A Review of the Tabu Search Literature on Traveling Salesman Problems

Sumanta Basu
Diptesh Ghosh

W.P. No. 2008-10-01
October 2008

INDIAN INSTITUTE OF MANAGEMENT
AHMEDABAD – 380015
INDIA

A Review of the Tabu Search Literature on Traveling Salesman Problems

Sumanta Basu[1]
Diptesh Ghosh[2]

Abstract

The Traveling Salesman Problem (TSP) is one of the most widely studied problems in combinatorial optimization. It has long been known to be $\mathscr{NP}$-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, point out trends in it, and bring out some interesting research gaps in this literature.

Keywords: Traveling Salesman Problems, Tabu Search

1 Introduction

Many managerial problems, like 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., [37]) we are given a weighted graph with n nodes, and are 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. The number n is commonly referred to as the size of the TSP. TSPs serve as a representation of many managerial problems, especially in logistics. 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 $\mathscr{NP}$-complete (see [35]), and has been a testbed for theoretical and computational studies ever since. Rich classes of benchmark problems exist for the TSP (see e.g., [44, 34]), 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 $\mathscr{NP}$-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. The term heuristic is normally used to describe an approximate solution method that is intended for one particular optimization problem, while the term metaheuristic is used to describe a more

[1]Wipro Technologies, Bangalore, India; Email: sumanta.basu2@wipro.com
[2]IIM Ahmedabad, India; Email: diptesh@iimahd.ernet.in

general framework that can be easily adapted to solve a wide variety of problems. 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 possibly the most widely used and most successful metaheuristic procedure to solve combinatorial optimization problems. Figure 1 shows the number of papers published every year on tabu search implementations for the TSP and related problems. It

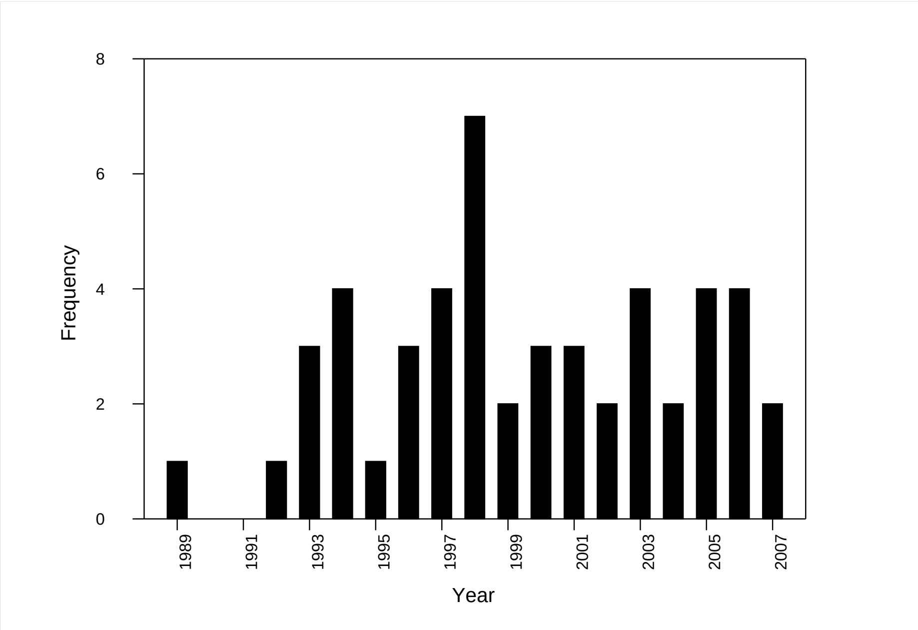

Figure 1: Papers dealing with tabu search implementations for the TSP and related problems

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 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 generation moves that it considers forbidden, or tabu (hence the name, tabu search) and ignores solutions that can be reached only through tabu moves while searching the neighborhood of a solution. Once a move enters the list of tabu moves, it stays there for a 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 like the TSP. 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 problems (VRP). We reviewed 58 papers on the application of tabu search to these problems. The papers that we reviewed mostly appeared in print in the last fifteen 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 Sizes Considered

44 papers describe experimental results of using tabu search on TSPs. Table 1 provides a summary of the maximum size of TSPs considered by different authors in the literature. It is interesting to note that even though metaheuristics are meant to handle large problems, more than half the authors deal only with TSPs with up to 100 nodes. Most of these authors have used the benchmark problems described in [53] which contain 56 instances with 100 nodes each. Till date, we have only seen three papers ([7], [16], and [32]) that implement tabu search on TSPs with between 500 and 1035 nodes.

