Control and Cybernetics

vol. **29** (2000) No. 3

Selecting a good configuration of one-way and two-way routes using tabu search†

by

Zvi Drezner* and Said Salhi**

*Department of Management Science/Information Systems
California State University-Fullerton
Fullerton, CA 92834, U.S.A.

**Management Mathematics Group
School of Mathematics and Statistics
University of Birmingham
Birmingham, United Kingdom.

Abstract: The problem of designing a near optimal configuration of a system of one-way and two-way routes is investigated. Each arc of the network can be designed as either a two-way arc or a one-way arc in one of the two directions. The traffic speed on a one-way arc is faster than the speed on a two-way arc by a given factor. The problem is to design a network which minimizes total travel time between all pairs of nodes by the proper selection of one-way and two-way arcs. Efficient implementations of the metaheuristic tabu search are designed for solving this network design problem. These approaches are tested on a set of network problems with encouraging results.

Keywords: network design, tabu search, metaheuristics.

1. Introduction

In recent years, network design problems have received increasing attention, see Marcotte (1983) and Magnanti and Wong (1984). The specific issue of designing a network with some of the links being one-way routes and some two-way routes was presented in Drezner and Wesolowsky (1997). The reader is referred to Drezner and Wesolowsky (1997) for a review of related papers.

†**Acknowledgment:** This research was done while the first author was visiting the Graduate School of Management, University of California-Irvine, and the second author was visiting the Department of Management Science/Information Systems, California State University-

Network design problems arise in several applications, varying from industrial installations requiring the transport of materials to communication networks. In this paper we address the problem of designing optimum configurations for a one- and two-way routes system. Our problem is defined as follows: A network of roads is given. Each road can be designed either as a two-way street or a one-way street. The objective is to optimize the configuration of one-way streets in a network for a given efficiency ratio for one-way travel. Making streets or routes one way is well known to increase the capacity of the artery, while at the same time increasing the travel distances for some flows. Many congested urban areas, for example, employ one-way streets. One-way traffic is also useful in many industrial installations requiring the transport of materials. Communication networks also use one-way flows. The model can be applied in a wide variety of congested traffic, shop-floor flow and communication situations. Drezner and Wesolowsky (1997) were the first ones to study this problem. They designed various heuristic procedures for its solution.

The aim of this study is two-fold: (i) to solve this new problem using a metaheuristic such as tabu search, and (ii) to put forward efficient implementations of tabu search. To achieve these objectives, we provide new, better results for a set of test problems, and suggest novel approaches to the implementation of the tabu search.

In the next section we define the problem. In the following sections we describe the tabu search technique in general and the specific parameters used in our analysis. We conclude with reports on the experimentation results for some test problems.

2. Notation and definitions

Let us introduce the following notations:

- n number of nodes in the network.
- m number of arcs in the network.
- S the set of all arcs. Each member in S is defined as an ordered pair (i,j) of nodes. Each arc is defined twice in both directions. The cardinality of S is $2m$.
- d_{ij} defined for all $(i,j) \in S$ is the length of the arc (i,j) when it is a two-way street. It is possible that $d_{ij} \neq d_{ji}$.
- α is the factor by which distances are multiplied when an arc is turned into a one-way street ($\alpha < 1$). $\frac{1}{\alpha}$ is the factor by which the speed is increased when a two-way street is turned into a one way street.
- w_{ij} is the number of vehicles traveling from origin i to destination j.
- W is the set of all origin-destination pairs (i,j) for which $w_{ij} > 0$.
- z_{ij} for $(i,j) \in S$ is a variable defining a particular one-way configuration for the network. $z_{ij} = 0$ if the arc from i to j is not available, and $z_{ij} = 1$ if the arc is available.

$s_{ij}(Z)$ is the shortest distance from i to j for a network defined by Z.
$F(Z)$ is the objective function for a given Z:

$$F(Z) = \sum_{(i,j)\in W} w_{ij}s_{ij}(Z). \tag{1}$$

An efficient procedure for the calculation of $F(Z)$ is detailed in Drezner and Wesolowsky (1997). If there is no feasible route between two nodes, $F(Z)$ is returned as a large number.

Various heuristic approaches of the greedy type are studied in Drezner and Wesolowsky (1997). The recommended procedure is termed there as "the modified algorithm". In this paper we refer to it as the "greedy" heuristic. The main steps of the greedy heuristic are as follows:

The Greedy Heuristic

1. The starting solution is all two-way streets.
2. Each arc i–j can assume three values: (i) two way street, (ii) one way from i to j, and (iii) one way from j to i. The direction of each arc can be changed in two ways. There are therefore $2m$ individual changes in arc directions. The $2m$ possible changes are checked starting from a randomly selected arc. Arcs are checked consecutively in a circular way until the arc preceding the one we started with is reached.
3. Each time a better solution is found, it is accepted as the next solution and a new iteration begins.
4. If none of the $2m$ possible changes produces a better solution, the algorithm terminates.

