OPERATIONS RESEARCH, Vol. 25, No. 3, May-June 1977

Technical Notes

An Effective Heuristic for the *M*-Tour Traveling Salesman Problem with Some Side Conditions

ROBERT A. RUSSELL

University of Tulsa, Tulsa, Oklahoma

(Received original September 1975; final, July 1976)

This note presents a heuristic for determining very good solutions for the symmetric *M*-tour traveling salesman problem with some side conditions. These side conditions pertain to load, distance and time, or sequencing restrictions. The heuristic is an extension of the highly successful one of Lin and Kernighan for the single traveling salesman problem. Computational experience with widely tested vehicle dispatch problems indicates that the proposed heuristic consistently yields better solutions than existing heuristics that have appeared in the literature. Run times grow approximately as $N^{2.3}$, where N is the number of cities. The heuristic is generally slower than the modified SWEEP heuristic except on problems having a large number of points per route.

THIS NOTE presents a heuristic algorithm that generates very good solutions to the symmetric M-tour traveling salesman problem with some side conditions. The specific problem is: Given an n by n symmetric matrix of distances between n cities, m salesmen, and a "load" associated with each city, find M tours of minimum total length that leave a depot, visit each city only once, return to the depot, and satisfy certain side conditions. These side conditions pertain to an upper bound on the total load or distance associated with each tour. We also consider time or sequencing restrictions in which individual cities may have due dates or interval constraints requiring that they be visited only during certain time intervals.

The above node routing problem is a generalization of the well-known vehicle dispatch problem [9]. Exact optimization techniques exist [4, 17] for vehicle dispatch problems but are severely limited by the size of the problem that can be solved. Good solutions to this class of combinatorial problems have been achieved only by heuristic approaches. Since Dantzig and Ramser [6] first formulated the vehicle dispatch problem in 1959, many heuristics have been developed [4, 5, 7–9, 12, 13, 20–22]. These heuristics solve vehicle routing problems with various degrees of speed and accuracy. However, none considers time interval or due date constraints.

The heuristic presented in this paper (henceforth referred to as MTOUR) yields better-quality solutions than the above methods and also accommodates sequencing and due date restrictions. The MTOUR heuristic is an extension of the highly successful Lin and Kernighan heuristic [15] for solving single-tour traveling salesman problems. It is directly analogous to Christofides and Eilon's method [4] in which they extend Lin's 3-OPT traveling salesman heuristic [14] to solve the vehicle dispatch problem. Thus MTOUR is not a radically new approach but rather an improved version that is more robust and flexible in terms of problem-solving capability.

1. THE ALGORITHM

MTOUR solves constrained multiple-tour problems as a single traveling salesman problem on an expanded graph. This trick of solving a multiple-tour problem as a single-tour problem is well documented in the literature [1, 4, 11, 16, 19]. The essential modification that MTOUR imparts to the Lin and Kernighan procedure is the explicit enforcement of the various side conditions. The procedure begins at a feasible starting point and makes distance (time) improving changes in the tours only if they are feasible.

The Lin and Kernighan and MTOUR algorithms are based on a general approach to heuristics that has applicability to graph partitioning problems as well as traveling salesman problems. Let S denote the set of all links in a traveling salesman problem. Then the constrained multiple traveling salesman problem may be stated as "find from set S a subset T that forms M distinct tours that satisfy all load, distance, and sequencing constraints and also minimize total tour length." If T denotes any current feasible M-tour, then the MTOUR algorithm proceeds by identifying links y_i in S-T to replace links x_i in T. This exchange process continues until no further improvement in tour length can be found. If $|x_i|$ denotes the length of a link, then k y_i links are selected to replace k x_i links only if the gain criterion for the set of links, $G_k = \sum_{i=1}^{k} (|x_i| - |y_i|) > 0$, is satisfied.

During an "iteration" the link exchange process continues in stages or levels. A level refers to the number of y_i links that are exchanged for x_i links. The algorithm builds lists of promising exchanges at various levels. Thus a list of 2-level exchanges is stored from which 3-level and 4-level exchanges are explored. Finally, an open-ended k-level exchange process is considered. Hence during one iteration many possible exchanges are examined. Some of these exchanges result in a feasible tour at a given level while others do not. A final measure of gain G^* is stored for every exchange that results in a connected valid tour. At the end of an iteration the exchange associated with the largest gain G^* is implemented and the process repeats. Refer to Lin and Kernighan's work [15] for a complete description of the unconstrained method.

