American Journal of Operations Research, **2012, 2, 163-173**
http://dx.doi.org/10.4236/ajor.2012.22019 Published Online June 2012 (http://www.SciRP.org/journal/ajor)

Tabu Search Implementation on Traveling Salesman Problem and Its Variations: A Literature Survey

Sumanta Basu
Indian Institute of Management Calcutta, Kolkata, India
Email: sumanta@iimcal.ac.in

Received April 20, 2012; revised May 22, 2012; accepted May 30, 2012

ABSTRACT

The Traveling Salesman Problem (TSP) and its allied problems like Vehicle Routing Problem (VRP) are one of the most widely studied problems in combinatorial optimization. It has long been known to be NP-hard and hence research on developing algorithms for the TSP has focused on approximate methods in addition to exact methods. Tabu search is one of the most widely applied metaheuristic for solving the TSP. In this paper, we review the tabu search literature on the TSP and its variations, point out trends in it, and bring out some interesting research gaps in this literature.

Keywords: Tabu Search; Traveling Salesman Problem; Vehicle Routing Problem

1. Introduction

Many managerial problems, like vehicle routing problems, facility location problems, scheduling problems, network design problems, can either be modeled as combinatorial optimization problems, or solve combinatorial optimization problems as sub-problems. A very commonly researched combinatorial optimization problem in this and other contexts is the Traveling Salesman Problem (TSP). In a TSP (see e.g. [1]), we are given a weighted graph with a nodeset V and an arcset A. Cardinality of the nodeset V defines the problem size, *i.e.* $|V| = n$, and number of arcs is denoted by $|A|$. Cost of each arc connecting node i and node j is represented by $c_{ij} \in R_+; i, j \in V$. Given these input parameters, it is required to find a tour in the graph visiting each node exactly once such that the sum of the costs of the edges or arcs in the tour is the minimum possible. TSPs serve as a representation of many managerial problems, especially in logistics and distribution. Many more problems, though not obviously related to the TSP can be modeled as TSPs. A large number of other problems are not equivalent to solving TSPs, but solve TSPs as subproblems.

Apart from being a recurrent problem in managerial situations, the TSP is among the most widely studied problems in combinatorial optimization. It was one of the first problems whose decision version was shown to be NP-complete (see [2]), and has been a testbed for theoretical and computational studies ever since. Rich classes of benchmark problems exist for the TSP (see e.g., [3,4]), as do efficient implementations for solving reasonably large problems to optimality (see for example, NEOS: http://neos.mcs.anl.gov/neos/solvers/co:concorde/TSP.html).

The TSP is known to be NP-hard. This means that no known algorithm is guaranteed to solve all TSP instances to optimality within reasonable execution time. So in addition to exact solution approaches, a number of heuristics and metaheuristics have been developed to solve problems approximately. Heuristics and metaheuristics trade optimality of the solutions that they output with execution times. They are used to find "good" quality solutions within 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. Typical examples of metaheuristics include local search, tabu search, simulated annealing, and genetic algorithms. The literature shows that tabu search is one of the most widely used metaheuristic procedures to solve combinatorial optimization problems. It is an improvement heuristic based on local search. It starts with an initial solution to the problem, (a tour in case of the TSP), calls it a current solution, and searches for the best solution in a suitably defined neighborhood (a collection of tours that can be "easily" reached from the current solution) of the solution. It then designates the best solution in the neighborhood as the current solution and starts the search process again. Tabu search terminates when certain terminating conditions, either involving execution time or maximum iteration count conditions, or solution quality objectives, or both, have been met. In order to prevent tabu search from considering solutions that it has visited in recent iterations, tabu search maintains a list of neighbor ge-

neration moves that it considers forbidden, or tabu (hence the name, tabu search) and ignores solutions that can be reached using those tabu moves while searching the neighborhood of a solution. Once a move enters the list of tabu moves, it stays there for a pre-specified number of tabu search iterations (called the tabu tenure of the move). The list of tabu moves therefore changes continuously during the execution of the search, making tabu search an adaptive memory search algorithm. Several researchers have added features that enrich the basic tabu search algorithm described here, such as intermediate term memory structures, long term memory structures, and aspiration criteria, which have been widely applied to tabu search implementations for most problems in TSP or in VRP. Other features that have been proposed, but not commonly implemented for tabu search on TSPs are strategic oscillation, path relinking, candidate list strategies etc.

In this paper, we review the literature on application of tabu search to TSPs and problems very closely related to it, like vehicle routing problem and its variations. We reviewed 76 papers on the application of tabu search to these problems. The papers that we reviewed mostly appeared in print in the last twenty years. We classify the literature based on problem size (Section 2), generation of initial solutions (Section 3), selection of moves (Section 4), the choice of short, medium, and long term memory structures (Section 5 through Section 7), and aspiration criteria (Section 8). We summarize our findings in Section 9.

2. Problem Size

60 papers describe the problem context on TSPs and on VRPs where tabu search was implemented. **Table 1** provides a summary of the problem sizes considered by different authors in the literature. For TSP, problem size is defined as number of nodes to be visited and for VRP, problem size is considered as number of customers (or demand nodes) to be covered. It is interesting to note that even though metaheuristics are meant to handle large problems, nearly half of the papers deal only with problem sizes up to 100 nodes. There are only 11 papers which address problem sizes of more than 400 nodes. Only three papers (e.g. [5-7]) address problem sizes between 500 and 1035 nodes. With the advent of more powerful computers one would expect recent papers to deal with larger problems. However, as is evident from **Table 1**, this trend is not observed in published literature. For example, in the last three years, we have not encountered a single paper that implements tabu search on TSPs with more than 400 nodes.

3. Initial Solution Generation

Initial solution plays a vital role in finding out a good solution using local search based metaheuristic like tabu search. We have identified various methodologies used to generate initial solutions and have categorized papers in published literature using those methodologies in **Table 2**. To keep number of categories manageable, we have ignored minor customizations used to generate initial solutions in some papers while putting it into a category.

General Randomized Adaptive Search Process (GRASP) is a modification of a greedy tour construction heuristic. In a greedy heuristic, a tour is constructed by growing a path in the graph and joining the end points once the path includes all the nodes in the graph. The first point is randomly chosen, and the node closest to the endpoints of the path being formed is the next point to be added to the path. In a GRASP heuristic, the point that is added to the path being grown is not necessarily the closest one to the endpoints, but a random one chosen from a set of points that are close enough to the endpoints of the existing path.