Table 1: Papers implementing tabu search on TSPs and problem sizes they consider

Number of nodes	Before 1996	1996 – 2000	2001 – 2005	After 2006	Total
100 or less	[22, 39, 50]	[2, 4, 8, 13, 20, 21, 26, 30, 41, 43, 51, 62]	[11, 31, 32, 33, 36, 56]	[6, 38]	23
101 – 150	—	[3]	[12, 15]	[49]	4
151 – 200	[24, 42, 46]	[5, 9, 63]	—	[1, 38]	8
201 – 250	[17]	—	—	[18]	2
251 – 300	—	[27, 25]	[15]	—	3
301 – 350	—	—	—	—	—
351 – 400	—	[45]	—	[40]	2
401 – 450	[59]	[55]	[14, 57]	—	4
451 or more	—	[61]	[7, 16, 32]	—	4

Numbers refer to citation numbers in References list.

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 practice. 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 Generation of Initial Solutions

Tabu search is an improvement heuristic. It needs to start with a feasible tour in the graph describing the TSP. 32 of the 58 papers surveyed describe the process used to obtain initial solutions. In these, 8 methods were used to generate initial solutions. These 8 methods are given below:

GRASP: (General Randomized Adaptive Search Process) This method 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.

RandIns: (Randomized insertion) This method 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.

NrNbr: (Nearest neighbor) This method starts with a partial tour. It then uses a modified Prim's algorithm to find a node in the graph which is not in the partial tour, and is nearest to a node already existing in the partial tour. It then 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 (see e.g., [8]).

NrMrgr: (Nearest merger) This method forms subtours in the graph describing the TSP. It then modifies Kruskal's algorithm to coalasce the subtours into a complete tour.

PrNrNbr: (Probabilistic nearest neighbor) This is a probabilistic version of nearest neighborhood heuristic. It starts with a partial tour. Then, for each node already in the partial tour, the neighboring nodes are assigned probabilities of being connected to the node. The probability is higher for a node which has lower cost of joining to the node being considered. The heuristic then chooses one of the neighboring nodes probabilistically and adds it to the partial tour. It stops when all the nodes in the graph are included in the tour.

Sweep: (Sweep method) This heuristic is suited for TSPs defined on a plane. An initial node is chosen, and all the other nodes are order according to the angles they make with the starting node. They are then added to the tour in the same order.

Solomon: (Solomon's heuristic) It is a combination of the nearest neighbor heuristic and the sweep heuristic.

GENI: (GENeralized Insertion) Developed by [23], it constructs a tour by inserting vertices between two non-successive nodes in a partial tour.

Table 2: Methods of forming initial solutions for tabu search

Method	Before 1996	1996 – 2000	2001 – 2005	After 2006	Total
GRASP	—	[20]	—	—	1
RandIns	—	[4, 55, 5, 41]	[15, 57, 52]	[58]	8
NrNbr	—	[8, 13, 20, 62, 5, 63]	[10, 12, 36, 31, 52]	[38, 40, 49]	14
NrMrgr	—	—	[7]	—	1
PrNrNbr	—	—	[7]	—	1
Sweep	—	—	[14]	[40, 49, 18]	4
Solomon	[22]	[43]	—	—	2
GENI	[24]	[25]	—	—	2

Numbers refer to citation numbers in References list.

Table 2 summarizes the use of the 8 methods to form initial solutions in the tabu search literature on the TSP. We observe that the nearest neighborhood heuristic (NrNbr) is the most common choice followed by randomized insertion heuristic (RndIns). We feel that these heuristics are chosen because they are the easiest to implement for TSPs, and also because given enough execution time, tabu search is thought to be powerful enough to obtain good solutions regardless of the initial solution presented to the search process.

4 Choice of 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 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, 42 papers describe the moves that they use in tabu search implementation. However, the only paper that emphasizes the influence of the choice of moves on the solution generated is [42], 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.

2-opt move: In this move, two edges are removed from an existing tour and two new edges are inserted as to create a new tour without creating subtours. Figure 2 demonstrates a 2-opt move for TSPs.

