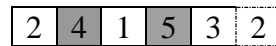


Figure (3)

The ordering in figure (3) specifies that city 2 is placed in the first position, followed by city 5, etc. Tabu search methods operate to identify the neighborhood that can be reached from any current solution. Identifying moves that lead from one solution to the next. In our problem, a swap exchanges the position of two cities as illustrated in figure (4):



Figure(4)swap of cities 5,4

Therefore, the complete neighborhood of a given current solution consists of 6 adjacent solutions that can be obtained by such swaps.

In this research the tabu tenure t was taken equal 3. The number of iterations tabu move is considered to remain tabu.

The core of the code programming designed in this research consists of many functions, and the program tabu makes use of other functions to perform the changes in the solution space. The input parameters of the program are the number of iterations, the matrix contains the number of the cities, distance between these cities, and the tabu tenure .

The final output of the program is an array of tabu list contains the last status of the problem and the best solution with the best cost as Table (3), and figure (5) provide a graphic illustration of the relationship between the iteration number with tour cost of our example .

Tabu Search Method for Solving the Travel salesman Problem

Below table (3): Shows 7 iterations of the basic tabu procedure that contain the current solution, best solution, tabu structure and candidate list of moves.

Iteration	The current sol.	Best solution	Best solution (Represent the path between cities)	Tabu structure (Tabu tener)	Candidate list of moves			
					Swap between 2 cities		Objec-tive functi-on	Tabu status & best sol.
0	804	804	2→5→1→4→3→2	[0 0 0 0 0]	5	1	889	*
					3	5	946	
					5	1	889	
1	889	804	2→1→5→4→3→2	[0 0 0 0 3]	1	5	804	Tabu
					1	5	804	
					3	4	907	
2	804	804	2→5→1→4→3→2	[0 0 0 0 2]	5	4	1098	
					4	1	928	Tabu
					1	5	889	
3	928	804	2→5→4→1→3→2	[0 0 0 3 1]	5	1	1085	
					5	4	1062	
					3	5	786	Tabu
4	786	786	2→3→4→1→5→2	[0 0 0 2 0]	4	1	946	
					4	1	946	
					3	4	768	
5	768	768	2→4→3→1→5→2	[0 0 0 1 0]	4	5	964	
					1	4	737	Tabu
					5	4	964	
6	737	737	2→1→3→4→5→2	[0 0 0 0 0]	1	3	946	
					3	4	791	*
					3	1	946	
7	791	737	2→1→4→3→5→2	[0 0 0 3 0]				

Table (3)

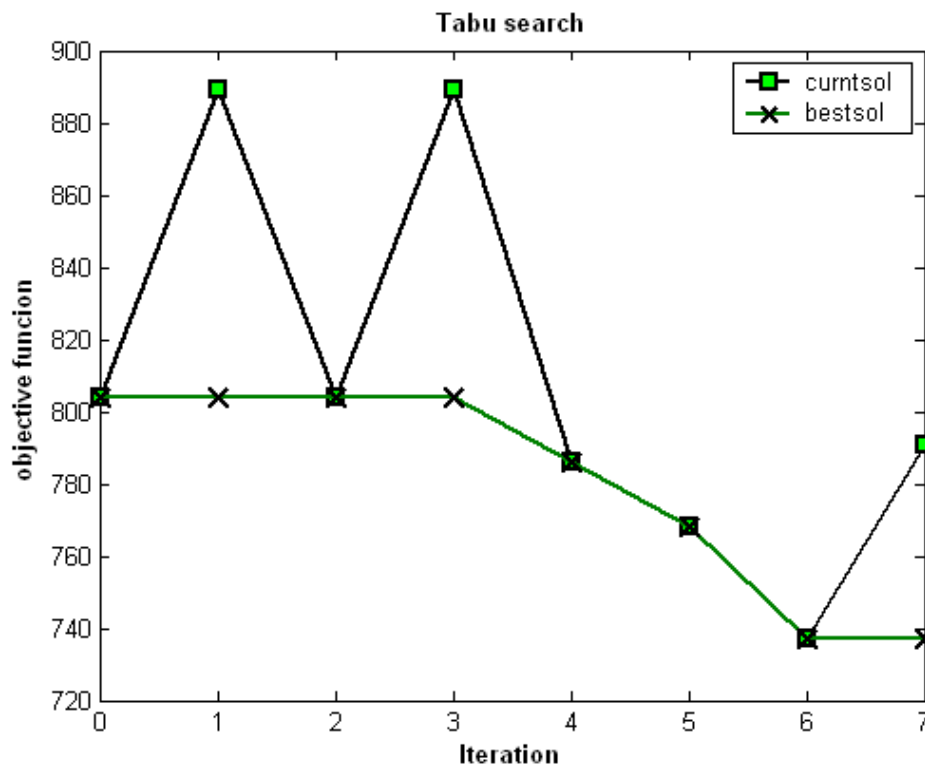


Figure (5)

4.2 Conclusion

Tabu search was used to solve the optimization problem. Tabu search is a powerful algorithmic approach that has been applied with great success to many difficult problems. The present research was studied traveling salesman problem because it is a touchstone for many general heuristics devised for combinatorial optimization like tabu search. The tabu search reached the optimal solution for the small size with 5 cities with 7 iterations. We noticed that the introduction of different neighborhood was necessary to reduce the computation time for tabu search, in which the approach swap between cities are efficient and create good solution, a neighborhood search algorithm searches among the neighbors of a candidate to find a better one. Tabu is useful to help the search move away from previously visited portions of the search space and perform more extensive exploration.

REFERENCES

- [1] Andersson, P., 2004, Using Tabu Search to Solve the 0-1 Knapsack Problem, Sweden, Institution for computer science, University of Linköping.
- [2] El-Amin, I., Duffuaa, S., 1999, Abbas, M. a Tabu Search Algorithm for Maintenance Scheduling of Generating Units, Electric Power System Research, 54 (2000) 91-99.
www.elsevier.com/locate/epsr.
- [3] Gendreau, M., 2002, An Introduction to Tabu Search , University of Montreal, Montreal, Canada.
- [4] Glover, F., 1986, Future Paths for Integer Programming and links to Artificial Intelligence, Comp. Oper. Res., Vol. 13, pp.533-549.
- [5] Glover, F., 1990, Tabu Search: A Tutorial Technical Report, Center for Applied Artificial Intelligence, University of Colorado, Boulder, Colorado.
- [6] Glover, F. and Laguna, M. 1993, Tabu Search, Colin R. Reeves (Ed.), 70—150, Blackwell Scientific publications, Oxford.
- [7] Glover, F. and Laguna, M. 1997, Tabu Search, Kluwer Academic Publishers, Norwell, MA.
- [8] Mahmood, A., Homeed, T.S.K., 2005, A Tabu Search Algorithm for Object Replication in Distributed Web Server Systems, Studies in Informatic and Control, Vol.14, No.2.
- [9] Nilsson, C. Heuristics for the Traveling Salesman Problem, University of Linköping.
- [10] Taha, H. A., 2003, Operations Research: An Introduction seventh Edition, Prentice Hall, Pearson Education International, Inc.
- [11] http://en.wikipedia.org/wiki/Tabu_search.
- [12] http://en.wikipedia.org/wiki/traveling_salesman_problem.
- [13] <http://unisci.com/stories/20021/0315023.htm>