3. Tabu search methods

Tabu search methods were proposed by Glover (1986). Tabu search concepts are derived from artificial intelligence where intelligent uses of memory help in exploiting useful historical information.

Tabu search heuristic, like descent (or greedy) heuristics, is a local search method that proceeds by examining some neighborhoods of the current solution. Unlike the descent method where the search terminates when there is no further improvement, tabu search allows the search to exploit inferior solutions as well as infeasible ones. This flexibility helps the search in getting out of local optimality when taking uphill moves or crossing infeasible regions. To avoid cycling, tabu search imposes some sort of tabu status to those attributes recently involved in the choice of the new solution.

To date, tabu search methods have proved successful in producing good results to several combinatorial optimization problems, namely scheduling, graph coloring and graph partitioning, telecommunication path assignment, vehicle routing, quadratic assignment, location problems, and others. The obtained references see Reeves, 1993; Glover et al., 1993; Osman and Laporte, 1996; and the book by Glover and Laguna, 1997).

3.1. The basic tabu search procedure

3.1.1. Initialization:

- Generate an initial solution, Z.
- Set the best current solution $Z_{best} = Z$.
- Define a neighborhood $N(Z)$.
- Define the tabu list and set values for the tabu size.
- Set the counter $iter = 0$ (current iteration).

3.1.2. The tabu list

- The tabu list contains a list of tabu moves.
- The length of the tabu list is termed "tabu size".
- When a move is performed (for example, z_{ij} is changed from 0 to 1), $z_{ij} = 0$ is added to the tabu list.
- A more efficient procedure is to record for each state (for example, $z_{ij} = 0$) the last iteration number for which it was changed from 0. A state is in the tabu list if the difference between the current iteration number and the recorded value is not greater than the tabu size.

3.1.3. Selection strategy:

1. Evaluate $F(Z')$ for $Z' \in N(Z)$ in a random order as the order in the greedy heuristic.
2. If $F(Z') < F(Z_{best})$ for any $Z' \in N(Z)$, terminate the iteration (i.e., do not proceed to check whether there are even better solutions in $N(Z)$), set $Z_{best} = Z^* = Z'$, and go to step 4.
3. Else, choose among the admissible solutions $Z' \in N(Z)$ the best one. Let the best one be Z^*. (An admissible solution is a solution whose move is not in the tabu list.)
4. Set $Z = Z^*$ and $iter = iter + 1$.
5. Update the tabu list and go to Step 1.

3.1.4. Stopping criterion:

A suitable stopping criterion such as a limit on the number of iterations is required.

3.1.5. Diversification (optional):

Apply some forms of diversification on well defined solutions. A diversification

These tabu search steps seem to be straightforward to implement. However, the success of the method is dependent on having a good insight of the problem. There are a few questions which usually help in devising a successful implementation. The success of tabu search methods depends on the choice of the tabu size, the definition of the neighborhood, how diversification schemes are developed and employed, the way previous solutions are identified, the efficiency of the computer program, and above all a good understanding of the problem. For further details on these issues, see Dammeyer and Voss (1993), Kelly, Laguna and Glover (1994), Glover (1989, 1995), Salhi (1996), Thomas and Salhi (1998), Battiti (1996).

4. Various parameters for the tabu search

4.1. The neighborhood

Since each arc i–j can assume three values, there are 3^m possible feasible solutions to the problem. The neighborhood to be searched consists of $2m$ possible changes in the road structure. These consist of changing a certain arc from the direction it is in the present iteration to one of the other two possible directions.

4.2. Starting solutions and stopping criterion

Two possible schemes are proposed for generating the set of starting solutions, and terminating the tabu search. In each, the greedy heuristic is applied K times producing K solutions. To have control over the running time of the tabu search, we adopted the following stopping rules:

- Each greedy solution defines a starting solution for the tabu search. The number of tabu iterations (a tabu iteration means moving to a neighbor) is set to the *same* number that were required for the greedy heuristic (a greedy iteration is finding a better solution). The number of iterations required for the greedy heuristic is an indicator for the complexity of the problem.
- The *best* greedy solution, selected from the K available solutions, is used as the starting solution in our tabu search methods. The number of iterations in the tabu search is K times the number of the iterations required for obtaining the best greedy solution.

Note that other measures, which are geared toward solution quality instead of computing time also exist. For instance, the use of non-improvement over the best solution after a permitted number of iterations, or no improvement in a certain number of successive iterations, etc. These stopping rules, though producing good results, can be too time consuming. The above two rules require the same total number of tabu iterations per problem and thus a similar

4.3. The tabu list

Two different schemes are proposed for handling the tabu list:

- Whenever a new best solution Z_{best} (including the final greedy solution) is found, the tabu list is emptied. We do not forbid previous tabu search moves leading to the best solution. We treat Z_{best} as if it were a starting solution for the tabu search.
- The tabu list was maintained throughout the search including the iterations performed by the greedy algorithm preceding the start of the tabu search.