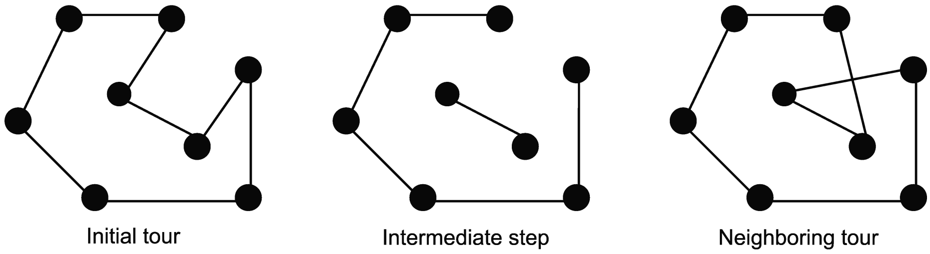

Figure 2: A 2-opt move

r-opt move: It is a generalization of the 2-opt move where $r > 2$ edges are involved in deletion/addition operation. Figure 3 demonstrates a 3-opt move (i.e, a r-opt move with $r = 3$) for TSPs.

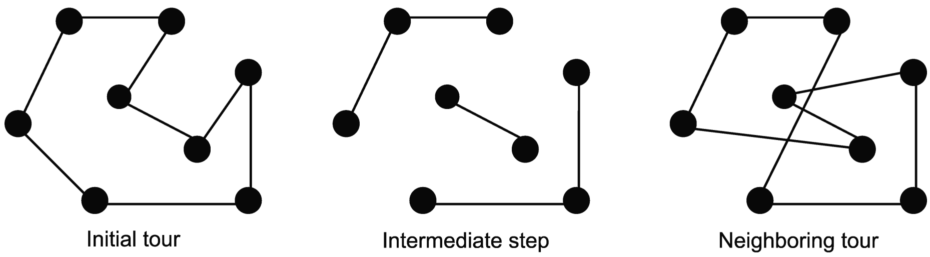

Figure 3: A 3-opt move

Vertex insertion: In this move, a vertex is removed from an existing tour and is inserted between two other vertices to create a new tour. Figure 4 demonstrates a vertex insertion move for TSPs in which the vertex kept out of the tour in the intermediate step is re-inserted in the last step.

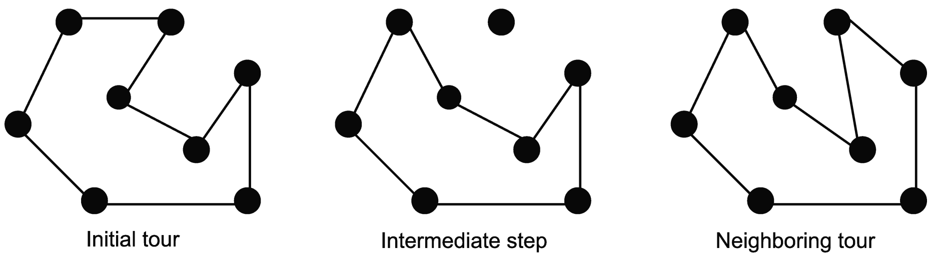

Figure 4: A vertex insertion move

Vertex Exchange: In this move the positions of two vertices are interchanged to create a new tour from the existing one. Figure 5 demonstrates a vertex exchange move for TSPs in which the two vertices colored white in the intermediate step are exchanged in the last step.

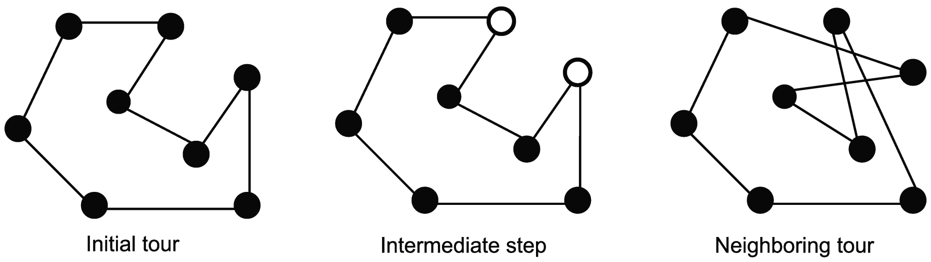

Figure 5: A vertex exchange move

Generalized Insertion (GENI): In a GENI move, one vertex is removed from the tour, and is joined to two vertices that were not adjacent in the original tour. The nodes displaced from the tour due to this operation are introduced at appropriate positions to complete the tour. For example, in Figure 6, node 2 is removed from the original tour, and is made adjacent to nodes 3 and 5. Node 4 which is consequently displaced is reintroduced in the tour, between nodes 6 and 7 to complete the tour.