In the traveling salesman problem with no side conditions, the Lin and Kernighan heuristic rapidly evaluates link exchanges through the simple gain criterion. In this case desirable exchanges are also feasible. In constrained multiple-tour problems, however, the feasibility as well as the desirability of an exchange must be checked. This can slow the process considerably, for when side conditions are tight, many infeasible exchanges might be enumerated before a feasible one is found. Thus side conditions are examined only in those exchanges for which a final gain G^* is measured. In some proposed link exchanges, the resulting tours can be rapidly evaluated for feasibility by using closed-form expressions for the associated loads, distances, or times. More complicated link exchanges require an enumerative procedure for checking side conditions.

The MTOUR heuristic is starting point-dependent and requires a feasible tour to begin computation. In the course of experimentation, several types of starting points were considered. For vehicle dispatch problems having only load and distance constraints, the best starting point is obtained by using a simple heuristic such as the one given by Clarke and Wright [5] or a streamlined version of the SWEEP heuristic [8]. To avoid local optima, the starting solution is perturbed by randomly permuting the sequence of points within each route. No algorithm for obtaining starting solutions for multiple traveling salesman problems having time interval restrictions has been developed. The starting points for problems 14 to 16 in Table III were developed manually.

2. COMPUTATIONAL EXPERIENCE

A computer program was developed to implement the generalized version of the Lin and Kernighan heuristic. The code was written in FORTRAN IV and executed on an IBM 370/168 computer using the level H compiler. For a problem having N points (excluding the depot or home city) and M tours, the code requires approximately $2\ (N + M)^2 + 52\ (N + M) + 9000$ words of core storage.

In order to test the computational performance of MTOUR in relation to other heuristics, we chose a widely tested set of vehicle dispatch problems. Table I identifies 16 of the problems chosen, their size, and origin. Although these first 13 problems consider only load and distance side conditions, they nevertheless demonstrate the robustness of MTOUR for this class of problems. Table II reports the computational results of

MTOUR and six other heuristics for solving vehicle dispatch problems. The other heuristics compared in Table II are Clarke and Wright's savings approach [5], Christofides and Eilon's 3-OPT method [4], Krolak, Felts and Nelson's man-machine approach [13], and Gillette and Johnson's modified SWEEP algorithm [8]. Also compared in Table II are two well-known indirect or decomposition methods that are practical approaches for incorporating side conditions into an M-vehicle routing problem. The cluster-first heuristic [2] decomposes an M-vehicle problem into M single-vehicle problems by first clustering the nodes into M sub-

TABLE I

LIST OF PROBLEMS

Problem no.	Source	No. cities	Max load	Max distance	Time constraints
1	[7]	21	6000	210	No
2	[7]	22	4500	240	No
3	[7]	29	4500	240	No
4	[7]	32	8000	240	No
5	[5]	30	140	Unlimited	No
6	[4]	50	160	"	No
7	[9]	75	100	"	No
8	[4]	75	140	"	No
9	[9]	75	180	"	No
10	[9]	75	220	"	No
11	[9]	100	112	"	No
12	[4]	100	200	"	No
13	New	159	290	"	No
14	[3]	12	Unlimited	"	Yes
15	New	47	Unlimited	"	Yes
16	New	163	275	"	Yes

sets, each satisfying the side conditions. The Clarke and Wright heuristic is used to cluster first with respect to distance and load capacity, and the Lin and Kernighan traveling salesman heuristic is used for routing within clusters. The cluster-second heuristic [2] decomposes a single unconstrained tour into M subsets satisfying the side conditions. In this approach the Lin and Kernighan procedure is used first to form a giant single tour that is subsequently decomposed. Beltrami and Bodin [2] report that the cluster-first heuristic works better when side conditions are very tight and the cluster-second heuristic works better when side conditions are loose and there are many nodes in each route. The computational information reported in Table II from left to right is solution value or total tour distance, number of tours generated, and computation time. In addition, the time per solution generated and time for generating a starting feasible solution