4.4. The tabu size

The common way of defining the tabu size is to set it to a fixed value a priori (Osman and Salhi, 1996) or to choose the tabu size from a defined range (Skorin-Kapov, 1990). In this study we define the tabu size in a dynamic manner by letting the tabu size value change at each iteration. Two ways to determine the tabu size at each iteration are proposed:

4.4.1. Alternating tabu size value

The tabu size TS is alternated between two values TS_{min} and TS_{max} regardless of the change in the value of the objective function. It is similar, in principle, to the random selection of the tabu size at every iteration. This is also similar in concept to the systematic dynamic tabu tenure where the values alternatively increased and then decreased within a range (Glover and Laguna, 1997). In our experiments the alternating approach seems to work well. In our view, this may provide researchers with another facet of tabu search.

4.4.2. Varying the tabu size

- Two values constraining the tabu size are set. These are TS_{min} and TS_{max}.
- The tabu size, TS, in any given iteration is always bounded by $TS_{min} \leq TS \leq TS_{max}$.
- Whenever a new best solution is found, or when a diversification is performed, the tabu size TS is set to TS_{max}.
- Each time the value of the objective function increases, TS is decreased by one as long as it does not go below TS_{min}.
- Each time the value of the objective function decreases, the tabu size TS is increased by one as long as it does not exceed TS_{max}.

4.5. A diversification scheme

This module is introduced to guide the search in exploring other regions which not to discover any promising solution. It may not be easy to detect the right timing for diversification or to define the appropriate way of diversifying the search. One common approach is to start from a random solution or a solution which has as much dissimilarity as possible with respect to other solutions. In this study the following diversification procedure is suggested:

- The tabu size is varied as in Section 4.4.2.
- A diversification is performed when $TS = TS_{min}$ is reached. Since TS is increased by a down move in the objective function and decreased by an up move, then, in most cases, a diversification is performed when there are $TS_{max} - TS_{min}$ more up moves than down moves since the last diversification (or a new best solution found). An exception to this rule occurs when a down move in the value of the objective function occurs when $TS = TS_{max}$. This is very rare and may result in a diversification when the difference between up and down moves is less than $TS_{max} - TS_{min}$.
- The diversification diverts the search back to Z_{best} but forcing the search to choose a different path from there. This different path is achieved by defining a forbidden list which contains the first two moves previously performed from Z_{best}. This forbidden list is augmented with new forbidden moves whenever we diversify back to the same z_{best}.
- The first two exchanges which are chosen after either a diversification or when the last new best solution was found are put in a new list which we refer to as the forbidden list. Note that this forbidden list is different from the tabu list. This list is emptied when a new best solution is found, initially contains the first two moves after the best new solution is found, and is increased by two moves at every diversification.
- The solution is returned to Z_{best} while stopping the search from following the same path which was previously selected when Z_{best} was initially found. This means that the moves in the forbidden list are not allowed to be used. A similar approach was successfully applied by Thomas and Salhi (1998) when solving the project scheduling problem with limited resources.
- After each diversification, the tabu list is emptied except for the forbidden list.

5. The test problems and computational results

We used three scenarios to evaluate the performance of the proposed tabu search heuristics. In the first two scenarios, the medium and small test problems given by Drezner and Wesolowsky (1997) are used. The third scenario is a real road network based on the highway system in Orange County, California.

5.1. Investigational experimentation

We tried many possible strategies in the implementation of the tabu search.

Drezner and Wesolowsky (1997). All these problems have the same network with $n = 40$ nodes and $m = 99$ arcs. The five problems differ in the value of α. The neighborhood for these problems consists of 198 possible exchanges. Evaluating the value of the objective function for each member in the neighborhood took less than 0.03 seconds on a Pentium 166 MHz computer. Therefore, each iteration required less than 6 seconds of computer time. We performed the greedy algorithm $K = 8$ times. Therefore, if the best greedy solution is used for the tabu search, the number of iterations is $K = 8$ times the number of the iterations required by the greedy heuristic. We checked all the cross possibilities of the options for starting solutions, handling the tabu list, and determining the tabu size, as described above.

5.1.1. The strategies used

We concentrated on four main approaches:

Fixed Using a fixed length tabu list (TS), i.e., $TS_{min} = TS_{max}$.

Alt Alternating between TS_{min} and TS_{max} in consecutive iterations regardless of the values of the objective function in these iterations.

Div The tabu size was varied as described in Section 4.4.2 and a diversification performed as given in Section 4.5.

NoDiv Varying the tabu size as described in Section 4.4.2 but without using diversification.