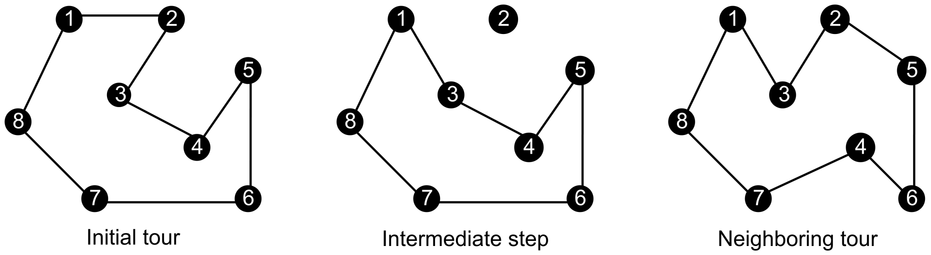

Figure 6: A GENI move

λ-interchange: Suggested by [42], 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.

Or-opt: This move is a modification of the r-opt move. It was proposed by Or in 1976. 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.

Table 3 summarizes the choice of moves with time. From the table, we see that the three moves that are most commonly used are vertex insertion, vertex exchange, and 2-opt. 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.

5 Choice of Short Term Memory Structures

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 number of tabu search iterations. This number is called its tabu tenure. Tabu search implementations either keep

Table 3: Types of moves used in tabu search implementations for TSP

Moves used	Before 1996	1996 – 2000	2001 – 2005	After 2006	Total
2-opt	[39, 42, 46, 22]	[43, 55, 62, 5, 27]	[12, 31, 57]	[40, 6, 58]	15
r-opt	—	[45, 55, 62]	—	—	3
Vertex insertion	[42, 50, 24]	[26, 9, 14, 61, 63, 5]	[41, 10, 16, 15, 36, 31, 7, 52, 56, 57]	[38, 40, 49, 6, 58]	24
Vertex exchange	—	[3, 9, 61, 5, 63, 41]	[10, 36, 31, 7, 32, 56, 57]	[38, 40, 49, 6]	17
GENI	—	[14]	[7]	[18]	3
λ-interchange	[42, 59]	[13]	[32]	—	4
Or-opt	[22]	[43, 55]	[32]	—	4

Numbers refer to citation numbers in References list.

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 non-random tabu tenures, and the last one as random tabu tenure.

Papers implementing non-random tabu tenures: 30 of the papers that we surveyed dealt with non-random tabu tenures. Of those, 16 dealt with tabu tenures that were fixed, and the other 14 papers dealt with tabu tenures that varied with the number of iterations already performed by tabu search, or with the instance size. Table 4 provides a summary of the papers that dealt with fixed tabu tenures. Table 5 presents a grouping of papers by the parameter that they use to determine the tabu tenure.

Table 4: Range of fixed tabu tenure used in tabu search implementations

Tenure range	Before 1996	1996 – 2000	2001 – 2005	After 2006	Total
1–4	[17]	—	—	—	1
5–9	[17, 22]	[43, 3, 2]	[64, 7, 52]	—	8
10–14	[59]	[3, 2]	[10, 64, 7, 32]	—	7
15–19	—	[3, 2]	[31]	—	3
20–24	—	[2]	[10, 57]	[1, 58]	5
25 or more	—	—	[10, 36]	—	2

Numbers refer to citation numbers in References list.

Table 5: Dependence of non-random tabu tenures on parameters in tabu search implementations

Tenure depends on	Before 1996	1996 – 2000	2001 – 2005	After 2006	Total
No. of TS iterations	[42]	[4, 55, 41]	—	[40, 48]	6
Size of the instance	[39, 42]	[41, 62, 13]	[60, 29, 16]	[18, 40, 49]	11

Numbers refer to citation numbers in References list.

From Table 4 we see that the range of values for the tabu tenure lies mostly in between 5 to 25 across years. Further, more than 50% of these papers use tabu tenures between 5 to 15. In papers like [17], [2], and [64], the authors conducted experiments to see the effect of different tenure values on solution quality. In [2], the authors showed that non-random tabu tenures 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 [2]. In [64], 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 a tabu tenure irrespective of the problem instance vary the tabu tenure based on the instance size. A relatively small number of papers make the tabu tenure dependent on the number of iterations already executed by tabu search. Only one of the papers that we surveyed ([42] for VRPs), describes an elaborate mechanism for determining the tabu tenure. In [42], 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 gives 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., [39], [50]) 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 [16], [14], [60], [18], and [49], the functional approach is followed to determine non-random tabu tenures. In [16] and [14], the form of the function is logarithmic, while in [49], both logarithmic and linear equations 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 non-random tabu tenures over different problem sizes appears in [62]. In it, 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.