Route First Cluster Second (RFCS) algorithm is used specifically in the context of VRP. Initially a giant tour is constructed by relaxing the vehicle capacity constraint as done in TSP. Then clusters are formed by breaking the giant tour into smaller tours which satisfy constraints specific to VRP, e.g. time window, capacity

Table 1. Problem sizes considered in published literature.

Problem Size	Years					
	Before 1996	1996-2000	2001-2005	2006-2010	After 2010	Total
100 or less	[8-10]	[11-13,20-22,26-31]	[7,14-16,23,24]	[17-19,25]	--	25
101 - 150	--	[32]	[33,34]	[35]	--	4
151 - 200	[36-38]	[39-41]	--	[17,42]	--	8
201 - 250	[43]	--	--	[44,45]	--	2
251 - 300	--	[46,47]	[34]	[48-50]	51	7
301 - 350	--	--	--	--	--	0
351 - 400	--	[52]	--	[53]	[54]	3
401 or more	[55,56]	[57-59]	[5-7,62]	[60,61]		11

Table 2. Methods to generate initial solutions.

Method	Years					
	Before 1996	1996-2000	2001-2005	2006-2010	After 2010	Total
GRASP	--	[30]	--	--	--	1
RFCS	--	[26]	--	[42]	[54]	3
CFRS	--	--	--	[42]	--	1
RandsIns	--	[12,28,40,57]	[34,62,62]	[19,25,60,61]	[51]	12
NN	--	[13,22,29,30,40,41]	[6,16,23,33,64,65]	[17,19,35,53]	--	15
Sweep	--	[58]	--	[35,44,49,53]	--	5
GENI	[36]	[46]	--	--	--	2
C&W	[37]	[11,59]	--	[17,49,50]	--	6

constraints etc.

Cluster First Route Second (CFRS) Algorithm is used in the context of VRP for initial allocation of customers to vehicles. In this algorithm, a set of nodes are identified to form a cluster by meeting VRP constraints. Then a feasible tour is constructed by visiting each of those nodes in a single cluster.

Randomized Insertion (RandIns) starts with a partial tour and inserts nodes randomly into tour without forming sub-tours in between. It stops when all nodes in the graph have been included in the tour.

Nearest Neighbor (NN) also starts with a partial tour. It then searches for a node not included in the partial tour and has the minimum distance from the last added node in the partial tour. It adds this node to the partial tour. If the graph describing the TSP is not complete, there is a possibility that after the completion of this procedure, some of the nodes remain unconnected to the tour. These nodes are then added to the tour using the RandIns procedure (by not forming sub-tours in between). In some papers, e.g. [6], authors implement minor variations of this method like double ended nearest neighbor etc.

Sweep heuristic is suited for TSPs defined on a plane. A reference point is chosen and an arbitrary axis is drawn through the reference point. Next the vertices are arranged in increasing order of angle between the line linking the vertex to the reference point and the axis defined. Ties are generally broken by choosing the vertex having the smallest distance to the reference point. Multiple reference points are used for multiple depots in the context of VRP.

Clarke and Wright (C & W) heuristic is widely used in the context of VRP (see [63]). It is based on the notion of savings. Initially routes are created to connect depot and customers. Those routes are then merged based on maximum savings possible in the parallel version of the algorithm. In the sequential version of the algorithm, the same route keeps expanding until no feasible route is possible.

We summarize our findings in usage of methods to generate initial solutions in **Table 2**. We can see that the two most popular methods are Randomized Insertion (RandsIn) and Nearest Neighbor (NN). Ease in implementation is a reason behind wide selection of those moves in published literature. Sweep and C & W heuristics are also used specifically in the context of VRP.

4. Moves

Tabu search being an improvement heuristic moves from one solution to the next in search of an optimal solution. The method of moving from one solution to another is described by a set of rules and is called a move. The set of all solutions that can be reached from a given solution using a pre-specified move is called the neighborhood of the solution. Out of the papers we reviewed, we found 98 instances of move description with multiple moves being discussed in some papers. One paper that emphasizes the influence of the choice of moves on the solution generated is by Osman ([37]), although it deals only with VRPs. The following types of moves have been used in the literature in the context of TSP and related problems. Again while putting the instances under a move category, we ignored the minor deviations from standard moves in some papers for problem specific implementation. The detailed paperwise summary is given in **Table 3**.

2-opt move involves removal of two non-adjacent edges from an existing tour and two new edges are inserted by connecting the head and tail nodes of the deleted edges to create a new tour without creating subtours. While performing this move in an asymmetric graph, an additional operation is required to reverse the intermediate arc chain and it increases the computational complexity. A minor variation of this move is used in the context of VRP which is denoted by 2-opt* move in **Table 3**.

r-opt move is a generalization of the 2-opt move where r > 2 edges are involved in deletion/addition operation. Computational complexity of an r-opt move is $O(n^r)$ in a symmetric graph where n is the problem size.

Table 3. Types of moves used in tabu search.

Method	Years					
	Before 1996	1996-2000	2001-2005	2006-2010	After 2010	Total
2-opt/2-opt*	[8,10,37,38]	[20,29,40,47,57]	[7,23,33,62]	[18,53,60,68]	[67]	19
r-opt (r > 2)	--	[29,52,57]	--	[25,61]	--	5
Or-opt	[10]	[20,57]	[7]	[68]	--	5
CrossEx	--	[12,57]	[65]	[18,53,68]	--	6
VertexIns	[9,36,37]	[11,28,39-41,58,59]	[5,6,16,23,24,34,50,64,62,65]	[17-19,35,45,49,53,60,61]	[54,69]	31
VertexEx	--	[28,32,39-41,59]	[6,7,16,23,24,62,65]	[17-19,35,45,49,53,50,61]	[54]	23
GENI	--	[58]	[6]	[44,48,68]	--	5
λ-InterCh	[37,55]	[22]	[7]	--	--	4

Hence mostly 3-opt or 4-opt moves are used in local search. It is practically infeasible to generate a neighborhood with r > 4 because it needs large computational time.

Or-opt move is a modification of the r-opt move. It was proposed by Or in 1976 (see [1] for details). It considers a small fraction of exchanges that would be considered by a regular r-opt move. In this move, due to computational convenience, only those exchanges are considered that would result in a string of up to three currently adjacent cities being inserted between two other cities.