Note that all approaches require about the same computer time because the number of iterations in the tabu search is set equal to the number of iterations in the greedy heuristic and thus was the same for all approaches.

We concluded from these tests that:

- Starting the tabu search from each terminal greedy solution is superior in most cases to selecting the best greedy solution and performing the tabu search with eight times the number of iterations.
- Emptying the tabu list whenever a new best solution is better than retaining the history of tabu moves in the majority of cases.
- We tested various fixed values of TS and various values for TS_{min} and TS_{max}. It was found that the values of 5% and 10% of the neighborhood size for TS_{min} and TS_{max}, respectively, are better than other values.
- The best performance was obtained by the alternating method. A close second was the diversification approach. However, most of the methods described above performed quite well.

5.1.2. Obtaining the benchmark solutions

Since the optimal solutions for the medium size problems are not known, we conducted an extensive computation to obtain high quality solutions which we

The alternating method was repeated one hundred times for each α to obtain a good solution for each case. The results are summarized in Table 1. The best solution that was found by the tabu search in these one hundred experiments, was run for additional 1000 iterations, using the alternating methods and then 1000 more iterations using the diversification method. This extensive tabu search resulted in the best solutions listed in Table 2. These solutions are used as the benchmark for comparison between various approaches, and they are depicted in Fig. 1.

α	Best† Known	Greedy						Tabu Search					
		Objective		Iterations		Time (Min.) per run		Objective		Total Iterations		Time (Min.) per run	
		Min	Avg.	Min	Avg.	Min	Avg.	Min	Avg.	Min	Avg.	Min	Avg.
0.5	109080	108560	110153	119	138	0.53	0.94	108264	109277	238	276	11.36	13.53
0.6	129122	128979	130459	105	126	0.55	0.94	128292	129322	210	251	9.51	12.14
0.7	147165	147529	148802	89	107	0.54	0.90	147012	148085	178	214	8.24	10.21
0.8	160608	160591	161112	51	66	0.53	0.77	160218	160836	102	132	4.60	6.41
0.9	166379	166379	166401	12	13	0.25	0.44	166379	166379	24	27	1.12	1.53

† Drezner and Wesolowsky (1997)

Table 1. Results of 100 runs of the alternating method

α	Best Known	Best Greedy	Benchmark
0.5	109080	108560	108253†
0.6	129122	128406	128144
0.7	147165	146775	146666
0.8	160608	160308	160192
0.9	166379	166379	166379

† A better solution of 108213 was obtained by starting with the $\alpha = 0.6$ solution and performing one greedy iteration.

Table 2. Using additional iterations on the best solution of Table 1

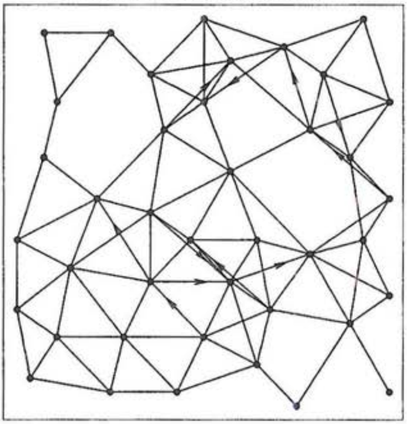
Since the tabu search required about 10–15 times more computer time than that required for the greedy heuristic, for comparison with the results in Tables 1 and 2 we repeated the greedy heuristic 1,000 times and report the best results in Table 2. Running the greedy heuristic 1,000 times required a similar run time to that required to get the benchmark results of Table 2.

5.2. Results of the three scenarios

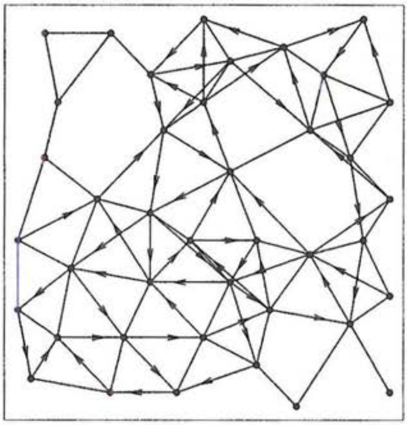
Scenario 1: Medium size problems

For comparison, five approaches were repeated ten times each ($K = 10$) (rather

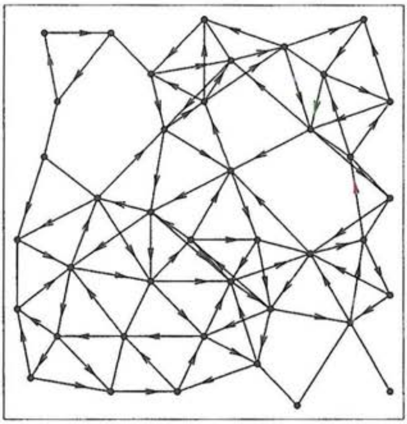
The Network for $\alpha = 0.9$ (Objective: 166379)