Papers implementing random tabu tenures: The change of tabu tenure from deterministic to random was initiated in [54] for the quadratic assignment problem. Among the papers we surveyed, 20 papers used random tabu tenures. In all of these papers, tabu tenures are picked randomly from a uniform distribution whose support was fixed. In 9 papers, the limits of the distributions from which the tabu tenure is drawn varied with the instance size.

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 [24] and was motivated by a suggestion in [28].

Table 6: Random independent tabu tenure ranges used in tabu search implementations

Range used	Before 1996	1996 – 2000	2001 – 2005	After 2006	Total
5–10	[24]	[45, 61, 5, 30]	[12, 56]	[38]	8
10–20	—	[5]	[10]	[40]	3
20–30	—	[51]	[10]	—	2

Numbers refer to citation numbers in References list.

Reference [46] influenced the trend of making tabu tenures depend on problem and search characteristics. In this paper, the authors modified the tabu tenure based on solution values in previous iterations. After this, 9 papers have appeared in the literature in which the suport of the distribution from which the tabu tenure is chosen is based on the problem being solved. All of them except [45] and [40]) considered instances with between 50 and 150 nodes. Reference [40] 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 co-authors. 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 [26], [27], and [8], the support of the distribution from which tabu tenures are chosen were directly proportional to the problem size, while in [14] and [15] 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 Structures

Intermediate term memory structures are used in tabu search to intensify the search by restricting it to promising regions of the solution space. 18 of the papers we reviewed implemented intermediate term memory structures. We classify the strategies used into the following 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: In the current iteration, 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.

Table 7: Intermediate term memory management strategies used in tabu search implementations

Strategy used	Before 1996	1996 – 2000	2001 – 2005	After 2006	Total
Strategy IT1	[46]	[43, 4, 13, 5]	[19, 57]	[40, 58]	9
Strategy IT2	[24]	[45, 55]	[56]	[48, 49]	6
Strategy IT3	[47]	[55]	[56]	—	3
Strategy IT4	[50, 46]	[21]	—	—	3

Numbers refer to citation numbers in References list.

Table 7 presents information about the use of the four strategies. We see that the first two strategies have been used in more than 80% 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., [46], [55], [56]), more than one of these strategies are used together.

7 Choice of Long Term Memory Structures

Tabu search uses long term memory structures to diversify the search to new regions in the solution space. We found 23 papers that used long term memory structures to achieve diversification. We list the strategies used below.

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

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

Strategy LT3: Diversification is achieved 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: Changing the stopping criterion to allow more non-improving moves.

Strategy LT5: 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.

Table 8: Long term memory management strategies used in tabu search implementations

Strategy used	Before 1996	1996 – 2000	2001 – 2005	After 2006	Total
Strategy LT1	—	[4, 8, 13, 55, 9, 30]	[10, 19, 7, 56]	[40]	11
Strategy LT2	[24]	[14]	[16, 15]	[49]	5
Strategy LT3	—	[43]	[12, 56]	—	3
Strategy LT4	—	[45, 21]	—	—	2
Strategy LT5	—	—	[57]	[58]	2

Numbers refer to citation numbers in References list.

Table 8 presents the usage of long term memory structures in the tabu search literature on TSPs. It is clear from the table that Strategy LT1 is the most commonly used strategy for using long term memory structures.

Reference [24] used Strategy LT2 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 [14], [16], [15], and [49] among others.

Reference [56] 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 [12], edges do not enter the tabu list if the cost of the tour is within 10% of the cheapest tour up to that iteration. Reference [43]'s implementation in used Or-opt moves instead of the usual variation on the 2-opt move when it could not improve the best tour for a specific number of iterations.

References [57] and [58] used influence measures to help diversify their tabu search. By inserting this expression in the objective function, they restricted moves to similar kinds of tours.

8 Choice of Aspiration Criteria

Aspiration criteria are criteria which if satisfied, permits tabu search to make use of tabu moves to reach neighboring solutions. 30 of the papers that we reviewed used aspiration criteria. Although the design of aspiration criteria can be finely tuned, all the papers except [14] 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. [14] developed attribute specific adaptive aspiration level functions instead of taking solution values.