Cross Exchange (CrossEx) is an advancement of 2-opt* move specifically used in VRP. Two edges are deleted from two different tours and the open nodes from each tour are connected to each other.

Vertex Insertion (VertexIns) consists of a move where a random vertex is chosen to remove from an existing tour and it is inserted between two other vertices to create a new neighboring tour.

Vertex Exchange (VertexEx) is a move where the positions of two vertices are interchanged to create a new tour from the existing one. In case of TSP, the vertices are exchanged in the same tour. In the context of VRP, vertices chosen can be from different tours as well.

Generalized Insertion (GENI) is a move introduced by Gendreau *et al.* ([66]). This move performs insertions of unrouted customers or removals of customers from their current routes and their reinsertions into different routes. In this move, one unrouted vertex is chosen, and is joined to two p-closest vertices (and not necessarily adjacent also) with p defined by a threshold distance. The nodes displaced from the route due to this operation are introduced at appropriate positions to complete the route.

λ-Interchange (λ-InterCh) is a move suggested by Osman ([37]), it works in the same principle as vertex exchange in a reduced neighborhood; only λ of the nearest nodes from a given node are considered for interchange.

Table 3 summarizes the choice of moves in published literature over time. It can easily be seen that the three frequently used moves are 2-opt, vertex insertion and vertex exchange. These moves are the easiest to implement among all the moves considered. They result in neighborhoods whose sizes are quadratic in the number of nodes in the TSP. Moves that result in larger neighborhoods often provide more improvement in each tabu search iteration, but searching them takes longer. For example, r-opt or GENI tends to produce a better solution in each iteration but in expense of higher computational time.

5. Choice of Short Term Memory Structure

Short term memory structures are used in tabu search to prevent the search from re-visiting solutions that it has visited in the immediate past. They are normally stored as a collection of forbidden moves in a list called the tabu list. Each move in the tabu list remains in the list for a pre-specified number of tabu search iterations. This number is called its tabu tenure. Tabu search implementations either keep the tabu tenure as a fixed number or one that changes deterministically with algorithm parameters (e.g., the number of tabu search iterations already executed) or problem parameters (e.g., problem size) or generate the tenure randomly within a pre-specified range. We refer to the first two kinds of tabu tenure as static tabu tenures, and the second one as random tabu tenure.

5.1. Static Tabu Tenure

47 of the instances in papers that we surveyed dealt with non-random tabu tenures. Of those, 26 dealt with tabu tenures that were fixed, and the other 21 instances dealt with tabu tenures that varied with the number of iterations already performed by tabu search, or with the instance size. While varying with number of iterations, search procedure actually sees the change in solution quality to determine the updated tabu tenure values. **Table 4** provides a list of the papers that dealt with fixed tabu tenures. **Table 5** presents a group of papers by the

Table 4. Papers with static tabu tenure with fixed tenure values.

Tenure Values	Years					
	Before 1996	1996-2000	2001-2005	2006-2010	After 2010	Total
0 - 4	[43]	--	--	--	--	1
5 - 9	[10,43]	[20,26,32]	[6,64,70]	--	--	8
10 - 14	[55]	[26,32]	[6,7,65]	--	--	6
15- 19	--	[26,32]	[23]	[45]	--	4
20 - 24	--	[26]	[62,65]	[42]	--	5
25 - 30	--	--	[65]	--	--	1
>30	--	--	[16]	--	--	1

Table 5. Papers with static tabu tenure with dependent tenure values.

Dependence Type	Years					
	Before 1996	1996-2000	2001-2005	2006-2010	After 2010	Total
Number of Iterations	--	[12,57]	--	[49,68]	[54]	5
Problem Size	[8,37]	[12,22,28,29,57]	[5,71,72]	--	--	8

parameter that they use to determine the tabu tenure.

From **Table 4** we see that the range of values for the tabu tenure lies mostly in between 5 to 25 across years. Further, nearly 60% of these papers use tabu tenures between 5 and 14. In papers like [26,43,70], the authors conducted experiments to see the effect of different tenure values on solution quality. In [26], the authors showed that static tenure outperforms all other kinds of tabu tenures in tabu search applied to TSPs. Of the different non-random tabu tenures used, a tabu tenure of 15 emerges as the best choice in [26]. In [70], the authors attempted to find out optimal tabu tenures for different sets of problems but were unable to generate any universal recommendation.

From **Table 5**, we observe that most of the papers that do not fix tabu tenure irrespective of the problem instance vary the tabu tenure based on the problem size. Only one of the papers that we surveyed ([37] for VRPs), describes an elaborate mechanism for determining the tabu tenure. In that paper describing tabu search implementation in VRP context by Osman, the tenure value depends on four problem parameters; a customer identification number, a vehicle identification number, a capacity ratio of the demand to the available vehicle capacities, and the type of moves considered. It was shown through computational experiments that results obtained by using this strategy gave a better solution than simulated annealing results, although the paper did not compare its strategy with other variants of tabu search.

There is also a more formalized approach called the functional approach (see e.g., [8,9]) to determine tabu tenures. In this approach the tabu tenure is determined using a function of problem specific parameters. The form of the function is pre-specified. The coefficients of the function are derived by regressing tenure value over other problem parameters for the best solutions found. In some papers like [5,35,44,58,71], the functional approach is followed to determine static tabu tenures. In [5,58], the form of the function is logarithmic, while in [35], both logarithmic and linear functions are used. The size of the TSP instance is taken as the only independent variable during regression in these papers. A detailed comparison of such static tabu tenures over different problem sizes appears in [29]. In this paper, tabu tenures ranging from n/32 to 3n/2 (where n is the size of the TSP) were tested for TSPs with sizes varying between 20 and 100 nodes. The paper concludes that the tabu tenure should be within n/8 and n/4 for 2-opt moves, and between n/16 and n/8 for 3-opt moves.

5.2. Random Tabu Tenure

The change of tabu tenure from deterministic to random was initiated in [74] for the quadratic assignment problem. Among the papers we surveyed, in 23 instances authors used random tabu tenures. In all of these papers, tabu tenures are picked randomly from a uniform distribution with fixed lower and upper bounds. In 10 papers, the limits of the distributions from which the tabu tenure is drawn varied with the problem size or number of iterations. **Table 6** summarizes the range of tabu tenures in papers where the support of the distribution from which tabu tenures did not depend on the problem being solved. It shows that the support of the distribution from which tabu tenure values were drawn was [5,10] in a majority of the papers. This support was first used in [36] and was motivated by a suggestion in [75].