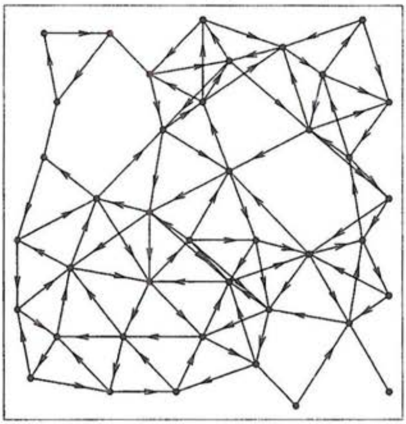
The Network for $\alpha = 0.8$ (Objective: 160192)



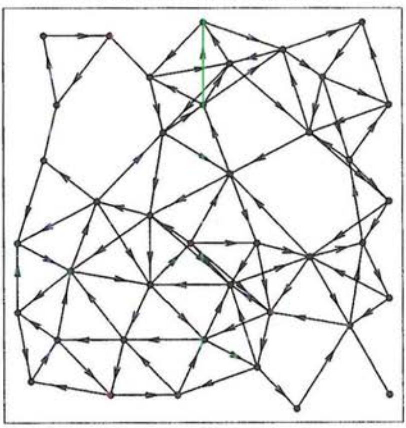
The Network for $\alpha = 0.7$ (Objective: 146666)



The Network for $\alpha = 0.6$ (Objective: 128144)



The Network for $\alpha = 0.5$ (Objective: 108253)



problem the best solution found in these ten experiments, and the average value of the objective function. The results are summarized in Table 3. Run times are not reported because they are the same as the times reported in Table 1. The best value in each column is marked in boldface. These results confirm the results of the preliminary experiments. The alternating method was best (or tied for best) in six out of ten measures. The diversification method was best in five, the no-diversification approach was best three times, and the fixed approach was best in four cases each.

Method	$\alpha = 0.5$		$\alpha = 0.6$		$\alpha = 0.7$		$\alpha = 0.8$		$\alpha = 0.9$	
	Min	Avg.	Min	Avg.	Min	Avg.	Min	Avg.	Min	Avg.
Benchmark	108213		128144		146666		160192		166379	
Alt	**108564**	109082	129047	129523	**147254**	**148020**	**160218**	160821	**166379**	**166379**
Div	108615	109163	**128718**	129433	147301	148111	**160218**	**160757**	**166379**	**166379**
NoDiv	108615	109113	128853	129545	147301	148127	**160218**	160781	**166379**	**166379**
Fixed (10)	108715	**109061**	129027	129609	**147254**	148204	160228	160822	**166379**	**166379**
Fixed (20)	108594	109091	128867	**129396**	147288	148114	**160218**	160768	**166379**	**166379**

Table 3. Ten repetitions of various tabu approaches

The best solution reported in Table 3 was allowed to continue for twice as many additional tabu search iterations. The results are summarized in Table 4. It turns out that the diversification approach was best in four cases out of five, while the alternating approach was best in only three cases. The other approaches tied for the best result only once.

Description	$\alpha = 0.5$	$\alpha = 0.6$	$\alpha = 0.7$	$\alpha = 0.8$	$\alpha = 0.9$
Benchmark	108213	128144	146666	160192	166379
Alt	**108330**	128854	**147103**	160218	**166379**
Div	108613	**128504**	**147103**	**160192**	**166379**
NoDiv	108512	128794	147301	160218	**166379**
Fixed (10)	108594	128974	147254	160218	**166379**
Fixed (20)	108548	128867	147288	160218	**166379**

Table 4. Various tabu approaches using additional iterations for the best result

In conclusion, the alternating approach is superior when the number of tabu iterations is relatively small. When many tabu iterations are available, the diversification approach is better.

