ZASTOSOWANIA MATEMATYKI
APPLICATIONES MATHEMATICAE
XVIII, 1 (1983), p. 91-95

A. ADRABIŃSKI and M. M. SYSŁO (Wrocław)

COMPUTATIONAL EXPERIMENTS WITH SOME APPROXIMATION ALGORITHMS FOR THE TRAVELLING SALESMAN PROBLEM

1. Introduction. In this paper we present some results of computational experiments with several heuristic algorithms for solving the travelling salesman problem. The computations were carried out for the Lin-Kernighan algorithm with the starting solutions obtained by different very fast approximation algorithms. For a great number of the literature examples (up to $n = 57$) the optimum solutions have been obtained. It was found that the farthest insertion method is superior to the other fast approximation methods and produces also comparatively best starting solutions for the Lin-Kernighan algorithm.

2. Preliminaries. The *travelling salesman problem* (TSP) can be formulated in several ways and ere we make use of the following one.

We are given a complete graph $G = (V, E)$ on n vertices and an $n \times n$ distance matrix $D = (d_{ij})$ that defines the distance function $d\colon V \times V \to R_+ \cup \{0\}$. We call $d(i,j)$ the *length of the arc* (i,j) and assume $d(i,i) = 0$ for all i. For a subset $S \subset E$, the length of S is defined as follows:

$$d(S) = \sum_{(i,j)\in S} d(i,j).$$

A *travelling salesman tour* or, simply, a *tour* is defined to be a closed simple path that passes through every vertex of G exactly once. The travelling salesman problem is to find a minimum length tour in G.

There are several special cases of the TSP which depend on some properties of the distance function. For example, if $d(i,j) = d(j,i)$ for all i and j, then we have a *symmetric TSP*. In this case, an arc (i,j) is called an *edge* and denoted by $\{i,j\}$. If $d(i,j) \leqslant d(i,k)+d(k,j)$ for all i, j, and k, then the function d satisfies the *triangle inequality*. If d defines the distance norm between the vertices of G, then we have a *Euclidean* or *geometric* case of the TSP.

It is well known that, except for some very special cases, the TSP is NP-complete, and hence unlikely to be solvable in polynomial time.

This motivates the interest in the study of polynomial approximation algorithms.

We present some computational results obtained for the ALGOL-60 implementations of several algorithms which find the approximate solution to the symmetric TSP. The computations were carried out for the Lin-Kernighan algorithm with the starting solutions obtained by different very fast approximation algorithms. For details of the algorithms and their ALGOL-60 implementations the reader is referred to [2], [4], [6], [8].

We have investigated the following algorithms:

A. the tree alteration algorithm, procedure *FRTSP* (in [2]),

B. the nearest neighbour algorithm, procedure *NNTSP*,

C. the nearest insertion algorithm, procedure *NITSP*,

D. the farthest insertion algorithm, procedure *FITSP*,

E. the nearest addition algorithm, procedure *NATSP*,

F. the Lin-Kernighan algorithm, procedure *LKTSP*.

The procedures A-E have running times bounded by $O(n^2)$, and procedure *LKTSP* by $O(n^3)$.

The following theorem contains the theoretical characterization of how the solutions obtained by the above methods compare with the optimum ones. Let T_{app} and T_{opt} denote an approximate tour and the optimal tour, respectively.

THEOREM ([4], [8]). *If a symmetric distance function d satisfies the triangle inequality, then*

$$d(T_{\text{app}}) \leqslant 2d(T_{\text{opt}})$$

for T_{app} *obtained by algorithms* A, C, D, E, *and*

$$d(T_{\text{app}}) \leqslant (\tfrac{1}{2}\lceil \log_2 n \rceil + \tfrac{1}{2})\, d(T_{\text{opt}})$$

for T_{app} *obtained by algorithm* B.

If the distance function is unconstrained by the triangle inequality, then for any constant $k \geqslant 1$ the problem of finding an approximate tour T such that $d(T) \leqslant kd(T_{\text{opt}})$ is NP-complete (see [9]).

The approximate solution obtained by the methods A-E can be used as initial solutions for the method F which in general produces also only an approximate solution.