Reference [38] influenced the trend of making tabu tenures depend on problem and search characteristics. In

Table 6. Papers with random tabu tenure with specified range.

Tenure Range	Years					
	Before 1996	1996-2000	2001-2005	2006-2010	After 2010	Total
5 - 10	[36]	[27,40,52,59]	[24,33]	[17]	--	8
10 - 20	--	[40]	[65]	[53]	--	3
20 - 30	--	[21]	[21]	--	--	2

this paper, the authors modified the tabu tenure based on solution values in previous iterations. After this, 10 papers have appeared in the literature in which the support of the distribution from which the tabu tenure is chosen is based on the problem being solved. All of them (except [52] and [53]), considered instances with between 50 and 150 nodes. Reference [53] considered instances with 400 nodes.

Most of the work on the dependence of tabu tenure on instance size is due to Cordeau, Gendreau and their coauthors. They developed a functional relation for the ranges of distribution based on the instance size. There are however two distinct trends in the modeling of the dependence. In [11,13,47], the support of the distribution from which tabu tenures are chosen were directly proportional to the problem size, while in [34,58], the dependence is logarithmic. There is no empirical evidence to show that one of these forms is better than the other.

6. Choice of Intermediate Term Memory Structure

Intermediate term memory structures are used in tabu search to intensify the search by restricting it to promising regions of the solution space. We found 24 instances in the papers reviewed implemented intermediate term memory structures. We classify the strategies used into the following broad four categories:

Strategy IT1: Edges that occur frequently in low cost tours are forced into candidate tours for next few iterations.

Strategy IT2: The search is re-started with a tour that was one of the lower cost tours found in previous iterations.

Strategy IT3: Higher probabilities are assigned to include the edges common to previously encountered low cost tours in a probabilistic tabu search.

Strategy IT4: Changing tabu tenure (in comparison to existing tenure value) whenever a local optimum is reached.

As mentioned in the previous sections, we ignored minor deviations from the categories identified for grouping convenience. **Table 8** presents information about the use of the four strategies. We see that the first two strategies have been used in two-third of the papers cited. Strategy IT1 has a wide acceptability because it restricts the solution space. In Strategy IT2, the search is re-initiated from a promising region without any particular restriction in the search process. The scarcity of papers involving Strategy IT3 is expected, because it works for probabilistic tabu search, which itself is not frequently used. In some papers (see e.g., [24,38,57]), more than one of these strategies are used together.

7. Choice of Long Term Memory Structure

Tabu search uses long term memory structures to diversify the search to new regions in the solution space. We found 34 instances in papers that used long term memory structures to achieve diversification. We list the broad strategies used below.

Strategy LT1 is a frequency based diversification scheme implemented by adding a penalty value to the cost of each edge. The penalty is proportional to the number of times the edge appeared in previously visited tours. The objective here is to create a disincentive for including edges that were often encountered previously.

Strategy LT2 is a modification of strategy LT1, in which the penalty value also includes terms that are not dependent on frequency measures.

Strategy LT3 attains diversification by changing the way in which tours are evaluated in order to move the search to new parts of the search space. It may also involve changing the move being used in the search.

Strategy LT4 creates a diversification mechanism by changing the stopping criterion to allow more non-improving moves.

Strategy LT5 is a diversification scheme which restarts tabu search iteration from different initial solutions. It helps to change the initial search space and creates a chance to reach to a different solution region.

Strategy LT6 is used in some papers for diversification if no improvement is seen in the best tour cost for certain number of iterations, diversification is attained by adding a parameter called influence measure to the tour costs for neighboring solutions. The influence measure measures the degree of similarity between two consecutive solutions.

This list of diversification strategies is not exhaustive and we clubbed minor variations of some strategies into broad categories defined above.

Table 9 presents the usage of long term memory structures in the tabu search literature on TSP and VRP. It is

Table 7. Papers with random tabu tenure with dependent range.

Range Dependency	Years					
	Before 1996	1996-2000	2001-2005	2006-2010	After 2010	Total
Number of Iterations	[38]	[11]	[24]	--	--	3
Problem Size	--	[11,13,31,39,47,58]	[34]	--	--	7

Table 8. Papers with intermediate term memory function.

Strategy	Years					
	Before 1996	1996-2000	2001-2005	2006-2010	After 2010	Total
IT1	[38]	[12,20,22,40]	[62,76]	[53,60]	--	9
IT2	[36]	[52,57]	[24]	[19,35,73]	--	7
IT3	[56]	[57]	[24]	--	--	3
IT4	[9,38]	[31]	--	[68]	[54]	5

Table 9. Papers with long term memory function.

Strategy	Years					
	Before 1996	1996-2000	2001-2005	2006-2010	After 2010	Total
LT1	[8]	[12,13,22,27,39,57]	[6,24,65,76]	[45,48,53]	[51,54]	16
LT2	[36]	[58]	[5,34]	[25,35,61]	--	7
LT3	--	[20]	[24,33]	[49]	--	4
LT4	--	[31,52]	--	[19]	--	3
LT5	[56]	--	[24]	--	--	2
LT6	--	--	[62]	[60]	--	2

clear from the table that Strategy LT1 is the most commonly used strategy for using long term memory structures. Reference [36] used Strategy LT2 for the first time in their tabu search implementation. In their implementation, the penalty value was dependent on a factor equal to absolute difference between two successive values of objective function, the square root of the neighborhood size for a particular move, and a scaling factor to control the intensity of diversification to be achieved. The concept of a scaling factor was also used in [5,34, 35,58], among others. Reference [24] used Strategy LT3 in their tabu search implementation. They modified the cost of tours to facilitate the inclusion of non-frequent moves. In the tabu search implementation in [33], edges do not enter the tabu list if the cost of the tour is within 10% of the cheapest tour up to that iteration. In reference [20], Or-opt moves were used instead of 2-opt move when it could not improve the best tour for a specific number of iterations. References [60,62] used influence measures to diversify their tabu search. By inserting this expression in the objective function, they restricted moves to visit similar kinds of tours.