9 Summary of Findings

The following is a summary of the trends that we observe in the literature on tabu search implementations for the TSP and allied problems.

1. Most authors have mostly considered tabu search implementations on symmetric TSPs described on complete graphs.

2. The size of problems that authors deal with have not shown much increase (Table 1). In fact even in more recent papers, authors concentrate on problems with less than 100 nodes. One of the reasons for this could be the lack of benchmark instances of large size. Such instances are now available (see [34]).

3. None of the papers that we reviewed used all of the features of tabu search in their tabu search implementations. Table 9 summarizes the data on the features used in tabu search implementations in the literature. Of course papers may have used tabu search features without discussing them in detail in the presentation, so this table is more of a list of features that authors have discussed in the papers on tabu search applied to TSPs.

Table 9: Tabu search features used in tabu search implementations

Component	Before 1996	1996 – 2000	2001 – 2005	After 2006	Total
Fixed tabu tenure	5	10	12	4	33
Random tabu tenure	3	13	5	2	23
Intermediate term memory structure	5	5	2	4	16
Long term memory structure	4	7	9	3	23
Aspiration criteria	4	10	10	5	29

4. Authors do not use very sophisticated methods to obtain initial solutions 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.

5. The issue of deciding tabu tenures have not received adequate attention in the literature. First, papers have looked at only linear and logarithmic dependences of the tabu tenure on problem size. They provide no 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.

6. Authors prefer moves that are simple to implement and which give rise to small neighborhoods (Table 3). In the last few years, they have only 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.

7. Most authors prefer to use short term memory structures in their tabu search implementations (Tables 4 through 6). In aggregate more authors have preferred fixed tabu tenures over random tabu tenures. This preference seems to have increased in recent years. Among authors who have chosen to use fixed tabu tenures, more authors are beginning to tailor the tabu tenures

based on the characteristics of the TSP being solved and the stage of execution that the search is in.

8. Relatively few papers use intermediate and long term memory structures (Tables 7 and 8). The trend to use these structures have not picked up over time. More papers use long term memory structures than intermediate term memory structures. Implementations may be conceived in which intermediate and long term memory structures are not used throughout, but switched on or off at particular points in the search process. Such implementations could be experimented with, and criteria for switching these structures on or off could be experimentally arrived at. For long term memory structures (especially Strategies LT1 and LT2), more research needs to be done to arrive at a good functional form of the penalty function.

9. Aspiration criteria is being used only in 50% of the tabu search implementations. There does not seem to be much work on fine-tuning the design of aspiration criteria in these implementations.

10. More advanced features of tabu search, such as strategic oscillation, use of elite candidate lists, etc. have not found much use in the tabu search literature on TSPs so far.

References

[1] D. Ahr and G. Reinelt. A tabu search algorithm for the minmax k-chinese postman problem. *Computers & Operations Research*, 33:3403–3422, 2006.

[2] A. Amberg, W. Domschke, and S. Voss. Multiple center capacitated arc routing problems: A tabu search algorithm using capacitated trees. *European Journal of Operational Research*, 124:360–376, 2000.

[3] P. Augerat, Belenguer, Benavent E. J.M., A. Corbern, and D. Naddef. Separating capacity constraints in the CVRP using tabu search. *European Journal of Operational Research*, 106:546–557, 1998.

[4] 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*, 5:109–122, 1997.

[5] G. Barbarosoglu and D. Ozgur. A tabu search algorithm for the vehicle routing problem. *Computers and Operations Research*, 26:255–270, 1999.

[6] N. Bianchessi and G. Righini. Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery. *Computers & Operations Research*, 34:578–594, 2007.

[7] A.L. Bouthillier and T.G. Crainic. A cooperative parallel meta-heuristic for the vehicle routing problem with time windows. *Computers & Operations Research*, 32:1685–1708, 2005.

[8] J. Brandão and A. Mercer. A tabu search algorithm for the multi-trip vehicle routing and scheduling problem. *European Journal of Operational Research*, 100:180–191, 1997.

[9] J.C.S. Brandão and A. Mercer. The multi-trip vehicle routing problem. *The Journal of the Operational Research Society*, 49:799–805, 1998.

[10] A.V. Breedam. Comparing descent heuristics and metaheuristics for the vehicle routing problem. *Computers & Operations Research*, 24:289–315, 2001.