TABLE II

Comparison of Algorithms on Vehicle Dispatch Problems

Problem no.	No cities	Clarke and Wright			3-OPT			Man-machine		Modified SWEEP			Cluster second			Cluster first			MTOUR			Min/Sol.	Start time
		Sol.	Tours	Min[a]	Sol	Tours	Min[a]	Sol.	Tours	Sol.	Tours	Min[b]	Sol	Tours	Min[b]	Sol.	Tours	Min[b]	Sol.	Tours	Min[b]		
1	21	598	4	0.1	585	4	0.6			589	4	0.02	631	4	0.02	598	4	0.02	585	4	0.06	0.02	0.003
2	22	955	5	0.1	949	5	0.5			956	5	0.02	1104	6	0.03	956	5	0.02	956	5	0.14	0.046	0.019
3	29	963	5	0 2	875	4	0 8			877	4	0.04	1036	5	0.26	962	5	0 04	875	4	0 23	0.076	0.025
4	32	839	5	0.2	810	4	0.8			813	4	0.05	925	5	0.22	835	5	0.04	810	4	0.27	0.09	0.058
5	30	1427	8	0.2	1414	8	0 8	1413	8				1457	9	0.21	1263	7	0.04	1214	7	0.31	0.078	—[c]
6	50	585	6	0.6	556	5	2.0	546	5	524	5	0.06	636	6	0.08	579	6	0.05	524	5	0.25	0.083	0.014
7	75									1096	15	0.08	1142	15	0.18	1054	15	0.11	1061	15	2.04	0.68	0 019
8	75	900	10	1.3	876	10	4.0	859	10	865	10	0.09	958	11	0.18	901	11	0.11	854	10	4.08	0.58	0.018
9	75									752	8	0.15	870	8	0.18	786	8	0.11	745	8	2.55	0.425	0.012
10	75									704	7	0.17	786	7	0.18	737	7	0.11	697	7	4.54	0.412	0.020
11	100									1146	14	0.13	1263	15	0.48	1136	14	0.18	1114	14	5.04	1.26	0.013
12	100	887	8	2.5	863		10.0	843		851	8	0.30	969	8	0.48	864	8	0.20	833	8	1.66	0.55	0.030
13	159									289	4	18.60	329	5	3.74	286	4	1.43	278	4	8.48	2.12	0.574

[a] IBM 7090.
[b] IBM 370/168.
[c] Manually derived.

are reported for M-tour. For problems that have no sequencing constraints, MTOUR generates a solution, permutes it, and restarts until two successive solutions yield no improvement.

From Table II it is clear that MTOUR dominates the other algorithms in terms of solution quality or accuracy. MTOUR yields the best solution found in 11 of the 13 vehicle dispatch problems. It is evident that the superiority of the Lin and Kernighan heuristic over the original 3-OPT heuristic extends to constrained multiple-tour problems.

It is difficult to compare the execution times of various methods with different programmers, compilers, and machines. However, in Table II the modified SWEEP and MTOUR heuristics were executed on the same IBM 370/168 computer using the same compiler. The execution times for the modified SWEEP heuristic in Table II are approximately 50% higher than those reported by Gillette and Johnson [8] on another 370/168 computer. The modified SWEEP heuristic is approximately an order of magnitude faster on problems with relatively few points per route. However, on vehicle dispatch problems with a large (more than 20) number of points per route, MTOUR is more efficient. For instance, on problem 13, which has an average of 40 points per route, MTOUR is roughly 2.2 times as fast as the modified SWEEP. Modified SWEEP, however, can be made slightly more efficient if the improvements implemented by Russell [18] are incorporated in the Gillette and Johnson algorithm. A curve fit analysis based on the limited data in Table II indicates that the execution times for MTOUR grow approximately as $N^{2.3}$, where N is the number of cities. Also, run times grow slightly with an increase in the number of routes for a given number of cities or points. Note that this phenomenon is opposite in SWEEP-type algorithms. The man-machine interaction approach yields "good" subjective solutions but at the expense of considerable human effort and special computer display equipment.

The two cluster heuristics yielded strikingly different results. The cluster-second procedure did not perform well, in spite of relatively loose side conditions for some of the problems. On the other hand, the cluster-first heuristic produced solutions values that averaged only 4.2% higher than MTOUR, the maximum increase being 10.4%. The efficiency of the cluster-first heuristic suggests that it is a viable tool for solving really large-scale problems. Recently Golden, Magnanti, and Nguyen [10] have modified the Clarke and Wright procedure to solve a 600-node problem in 20 sec on IBM 370/168 computer. This procedure, however, does not handle time constraints.

The results reported in Table III pertain to time-constrained test problems. Little literature has appeared and consequently few test problems exist for this class of problems. Problem 14 is from Biles and Bradford [3] and considers only due dates. Problems 15 and 16 are real-world problems in which minimum and maximum allowable visitation times are specified. These time limits pertain to the earliest and latest times during which service may occur. Problem 15 is data derived from the actual intra-city delivery schedule of two beer trucks. Problem 16 is data derived from the refuse collection schedule of four industrial hoist compacter trucks in a large southwestern city. The details of problems 13, 15, and 16 can be obtained from the author.

As could be expected, the execution times of time-constrained problems are comparable to the execution times for the closely related vehicle dispatch problems of equivalent size. The solutions derived in problems 15 and 16 constitute a 25% to 30% improvement in total distance as compared with the original ad hoc solutions.

TABLE III

COMPUTATIONAL RESULTS OF MTOUR ON TIME-CONSTRAINED PROBLEMS

Problem no.	No. of points	Total distance	Travel time	Total time	No. of tours	Computation time (min)[a]
14	12	—	228 days	446 days	2	0.01
15	47	18.87	56.63 min	1331.63 min	2	0.09
16	163	311.35	539.75 min	1569.75 min	4	1.27

[a] IBM 370/168.