8. Aspiration Criteria

Aspiration criteria are a set of conditions which if satisfied, permits tabu search to make use of tabu moves to reach neighboring solutions. 40 of the papers that we reviewed used aspiration criteria. Although the design of aspiration criteria can be finely tuned, all the papers except [58] chose to overrule the tabu status of moves if they allowed the search to find a tour that was better than the best found until that iteration. [58] developed attribute specific adaptive aspiration level functions instead of taking solution values.

9. Conclusion and Future Directions

In this literature, we examine various aspects of tabu search implementation in the context of TSP and allied problems. We reviewed this literature from implementation as well as from methodological point of view. While seeing the implementation aspect, we find from **Table 1** that most of the papers do not address a practically sized large problem size. More than 40% of the papers chose a problem size of 100 nodes or less. Also most authors consider symmetric TSPs (STSP) described on complete graphs for tabu search implementation. If we look into ATSP, success of tabu search is rather limited if compared with the success in the context of STSP. As mentioned in [3], one reason for this could be the absence of particular instance type in ATSP which

enables specific heuristics to reduce computational time by exploiting problem characteristics. Contrary to STSP, where two dimensional geometric instances dominate, there is no single instance type which dominates in the field of ATSP. A particular ATSP instance can be represented by a list of n(n – 1) inter-city distances, where n is the problem size. This creates a large space requirement and makes it difficult to handle large problem instances. An attempt to address ATSP instances is more practically motivated because of two reasons; most practical networks are asymmetric, and the STSP is a special case of the ATSP. If we take any distribution and logistics problem to see the nature of underlying graph, it is likely to be asymmetric and typically problem size is large. We believe these two important aspects of problem context, *i.e.* size and asymmetry, are missing in existing literature.

While generating initial solutions for tabu search, authors do not use very sophisticated methods for tabu search implementations (**Table 2**). It is not clear whether they do this because the neighborhoods of good initial solutions do not provide other good solutions, or whether tabu search is found to be powerful enough to generate good quality solutions regardless of the initial solution. Computational effort will surely be less to generate initial solutions using Random Insertion or Nearest Neighborhood method.

To generate neighboring solutions from an initial solution, authors prefer moves that are simple to implement and which give rise to small neighborhoods. In the last few years, they have primarily considered vertex insertion, vertex exchange,and 2-opt moves. It seems that they find running more tabu search iterations with less improvement per iteration to be a better option than running few tabu search iterations each of which could provide potentially much larger improvement. Reduction on computational time becomes specifically important for increasing problem size (with asymmetry).

The intensification and diversification within the search process are achieved by three different memory structures: short term, intermediate term and long term. Most authors prefer to use short term memory structures in their tabu search implementations. The issue of deciding tabu tenures has not received adequate attention in the literature. In aggregate more authors have preferred fixed tabu tenures over random tabu tenures. This preference seems to have increased in recent years. The problem comes while implementing tabu search on large problem sizes because commonly used fixed tenure values tend to fail since the neighborhood size is large. Among authors who have chosen to use fixed tabu tenures, more authors are beginning to tailor the tabu tenures. First, papers have looked at only linear and logarithmic dependences of the tabu tenure on problem size. We did not find any justification for restricting themselves to only these functional forms. It would also be interesting to experiment with other functional forms of dependence of the tabu tenure on problem size. Second, papers that have modified tabu tenures based on the problems being solved look at the size of the problem as the only problem specific criterion. It would be interesting to see if the length of the tabu tenure could be made dependent on other problem characteristics to yield better results.

Intermediate term memory structure is used mostly for intensification to focus the search process into a specific region. Strategy IT1 has its advantage of reduction of neighborhood size and hence reduction in computational time. In strategy IT4, typically the tabu tenure is reduced to intensify the search process to a specific region. In case of functional tenure values, the process of changing the tenure values is not explained very clearly in the literature. Changes in the functional form is also necessary. Detailed study of using a particular type of intermediate term memory based on the neighborhood structure is required to draw some meaningful conclusion over the usefulness of this memory structure. Some composite strategy for intensification can also be experimented.

Long term memory structure leads to the diversification of the search process. From the published literature, strategy LT1 and LT2 are the two strategies which are used over the years to implement this structure in tabu search. Following in similar line as suggested in intermediate term memory structure, composite strategies can be developed for diversification as well.

None of the papers that we reviewed used all of the features of tabu search in their tabu search implementations. Although in recent years, more number of published papers address tabu search implementation with more than two or three features. Components like strategic oscillation were ignored in past due to its difficulty in implementation. A shift in this area is also observed as more infrequent features of tabu search are included in recent papers.

10. Acknowledgements

I thank Prof. Diptesh Ghosh and Dr. G. S. Ravindra for their valuable inputs and suggestions in various phases of this work. I am also thankful to the anonymous referees for their helpful suggestions and comments to improve the quality and clarity of the paper.

REFERENCES

[1] E. Lawler, J. lenstra, K. Rinnooy and D. Shmoys, "The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization," Wiley-Interscience Publication,

1985.

[2] R. Karp, "Reducibility among Combinatorial Problems," Plenum Press, New York, 1972, pp. 85-103.

[3] D. Johnson, G. Gutin, L. McGeoch, A. Yeo, W. Zhang and A. Zverovich, "The Traveling Salesman Problem and Its Variations," Kluwer Academic Publishers, Boston, 2002.

[4] G. Reinelt, "TSPLIB—A Traveling Salesman Problem Library," *INFORMS Journal of Computing*, Vol. 3, 1991, pp. 376-384. doi:10.1287/ijoc.3.4.376

[5] J. Cordeau, G. Laporte and A. Mercier, "A Unified Tabu Search Heuristic for Vehicle Routing Problems with Time Windows," *The Journal of the Operational Research Society*, Vol. 52, No. 8, 2001, pp. 928-936. doi:10.1057/palgrave.jors.2601163

[6] A. Bouthillier and T. Crainic, "A Cooperative Parallel Meta-Heuristic for the Vehicle Routing Problem with Time Windows," *Computers & Operations Research*, Vol. 32, No. 7, 2005, pp. 1685-1708. doi:10.1016/j.cor.2003.11.023

[7] J. Homberger and H. Gehring, "A Two-Phase Hybrid Metaheuristic for the Vehicle Routing Problem with Time Windows," *European Journal of Operational Research*, Vol. 162, No. 1, 2005, pp. 220-238. doi:10.1016/j.ejor.2004.01.027