[11] O. Brysy and M. Gendreau. Tabu search heuristics for the vehicle routing problem with time windows. Technical report, SINTEF Applied Mathematics, Department of Optimisation, Oslo, Norway, 2001.

[12] I.M. Chao. A tabu search method for the truck and trailer routing problem. *Computers & Operations Research*, 29:33–51, 2002.

[13] W.C. Chiang and R.A. Russell. A reactive tabu search metaheuristic for the vehicle routing problem with time windows. *INFORMS Journal of Computing*, 9:417–430, 1997.

[14] J.F. Cordeau, M. Gendreau, and G. Laporte. A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks*, 30:105–119, 1998.

[15] J.F. Cordeau and G. Laporte. A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transportation Research Part B: Methodological*, 37:579–594, 2003.

[16] J.F. 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*, 52:928–936, 2001.

[17] 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*, 41:359–383, 1993.

[18] B. Crevier, J.F. Cordeau, and G. Laporte. The multi-depot vehicle routing problem with inter-depot routes. *European Journal of Operational Research*, 176:756–773, 2007.

[19] J.R. Crino, J.T. Moore, Barnes, J.W., and W.P. Nanry. Solving the theater distribution vehicle routing and scheduling problem using group theoretic tabu search. *Mathematical and Computer Modelling*, 139:599–616, 2004.

[20] R.L. Daniels, J.L. Rummel, and R. Schantz. A model for warehouse order picking. *European Journal of Operational Research*, 105:1–17, 1998.

[21] P.M. Franca, N.M. Sosa, and V. Pureza. An adaptive tabu search algorithm for the capacitated clustering problem. *International Trasactions in Operations Research*, 6:665–678, 1999.

[22] B.L. Garcia, J.Y. Potvin, and J.M. Rousseau. A parallel implementation of the tabu search heuristic for vehicle routing problems with time window constraints. *Computers & Operations Research*, 21:1025–1033, 1994.

[23] M. Gendreau, A. Hertz, and G. Laporte. New insertion and postoptimization procedures for the traveling salesman problem. *Operations Research*, 40:1086–1094, 1992.

[24] M. Gendreau, A. Hertz, and G. Laporte. A tabu search heuristic for the vehicle routing problem. *Management Science*, 40:1276–1290, 1994.

[25] M. Gendreau, G. Laporte, and F. Semet. A tabu search heuristic for the undirected selective travelling salesman problem. *European Journal of Operational Research*, 106:539–545, 1998.

[26] M. Gendreau, G. Laporte, and R. Sguin. A tabu search heuristic for the vehicle routing problem with stochastic demands and customers. *Operations Research*, 44:469–477, 1996.

[27] M. Gendreau, G. Laporte, and D. Vigo. Heuristics for the traveling salesman problem with pickup and delivery. *Computers & Operations Research*, 26:699–714, 1999.

[28] F. Glover and M. Laguna. *"Tabu Search," Modern Heuristic techniques for Combinatorial Problems.* Blackwell Scientific Publications, 1993.

[29] P. Greistorfer. A tabu scatter search metaheuristic for the arc routing problem. *Computers and Industrial Engineering*, 44:249–266, 2003.

[30] A. Hertz, G. Laporte, and M. Mittaz. A tabu search heuristic for the capacitated arc routing problem. *Operations Research*, 48:129–135, 2000.

[31] S.C. Ho and D. Haugland. A tabu search heuristic for the vehicle routing problem with time windows and split deliveries. *Computers & Operations Research*, 31:1947–1964, 2004.

[32] J. Homberger and H. Gehring. A two-phase hybrid metaheuristic for the vehicle routing problem with time windows. *European Journal of Operational Research*, 162:220–238, 2005.

[33] S. Ichoua, M. Gendreau, and J.Y. Potvin. Vehicle dispatching with time-dependent travel times. *European Journal of Operational Research*, 144:379–396, 2003.

[34] D.S. Johnson, G. Gutin, L.A. McGeoch, A. Yeo, W. Zhang, and A. Zverovich. *The Traveling Salesman Problem and Its Variations*, chapter Experimental Analysis of Heuristics for the ATSP. Kluwer Academic Publishers, 2002.

[35] R.M. Karp. *Reducibility among Combinatorial Problems*, pages 85–103. Plenum Press, 1972.

[36] H.C. Lau, M. Sim, and K.M. Teo. Vehicle routing problem with time windows and a limited number of vehicles. *European Journal of Operational Research*, 148:559–569, 2003.