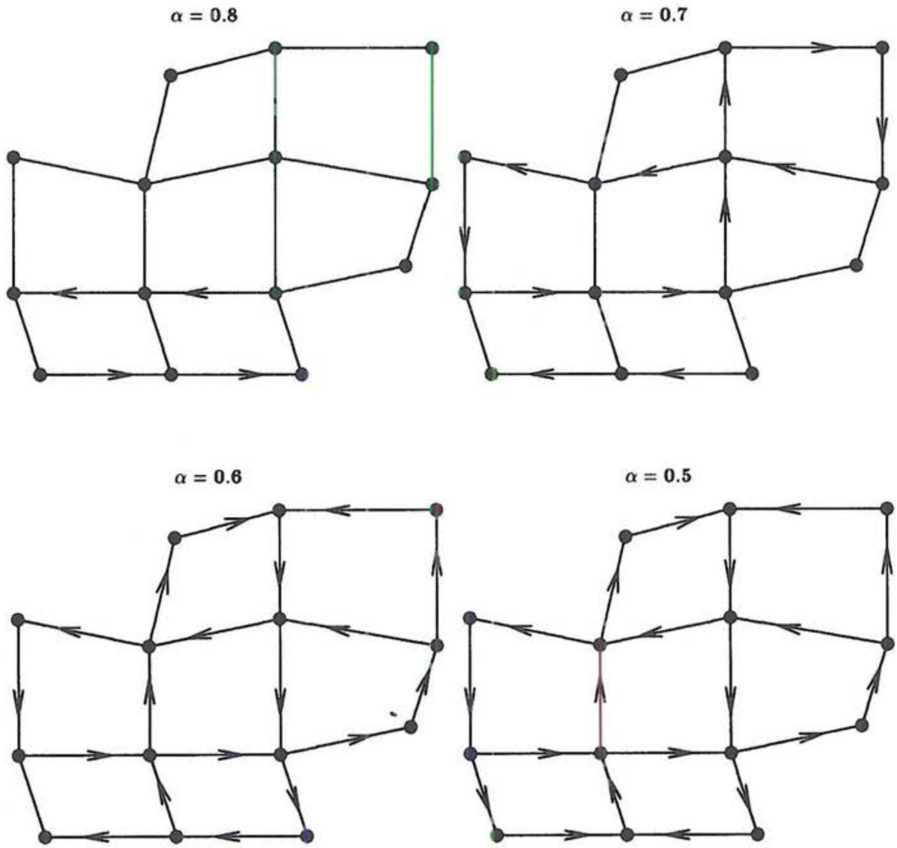
All tabu search approaches are better than the greedy heuristic when the same computer time is required. The results in Table 3 should be compared with the best greedy result reported in Table 1.

For each α there must be a range of α's for which the optimal configuration is the same. We therefore tried to use the best solution for one α as a starting configuration for another α. We usually obtained better than average solutions.

configuration was used for $\alpha = 0.5$, one greedy iteration provided a better solution than the best $\alpha = 0.5$ solution (see Table 2). Designing special algorithms by varying the value of α is left for future research.

Scenario 2: Small problems

We experimented with the five small problems tested in Drezner and Wesolowsky (1997), containing $n = 14$ nodes, and $m = 20$ arcs (see Fig. 2). The size of the neighborhood is therefore $2m = 40$. The 5%–10% rule for the tabu size leads to tabu size between 2 and 4. For such a small problem these tabu sizes are too small. We obtained better results by using tabu sizes of 3 and 6. The results are summarized in Table 5. Each procedure was repeated fifty times. One run took about one second of computer time. In the table we give the optimal solution and two simulated annealing experiments that are reported in Drezner and Wesolowsky (1997). We report new results for the greedy algorithm, and the five



Desc.	$\alpha = 0.5$ †	$\alpha = 0.5$ Avg.	$\alpha = 0.6$ †	$\alpha = 0.6$ Avg.	$\alpha = 0.7$ †	$\alpha = 0.7$ Avg.	$\alpha = 0.8$ †	$\alpha = 0.8$ Avg.	$\alpha = 0.9$ †	$\alpha = 0.9$ Avg.
Optimum	1385.0		1656.6		1859.2		1973.0		1998.0	
Greedy	4	1439.4	1	1708.0	6	1880.1	**50**	**1973.0**	**50**	**1998.0**
Sim. An.(1)	5	1423.4	10	1690.4	7	1871.5	35	1974.1	48	1998.2
Sim. An.(2)	12	1419.3	6	1691.4	5	1872.1	21	1977.0	39	1999.2
Alt	17	1405.5	23	1681.3	43	1864.1	**50**	**1973.0**	**50**	**1998.0**
Div	**38**	**1390.9**	**43**	**1663.2**	**50**	**1859.2**	**50**	**1973.0**	**50**	**1998.0**
NoDiv	19	1405.2	27	1676.1	41	1864.2	**50**	**1973.0**	**50**	**1998.0**
Fixed (3)	12	1422.6	30	1679.1	11	1866.7	**50**	**1973.0**	**50**	**1998.0**
Fixed (6)	28	1396.8	36	1669.0	11	1861.2	**50**	**1973.0**	**50**	**1998.0**

† Number of optimal solutions

Table 5. Results for the small problems

approaches tested for the medium size problem. For these particular problems, all methods obtained the optimal solution at least once. The preferred method for these small problems is clearly the diversification approach. The solutions to the small problems are depicted in Fig. 2. The solution to the $\alpha = 0.9$ problem was all two-way roads. Therefore, we do not present it in Fig. 2.