[8] M. Malek, M. Guruswamy, M. Pandya and H. Owens, "Serial and Parallel Simulated Annealing and Tabu Search Algorithms for the Traveling Salesman Problem," *Annals of Operations Research*, Vol. 21, No. 1, 1989, pp. 59-84. doi:10.1007/BF02022093

[9] F. Semet and E. Taillard, "Solving Real-Life Vehicle Routing Problems Efficiently Using Taboo Search," *Annals of Operations Research*, Vol. 41, 1993, pp. 469-488. doi:10.1007/BF02023006

[10] B. Garcia, J. Potvin and J. Rousseau, "A Parallel Implementation of the Tabu Search Heuristic for Vehicle Routing Problems with Time Window Constraints," *Computers & Operations Research*, Vol. 21, No. 9, 1994, pp. 1025-1033. doi:10.1016/0305-0548(94)90073-6

[11] M. Gendreau, G. Laporte and R. Séguin, "A Tabu Search Heuristic for the Vehicle Routing Problem with Stochastic Demands and Customers," *Operations Research*, Vol. 44, No. 3, 1996, pp. 469-477. doi:10.1287/opre.44.3.469

[12] P. Badeau, M. Gendreau, F. Guertin, J.-Y. Potvin and E. Taillard, "A Parallel Tabu Search Heuristic for the Vehicle Routing Problem with Time Windows," *Transportation Research*, Vol. 5, 1997, pp. 109-122. doi:10.1016/S0968-090X(97)00005-3

[13] J. Brandão and A. Mercer, "A Tabu Search Algorithm for the Multi-Trip Vehicle Routing and Scheduling Problem," *European Journal of Operational Research*, Vol. 100, No. 1, 1997, pp. 180-191. doi:10.1016/S0377-2217(97)00010-6

[14] O. Bräysy and M. Gendreau, "Tabu Search Heuristics for the Vehicle Routing Problem with Time Windows," Technical Report, SINTEF Applied Mathematics, Department of Optimisation, Oslo, 2001.

[15] P. Caricato, G. Ghiani, A. Grieco and E. Guerriero, "Parallel Tabu Search for a Pickup and Delivery Problem under Track Contention," *Parallel Computing*, Vol. 29, No. 5, 2003, pp. 631-639. doi:10.1016/S0167-8191(03)00046-2

[16] H. Lau, M. Sim and K. Teo, "Vehicle Routing Problem with Time Windows and a Limited Number of Vehicles," *European Journal of Operational Research*, Vol. 148, No. 3, 2003, pp. 559-569. doi:10.1016/S0377-2217(02)00363-6

[17] C. Lin and R. Kwok, "Multi-Objective Metaheuristics for a Location-Routing Problem with Multiple Use of Vehicles on Real Data and Simulated Data," *European Journal of Operational Research*, Vol. 175, No. 3, 2006, pp. 1833-1849.

[18] N. Bianchessi and G. Righini, "Heuristic Algorithms for the Vehicle Routing Problem With Simultaneous Pick-Up and Delivery," *Computers & Operations Research*, Vol. 34, No. 2, 2007, pp. 578-594. doi:10.1016/j.cor.2005.03.014

[19] J. Brandão, "A Deterministic Tabu Search Algorithm For the Fleet Size and Mix Vehicle Routing Problem," *European Journal of Operational Research*, Vol. 195, 2009, pp. 716-728. doi:10.1016/j.ejor.2007.05.059

[20] J. Potvin, T. Kervahut, B. Garcia and J. Rousseau, "The Vehicle Routing Problem with Time Windows—Part I: Tabu Search," *INFORMS Journal of Computing*, Vol. 8, 1996, pp. 158-164. doi:10.1287/ijoc.8.2.158

[21] Y. Sharaiha, M. Gendreau, G. Laporte and I. Osman, "A Tabu Search Algorithm for the Capacitated Shortest Spanning Tree Problem," *Networks*, Vol. 29, 1997, pp. 209-223. doi:10.1002/(SICI)1097-0037(199705)29:3<161::AID-NET4>3.0.CO;2-F

[22] W. Chiang and R. Russell, "A Reactive Tabu Search Metaheuristic for the Vehicle Routing Problem with Time Windows," *INFORMS Journal of Computing*, Vol. 9, 1997, pp. 417-430. doi:10.1287/ijoc.9.4.417

[23] S. Ho and D. Haugland, "A Tabu Search Heuristic for the Vehicle Routing Problem with Time Windows and Split Deliveries," *Computers & Operations Research*, Vol. 31, No. 12, 2004, pp. 1947-1964. doi:10.1016/S0305-0548(03)00155-2

[24] H. Tang and E. Hooks, "A TABU Search Heuristic for the Team Orienteering Problem," *Computers & Operations Research*, Vol. 32, No. 6, 2005, pp. 1379-1407. doi:10.1016/j.cor.2003.11.008

[25] M. Bolduc, G. Laporte, J. Renaud and F. Boctor, "A Tabu Search Heuristic for the Split Delivery Vehicle Routing Problem with Production and Demand Calendars," *European Journal of Operational Research*, Vol. 202, 2010, pp. 122-130. doi:10.1016/j.ejor.2009.05.008

[26] A. Amberg, W. Domschke and S. Vob, "Multiple Center Capacitated Arc Routing Problems: A Tabu Search Algorithm Using Capacitated Trees," *European Journal of Operational Research*, Vol. 124, No. 2, 2000, pp. 360-376.

[27] A. Hertz, G. Laporte and M. Mittaz, "A Tabu Search Heuristic for the Capacitated Arc Routing Problem," *Operations Research*, Vol. 48, No. 1, 2000, pp. 129-135.

doi:10.1287/opre.48.1.129.12455

[28] W. Nanry and J. Barnes, "Solving the Pickup and Delivery Problem with Time Windows Using Reactive Tabu Search," *Transportation Research Part B*: *Methodological*, Vol. 34, No. 2, 2000, pp. 107-121. doi:10.1016/S0191-2615(99)00016-8

[29] S. Tsubakitani and J. Evans, "Optimizing Tabu List Size for the Traveling Salesman Problem," *Computers & Operations Research*, Vol. 25, No. 2, 1998, pp. 91-97. doi:10.1016/S0305-0548(97)00030-0

[30] R. Daniels, J. Rummel and R. Schantz, "A Model for Warehouse Order Picking," *European Journal of Operational Research*, Vol. 105, No. 1, 1998, pp. 1-17. doi:10.1016/S0377-2217(97)00043-X

[31] P. Franca, N. Sosa and V. Pureza, "An Adaptive Tabu Search Algorithm for the Capacitated Clustering Problem," *International Transactions in Operations Research*, Vol. 6, 1999, pp. 665-678.