The cluster-second heuristic is not used on the time-constrained problems because of the tightness of the side conditions. The cluster-first heuristic is also not attempted because the Clarke and Wright savings approach commences with all points directly connected to the depot; this is highly infeasible for time-constrained problems. To develop a clustering heuristic for distance, load capacity, and time constraints is an area for further research.

3. CONCLUSION

The Lin and Kernighan heuristic successfully generalizes to single-depot symmetric M-tour traveling salesman problems having a small number of side conditions such as load, distance, or sequencing constraints. The heuristic performs best when side conditions are not extremely tight since tight constraints require rejection of a higher percentage of infeasible link exchanges. Although less efficient, MTOUR is more accurate than other heuristics in solving vehicle-dispatched-type problems. Additionally, it can handle time or sequencing constraints.

REFERENCES

1. M. BELLMORE AND S. HONG, "Transformation of Multisalesmen Problem to the Standard Traveling Salesman Problem," *J. ACM* **21**, 500–504 (1974).
2. E. J. BELTRAMI AND L. D. BODIN, "Networks and Vehicle Routing for Municipal Waste Collection," *Networks* **4**, 65–94 (1974).

3. W. Biles and J. Bradford, "A Heuristic Approach to Vehicle Scheduling with Due-Date Constraints," paper presented at the Spring National ORSA Meeting, Chicago, Illinois, 1975.
4. N. Christofides and S. Eilon, "An Algorithm for the Vehicle-Dispatching Problem," *Opnl. Res. Quart.* **20**, 309–318 (1969).
5. G. Clarke and J. Wright, "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points," *Opns. Res.* **12**, 568–581 (1964).
6. G. Dantzig and J. Ramser, "The Truck Dispatching Problem," *Management Sci.* **6**, 81–91 (1969).
7. T. Gaskell, "Bases for Vehicle Fleet Scheduling," *Opnl. Res. Quart.* **18**, 281–295 (1957).
8. B. Gillett and T. Johnson, "Sweep Algorithm for the Multiple Depot Vehicle Dispatch Problem," paper presented at the Fall ORSA/TIMS Meeting, San Juan, Puerto Rico, 1974.
9. B. E. Gillett and L. R. Miller, "A Heuristic Algorithm for the Vehicle Dispatch Problem," *Opns. Res.* **22**, 340–349 (1974).
10. B. Golden, T. Magnanti, and H. Nguyen, "Implementing Vehicle Routing Algorithms," Technical Report No. 115, Operations Research Center, Massachusetts Institute of Technology, 1975. (to appear in *Networks*).
11. S. Gorenstein, "Printing Press Scheduling for Multi-Edition Periodicals," *Management Sci.* **16**, B373–383 (1970).
12. R. Hayes, "The Delivery Problem," Carnegie-Mellon University, Graduate School of Industrial Administration, Pittsburgh, Report No. MSR 106, 1967.
13. P. Krolak, W. Felts, and J. Nelson, "A Man-Machine Approach toward Solving the Generalized Truck-Dispatching Problem," *Transportation Sci.* **6**, 149–169 (1972).
14. S. Lin, "Computer Solutions of the Traveling Salesman Problem," *Bell System Tech. J.* **44**, 2245–2269 (1965).
15. S. Lin and B. Kernighan, "An Effective Heuristic Algorithm for the Traveling Salesman Problem," *Opns. Res.* **21**, 498–516 (1973).
16. C. S. Orloff, "Routing a Fleet of M Vehicles to/from a Central Facility," *Networks* **4**, 147–162 (1974).
17. J. Pierce, "Direct Search Algorithms for Truck Dispatching Problems, Part I," *Transportation Res.* **3**, 1–42 (1969).
18. R. Russell, "Efficient Truck Routing for Industrial Refuse Collection," paper presented at the Fall ORSA/TIMS Meeting, San Juan, Puerto Rico, 1974.
19. J. Svestka and V. E. Huckfeldt, "Computational Experience with an M-Salesman Traveling Salesman Algorithm," *Management Sci.* **19**, 790–799 (1973).
20. F. Tillman and H. Cochran, "A Heuristic Approach for Solving the Delivery Problem," *J. Ind. Eng.* **19**, 354–358 (1968).
21. M. Tyagi, "A Practical Method for the Truck Dispatching Problem, *J. Opns. Res. Soc. Japan* **10**, 76–92 (1968).
22. P. Yellow, "A Computational Modification to the Savings Method of Vehicle Scheduling," *Opnl. Res. Quart.* **21**, 281–283 (1970).