[37] E.L. Lawler, J.K. lenstra, K. Rinnooy, and D.B. Shmoys. *The Traveling Salesman Problem: A guided tour of combinatorial optimization.* Wiley-Interscience Publication, 1985.

[38] C.K.Y. Lin and R.C.W. 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*, 175:1833–1849, 2006.

[39] M. Malek, Guruswamy M., Pandya M., and Owens H. Serial and parallel simulated annealing and tabu search algorithms for the traveling salesman problem. *Annals of Operations Research*, 21:59– 84, 1989.

[40] F.A.T. Montane and R.D. Galvao. A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. *Computers & Operations Research*, 33:595–619, 2006.

[41] W.P. Nanry and J.W. Barnes. Solving the pickup and delivery problem with time windows using reactive tabu search. *Transportation Research Part B: Methodological*, 34:107–121, 2000.

[42] I.H. Osman. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research*, 41:421–451, 1993.

[43] J.Y. Potvin, T. Kervahut, B.L. Garcia, and J.M. Rousseau. The vehicle routing problem with time windows - part I: Tabu search. *INFORMS Journal of Computing*, 8:158–164, 1996.

[44] G. Reinelt. Tsplib - a traveling salesman problem library. *INFORMS Journal of Computing*, 3:376–384, 1991.

[45] J. Renaud, G. Laporte, and F.F. Boctor. A tabu search heuristic for the multi-depot vehicle routing problem. *Computers & Operations Research*, 23:229–235, 1996.

[46] Y. Rochat and F. Semet. A tabu search approach for delivering pet food and flour in Switzerland. *The Journal of the Operational Research Society*, 45:1233–1246, 1994.

[47] Y. Rochat and E.D. Taillard. Probabilistic diversification and intensification in local search for vehicle routing. *Journal of Heuristics*, 1:147–167, 1995.

[48] R.A. Russell and W.C. Chiang. Scatter search for the vehicle routing problem with time windows. *European Journal of Operational Research*, 169:606–622, 2006.

[49] S. Scheuerer. A tabu search heuristic for the truck and trailer routing problem. *Computers & Operations Research*, 33:894–909, 2006.

[50] F. Semet and E. Taillard. Solving real-life vehicle routing problems efficiently using taboo search. *Annals of Operations Research*, 41:469–488, 1993.

[51] Y.M. Sharaiha, M. Gendreau, G. Laporte, and I.H. Osman. A tabu search algorithm for the capacitated shortest spanning tree problem. *Networks*, 29:209–223, 1997.

[52] J. Shen, F. Xu, and P. Zheng. A tabu search algorithm for the routing and capacity assignment problem in computer networks. *Computers & Operations Research*, 32:2785–2800, 2005.

[53] M. Solomon. Algorithms for the vehicle routing and scheduling problem with time window constraint. *Operations Research*, 33:254–265, 1987.

[54] E. Taillard. Robust taboo search for the quadratic assignment problem. *Parallel Computing*, 17:433–445, 1991.

[55] 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*, 31:170–186, 1997.

[56] H. Tang and E.M. Hooks. A tabu search heuristic for the team orienteering problem. *Computers & Operations Research*, 32:1379–1407, 2005.

[57] C.D. Tarantilis. Solving the vehicle routing problem with adaptive memory programming methodology. *Computers & Operations Research*, 32:2309–2327, 2005.

[58] C.D. Tarantilis and C.T. Kiranoudis. A flexible adaptive memory-based algorithm for real-life transportation operations: Two case studies from dairy and construction sector. *European Journal of Operational Research*, 179:806–822, 2007.

[59] 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.

[60] C.K. Ting, S.T. Li, and C. Lee. On the harmonious mating strategy through tabu search. *Information Sciences*, 156:189–214, 2003.

[61] P. Toth and D. Vigo. The granular tabu search and its application to the VRP. Technical report, University of Bologna, 1998.

[62] S. Tsubakitani and J.R. Evans. Optimizing tabu list size for the traveling salesman problem. *Computers & Operations Research*, 25:91–97, 1998.

[63] D. Tuzun and L.I. Burke. A two-phase tabu search approach to the location routing problem. *European Journal of Operational Research*, 116:87–99, 1999.

[64] T.H. Wu, C. Low, and J.W. Bai. Heuristic solutions to multi-depot location-routing problems. *Computers & Operations Research*, 29:1393–1415, 2002.