[32] P. Augerat, J. M. Belenguer, E. Benavent, A. Corberán and D. Naddef, "Separating Capacity Constraints in the CVRP Using Tabu Search," *European Journal of Operational Research*, Vol. 106, No. 2-3, 1998, pp. 546-557. doi:10.1016/S0377-2217(97)00290-7

[33] I. Chao, "A Tabu Search Method for the Truck and Trailer Routing Problem," *Computers & Operations Research*, Vol. 29, No. 1, 2002, pp. 33-51. doi:10.1016/S0305-0548(00)00056-3

[34] J. Cordeau and G. Laporte, "A Tabu Search Heuristic for the Static Multi-Vehicle Dial-a-Ride Problem," *Transportation Research Part B*: *Methodological*, Vol. 37, No. 6, 2003, pp. 579-594. doi:10.1016/S0191-2615(02)00045-0

[35] S. Scheuerer, "A Tabu Search Heuristic for the Truck and Trailer Routing Problem," *Computers & Operations Research*, Vol. 33, No. 4, 2006, pp. 894-909. doi:10.1016/j.cor.2004.08.002

[36] M. Gendreau, A. Hertz and G. Laporte, "A Tabu Search Heuristic for the Vehicle Routing Problem," *Management Science*, Vol. 40, No. 10, 1994, pp. 1276-1290. doi:10.1287/mnsc.40.10.1276

[37] I. Osman, "Metastrategy Simulated Annealing and Tabu Search Algorithms for the Vehicle Routing Problem," *Annals of Operations Research*, Vol. 41, 1993, pp. 421-451. doi:10.1007/BF02023004

[38] Y. Rochat and F. Semet, "A Tabu Search Approach for Delivering Pet Food and Flour in Switzerland," *The Journal of the Operational Research Society*, Vol. 45, No. 11, 1994, pp. 1233-1246.

[39] J. Brandão and A. Mercer, "The Multi-Trip Vehicle Routing Problem," *The Journal of the Operational Research Society*, Vol. 49, No. 8, 1998, pp. 799-805.

[40] G. Barbarosoglu and D. Ozgur, "A Tabu Search Algorithm for the Vehicle Routing Problem," *Computers and Operations Research*, Vol. 26, No. 3, 1999, pp. 255-270. doi:10.1016/S0305-0548(98)00047-1

[41] D. Tuzun and L. Burke, "A Two-Phase Tabu Search Approach to the Location Routing Problem," *European Journal of Operational Research*, Vol. 116, 1999, pp. 87-99. doi:10.1016/S0377-2217(98)00107-6

[42] D. Ahr and G. Reinelt, "A Tabu Search Algorithm for the Min-Max k-Chinese Postman Problem," *Computers & Operations Research*, Vol. 33, No. 12, 2006, pp. 3403-3422. doi:10.1016/j.cor.2005.02.011

[43] T. G. Crainic, M. Gendreau, P. Soriano and M. Toulouse, "A Tabu Search Procedure for Multicommodity Location/Allocation with Balancing Requirements," *Annals of Operations Research*, Vol. 41, 1993, pp. 359-383. doi:10.1007/BF02023001

[44] B. Crevier, J. Cordeau and G. Laporte, "The Multi-Depot Vehicle Routing Problem with Inter-Depot Routes," *European Journal of Operational Research*, Vol. 176, No. 2, 2007, pp. 756-773. doi:10.1016/j.ejor.2005.08.015

[45] E. Zachariadis, C. Tarantilis and C. Kiranoudis, "A Guided Tabu Search for the Vehicle Routing Problem with Two-Dimensional Loading Constraints," *European Journal of Operational Research*, Vol. 195, 2009, pp. 729-743. doi:10.1016/j.ejor.2007.05.058

[46] M. Gendreau, G. Laporte and F. Semet, "A Tabu Search Heuristic for the Undirected Selective Travelling Salesman Problem," *European Journal of Operational Research*, Vol. 106, 1998, pp. 539-545. doi:10.1016/S0377-2217(97)00289-0

[47] M. Gendreau, G. Laporte and D. Vigo, "Heuristics for the Traveling Salesman Problem with Pickup and Delivery," *Computers & Operations Research*, Vol. 26, No. 7, 1999, pp. 699-714. doi:10.1016/S0305-0548(98)00085-9

[48] M. Gendreau, M. Iori, G. Laporte and S. Martello, "A Tabu Search Heuristic for the Vehicle Routing Problem with Two-Dimensional Loading Constraints," *Networks*, Vol. 51, No. 1, 2008, pp. 4-18. doi:10.1002/net.20192

[49] N. A. Wassan, A. Wassan and G. Nagy, "A Reactive Tabu Search Algorithm for the Vehicle Routing Problem with Simultaneous Pickups and Deliveries," *Journal of Combinatorial Optimization*, Vol. 15, 2008, pp. 368-386. doi:10.1007/s10878-007-9090-4

[50] H. Tamashiro, M. Nakamura, T. Okazaki and D. Kang, "A Tabu Search Approach Combined with an Extended Saving Method for Multi-Depot Vehicle Routing Problems with Time Windows," *Biomedical Soft Computing and Human Sciences*, Vol. 15, No. 1, 2010, pp. 31-39.