Scenario 3: Real network

In this experiment we generated a realistic problem based on the highway system in Orange County, California. Sixteen communities in Orange County represent the nodes and twenty three arcs depict the major highways in the area (see Fig. 3). The coordinates and population figures for these communities are given in Drezner and Drezner (1998). The traffic between any two communities is calculated as the geometric mean of the populations of the two communities divided by the shortest distance between the communities, and then rounded down to the nearest integer. All the computer runs for this problem were performed on

Desc.	$\alpha = 0.5$ †	$\alpha = 0.5$ Avg.	$\alpha = 0.6$ †	$\alpha = 0.6$ Avg.	$\alpha = 0.7$ †	$\alpha = 0.7$ Avg.	$\alpha = 0.8$ †	$\alpha = 0.8$ Avg.	$\alpha = 0.9$ †	$\alpha = 0.9$ Avg.
Optimum	16964.0		18320.2		18930.4		19304.6		19500.4	
Time (min)	199.14		156.17		64.47		16.41		1.80	
Greedy	0	17170.0	4	18427.2	30	18936.0	49	**19304.9**	**100**	**19500.4**
Alt	20	16991.8	66	188350.9	**100**	**18930.4**	**100**	**19304.6**	**100**	**19500.4**
Div	**28**	**16990.6**	**100**	**18320.2**	**100**	**18930.4**	**100**	**19304.6**	**100**	**19500.4**
NoDiv	15	16992.8	**100**	**18320.2**	**100**	**18930.4**	**100**	**19304.6**	**100**	**19500.4**
Fixed (4)	9	17002.4	63	18353.7	66	18935.2	**100**	**19304.6**	**100**	**19500.4**
Fixed (8)	15	16992.4	**100**	**18320.2**	**100**	**18930.4**	**100**	**19304.6**	**100**	**19500.4**

† Number of optimal solutions

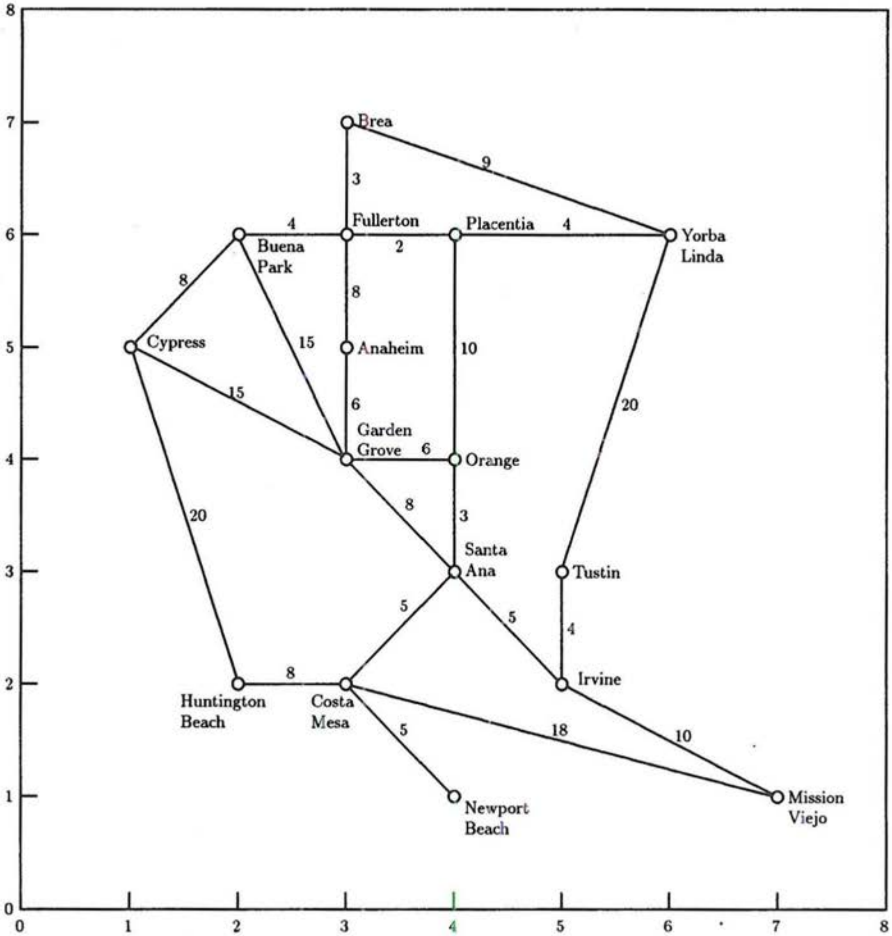


Figure 3. The Orange County Problem.
5-Freeway: Buena Park – Garden Grove – Santa Ana – Irvine – Mission Viejo
22-Freeway: Cypress – Garden Grove – Orange
55-Freeway: Placentia – Orange – Santa Ana – Costa Mesa – Newport Beach
57-Freeway: Brea – Fullerton – Anaheim – Garden Grove
91-Freeway: Cypress – Buena Park – Fullerton – Placentia – Yorba Linda
405-Freeway: Cypress – Huntington Beach – Costa Mesa – Mission Viejo
Eastern Corridor: Yorba Linda – Tustin – Irvine
Imperial Highway: Brea – Yorba Linda