[51] J. Cordeau and M. Maischberger, "A Parallel Iterated Tabu Search Heuristic for Vehicle Routing Problems," *Computers & Operations Research*, Vol. 39, 2012, pp. 2033-2050. doi:10.1016/j.cor.2011.09.021

[52] J. Renaud, G. Laporte and F. Boctor, "A Tabu Search Heuristic for the Multi-Depot Vehicle Routing Problem," *Computers & Operations Research*, Vol. 23, 1996, pp. 229-235. doi:10.1016/0305-0548(95)O0026-P

[53] F. Montane and R. Galvao, "A Tabu Search Algorithm for the Vehicle Routing Problem with Simultaneous Pick-Up and Delivery Service," *Computers & Operations Research*, Vol. 33, No. 3, 2006, pp. 595-619. doi:10.1016/j.cor.2004.07.009

[54] J. Brandão, "A Tabu Search Algorithm for the Heterogeneous Fixed Fleet Vehicle Routing Problem," *Computers & Operations Research*, Vol. 38, 2011, pp. 140-151.

doi:10.1016/j.cor.2010.04.008

[55] S. Thangiah, I. Osman and T. Sun, "Hybrid Genetic Algorithm, Simulated Annealing and Tabu Search Methods for Vehicle Routing Problem with Time Windows," Technical Report, Computer Science Department, Slippery Rock University, 1994.

[56] Y. Rochat and E. Taillard, "Probabilistic Diversification and Intensification in Local Search for Vehicle Routing," *Journal of Heuristics*, Vol. 1, 1995, pp. 147-167. doi:10.1007/BF02430370

[57] V. E. Taillard, P. Badeau, M. Gendreau, F. Guertin, and J.-Y. Potvin, "A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows," *Transportation Science*, Vol. 31, No. 2, 1997, pp. 170-186. doi:10.1287/trsc.31.2.170

[58] J. Cordeau, M. Gendreau and G. Laporte, "A Tabu Search Heuristic for Periodic and Multi-Depot Vehicle Routing Problems," *Networks*, Vol. 30, No. 2, 1998, pp. 105-119. doi:10.1002/(SICI)1097-0037(199709)30:2<105::AID-NET5>3.0.CO;2-G

[59] P. Toth and D. Vigo, "The Granular Tabu Search and Its Application to the VRP," Technical Report, University of Bologna, 1998.

[60] E. Zachariadis, C. Tarantilis and C. Kiranoudis, "A Guided Tabu Search for the Vehicle Routing Problem with Two-Dimensional Loading Constraints," *European Journal of Operational Research*, Vol. 195, 2009, pp. 729-743. doi:10.1016/j.ejor.2007.05.058

[61] J. Côté and J.-Y. Potvin, "A Tabu Search Heuristic for the Vehicle Routing Problem with Private Fleet and Common Carrier," *European Journal of Operational Research*, Vol. 198, 2009, pp. 464-469. doi:10.1016/j.ejor.2008.09.009

[62] C. Tarantilis, "Solving The Vehicle Routing Problem with Adaptive Memory Programming Methodology," *Computers & Operations Research*, Vol. 32, No. 9, 2005, pp. 2309-2327. doi:10.1016/j.cor.2004.03.005

[63] G. Clarke and J. Wright, "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points," *Operations Research*, Vol. 12, 1964, pp. 568-581. doi:10.1287/opre.12.4.568

[64] J. Shen, F. Xu and P. Zheng, "A Tabu Search Algorithm for the Routing and Capacity Assignment Problem in Computer Networks," *Computers & Operations Research*, Vol. 32, No. 11, 2005, pp. 2785-2800. doi:10.1016/j.cor.2004.04.004

[65] A. Breedam, "Comparing Descent Heuristics and Metaheuristics for the Vehicle Routing Problem," *Computers & Operations Research*, Vol. 24, No. 4, 2001, pp. 289-315. doi:10.1016/S0305-0548(99)00101-X

[66] M. Gendreau, A. Hertz and G. Laporte, "New Insertion and Post Optimization Procedures for the Traveling Salesman Problem," *Operations Research*, Vol. 40, No. 6, 1992, pp. 1086-1094. doi:10.1287/opre.40.6.1086

[67] X. Fan, N. Li, B. Zhang and Z. Liu, "Research on Vehicle Routing Problem with Soft Time Windows Based on Tabu Search Algorithm," *IEEE* 18*th International Conference*, 3-5 September 2011.

[68] D. Paraskevopoulos, P. Repoussis, C. Tarantilis, G. Ioannou and G. Prastacos, "A Reactive Variable Neighborhood Tabu Search for the Heterogeneous Fleet Vehicle Routing Problem with Time Windows," *Journal of Heuristics*, Vol. 14, 2008, pp. 425-455. doi:10.1007/s10732-007-9045-z

[69] L. Moccia, J. Cordeau and G. Laporte, "An Incremental Tabu Search Heuristic for the Generalized Vehicle Routing Problem with Time Windows," *Journal of the Operational Research Society*, Vol. 63, 2012, pp. 232-244. doi:10.1057/jors.2011.25

[70] T. Wu, C. Low and J. Bai, "Heuristic Solutions to Multi-Depot Location-Routing Problems," *Computers & Operations Research*, Vol. 29, No. 10, 2002, pp. 1393-1415. doi:10.1016/S0305-0548(01)00038-7

[71] C. Ting, S. Li and C. Lee, "On the Harmonious Mating Strategy through Tabu Search," *Information Sciences*, Vol. 156, No. 3-4, 2003, pp. 189-214. doi:10.1016/S0020-0255(03)00176-2

[72] P. Greistorfer, "A Tabu Scatter Search Metaheuristic for the Arc Routing Problem," *Computers and Industrial Engineering*, Vol. 44, No. 2, 2003, pp. 249-266. doi:10.1016/S0360-8352(02)00178-X

[73] R. Russell and W. Chiang, "Scatter Search for the Vehicle Routing Problem with Time Windows," *European Journal of Operational Research*, Vol. 169, No. 2, 2006, pp. 606-622. doi:10.1016/j.ejor.2004.08.018

[74] E. Taillard, "Robust Taboo Search for the Quadratic Assignment Problem," *Parallel Computing*, Vol. 17, 1991, pp. 433-445. doi:10.1016/S0167-8191(05)80147-4

[75] F. Glover, M. Laguna and C. Reeves, "Tabu Search," In: C. Reeves, Ed., *Modern Heuristic Techniques for Combinatorial Problems*, Blackwell Scientific Publications, Oxford, 1993.

[76] J. Crino, J. Moore, J. W. Barnes and W. Nanry, "Solving the Theater Distribution Vehicle Routing and Scheduling Problem Using Group Theoretic Tabu Search," *Mathematical and Computer Modelling*, Vol. 139, No. 6-8, 2004, pp. 599-616. doi:10.1016/S0895-7177(04)90543-2