a 233 MHz computer. The ratio of run times between the 166 MHz computer and 233 MHz computer is 1.4. In Table 6 we report results for the optimal solution, and five different strategies of the tabu search. Since the neighborhood size is 46, we first tested the range for tabu size between 2 (5%) and 5 (10%) but found the range of 4 to 8 better. Average run times for the greedy search range from 0.10 to 0.27 seconds. The average run times for the tabu search range from 0.76 to 5.79 seconds. In this example the optimal solutions were also found by branch and bound (Drezner and Wesolowsky, 1997). The diversification approach was the best performer with average results within 0.2% of optimality as reported in Table 6.

6. Conclusions and future research

This study investigates the optimal design of a road system which may include one-way and two-way routes. The objective is to minimize the total travel time of the system users. A tabu search technique is developed in order to find good solutions for this problem. New ways of handling the tabu size and a new diversification strategy are proposed. These implementations are tested on two random networks and one based on real road network from Orange County, California. The results obtained are encouraging. These approaches yield better results than those obtained by commonly used strategies employed in tabu search.

The diversification approach was found to be the best one when a relatively large number of iterations are allowed in the tabu search. When we limit the number of iterations to a smaller number, the alternating approach performs slightly better.

It was found that our tabu search implementation performs better than the simulated annealing algorithm reported in Drezner and Wesolowsky (1997). It is reported there that using similar run times, the greedy heuristic provided better results than simulated annealing. We are investigating the effectiveness of an improved simulated annealing and genetic algorithms for the solution of our problem. The results of this investigation will be presented in future papers.

References

BATTITI, R. (1996) Reactive tabu search: Toward self-tuning heuristics. In: V.J. Rayward-Smith, I.H. Osman, C.R. Reeves and G.D. Smith, eds., *Modern Heuristic Search Techniques*, 61-83. Wiley.

BATTITI, R. and TECCHIOLLI, G. (1994) The reactive tabu search. *ORSA Journal on Computing*, **6**, 126-140.

DAMMEYER, F. and VOSS, S. (1993) Dynamic tabu list management using the reverse elimination technique. *Annals of Operations Research*, **41**, 31-46.

DREZNER, Z. and DREZNER, T. (1998) Applied location theory models. In: G.A. Marcoulides, ed., *Modern Methods for Business Research*, LEA, DREZNER, Z. and WESOLOWSKY, G.O. (1997) Selecting an optimum configuration of one-way and two-way routes. *Transportation Science*, to appear.

GLOVER, F. (1986) Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, **13**, 533-549.

GLOVER, F. (1989) Tabu search—part I. *ORSA Journal on Computing* **1**, 190-206.

GLOVER, F. (1995) Tabu thresholding: Improved search for nonmonotonic trajectories. *ORSA Journal on Computing*, **7**, 426-442.

GLOVER, F. and LAGUNA, M. (1997) *Tabu Search.* Kluwer Academic Publishers, Boston.

GLOVER, F., LAGUNA, M., TAILLARD, E. and DE WERRA, D., eds. (1993) Tabu Search. *Annals of Operations Research*, **41**.

KELLY, J., LAGUNA, M. and GLOVER, F. (1994) A study of diversification strategies for the quadratic assignment problem. *Computers & Operations Research*, **21**, 665-893.

MAGNANTI, T.L. and WONG, R.T. (1984) Network design and transportation planning: Models and algorithms. *Transportation Science*, **18**, 1-55.

MARCOTTE, P. (1983) Network optimization with continuous control parameters. *Transportation Science*, **17**, 181-197.

OSMAN, I.H. (1993) Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research*, **41**, 421-451.

OSMAN, I.H. and LAPORTE, G. (1996) Metaheuristics: A bibliography. *Annals of Operations Research*, **63**, 513-623.

OSMAN, I.H. and SALHI, S. (1996) Local search strategies for the vehicle fleet mix problem. In: V.J. Rayward-Smith, I.H. Osman, C.R. Reeves and G.D. Smith, eds., *Modern Heuristic Search Techniques*, 131-154. Wiley.

REEVES, C.R. (1993) Modern heuristic techniques for combinatorial problems. *Blackwell Scientific Publications.*

SALHI, S. (1996) An insight into constructive and tabu search heuristic design: Application to a class of location problems. *IFORS XXIX*, Vancouver.

SKORIN-KAPOV, J. (1990) Tabu search applied to the quadratic assignment problem. *ORSA Journal on Computing*, **2**, 33-45.

THOMAS, P. and SALHI, S. (1998) A tabu search heuristic for the resource constrained project scheduling problem. *Journal of Heuristics*, **4**, 123-139.