3. Computational experiments. The algorithms mentioned in Section 2 have been tested on the Odra-1305 computer for several examples taken from the literature. The results of the computations are contained in Tables 1 and 2. For each example, each of the algorithms A-E has been run with the starting point varying from 1 to n. The columns of Table 1, which correspond to the algorithms, contain the best and the average lengths of the solutions obtained. Then the best solutions have been used as initial ones in procedure *LKTSP*.

n (author)	*FITSP* best solution	*FITSP* average solution	*FRTSP* best solution	*FRTSP* average solution	*NATSP* best solution	*NATSP* average solution	*NITSP* best solution	*NITSP* average solution	*NNTSP* best solution	*NNTSP* average solution	*LKTSP* initial solution values	*LKTSP* initial solution procedures	*LKTSP* solutions	Optimum solutions
10 (Christofides)	212	214	212	241	216	241	216	226	212	230	212	*FITSP*	212	212
												FRTSP		
												NNTSP		
10 (Christofides)	292	293	292	297	326	347	292	301	299	331	292	*FITSP*	292	292
												FRTSP		
												NITSP		
10 (Barachet)	378	385	378	402	389	404	378	391	381	421	378	*FITSP*	378	378
												FRTSP		
												NITSP		
20 (Croes)	316	369	317	371	459	491	340	349	281	379	281	*NNTSP*	246	246
25 (Held, Karp)	1711	1724	1875	2076	1962	2084	1917	1977	1772	1903	1711	*FITSP*	1711	1711
27 ([1])	3719	3819	3965	4292	4450	4726	3831	4296	4074	4421	3719	*FITSP*	3719	3719
27 ([1])	3348	3489	3573	3798	3799	4112	3566	3968	3471	3903	3348	*FITSP*	3336	3336
33 (Karg)	10929	11154	11856	12785	13056	13498	12483	12755	11711	12599	10929	*FITSP*	10861	10861
42 (Dantzig)	709	742	763	818	776	861	775	811	864	937	709	*FITSP*	704	699
											712	*FITSP*	699	
48 (Held, Karp)	11474	12021	13140	14124	13280	14159	12551	13380	12137	13577	11474	*FITSP*	11474	11461
											11505	*FITSP*	11470	
											11577	*FITSP*	11461	
57 (Karg)	13121	13750	14271	14910	14867	15387	14153	14764	14411	15912	13121	*FITSP*	12985	12955
											13128	*FITSP*	13097	
											13193	*FITSP*	13133	
											13207	*FITSP*	12955	
120 ([3])	7155	7458	7951	8383	8454	8774	8141	8364	8245	8815	7155	*FITSP*	7026	6942
											7174	*FITSP*	7007	
											7225	*FITSP*	7059	
											7244	*FITSP*	7005	

The best results have been obtained by *LKTSP* with the initial solutions produced by *FITSP*. In this case, for the best solution obtained by *FITSP* (which was also the optimum solution in 5 cases) procedure *LKTSP* has produced the optimum solutions for $n \leqslant 33$. For other examples with $n \leqslant 57$, *LKTSP* with the best *FITSP* solutions has produced solutions 1 % worse than the optimum ones, and 1.2 % worse for $n = 120$. Using other *FITSP* solutions (i.e., next in the non-decreasing order of the *FITSP* solutions obtained for other starting points), the optimum solutions have been obtained for all examples with $n \leqslant 57$. For $n = 42$, the optimum solution has been obtained for the second best *FITSP* solution, for $n = 48$ — for the third best, and for $n = 57$ — for the fourth best. The best solution for $n = 120$, obtained with the *FITSP* starting solutions, was 0.9 % worse than the optimum one and has been obtained for the fourth best *FITSP* solution.

TABLE 2

n	*FITSP*		*FRTSP*		*NATSP*		*NITSP*		*NNTSP*	
	best solution	average solution	best solution	average solution	best solution	average solution	best solution	average solution	best solution	average solution
10	0	0.94	0	13.68	1.89	13.68	1.89	6.60	0	8.49
10	0	0.34	0	1.71	11.64	18.83	0	3.08	2.40	13.36
10	0	1.85	0	6.35	2.91	6.88	0	3.44	0.79	11.38
20	28.45	5C	28.86	50.81	86.58	99.59	38.21	41.87	14.23	54.07
25	0	0.76	9.59	21.33	14.70	21.80	12.04	15.55	3.51	11.22
27	0	2.69	6.61	15.41	19.66	27.08	3.01	15.51	9.55	18.88
27	0.35	4.59	7.10	13.85	13.88	18.87	6.89	18.94	4.05	17.00
33	0.62	2.70	9.16	17.71	20.21	24.28	14.93	17.43	7.83	16.00
42	1.43	6.15	9.16	17.02	11.02	23.17	10.87	16.02	23.60	34.04
48	0.11	4.89	14.65	23.24	15.87	23.54	9.51	16.74	5.90	18.46
57	1.28	6.14	10.16	15.09	14.76	18.77	9.25	13.96	11.24	22.82
120	3.06	7.43	14.53	20.76	21.78	26.39	17.27	20.48	18.77	26.98

Table 2 shows that the best and the average solutions obtained by algorithms A-E are much better than indicated in the Theorem (see Section 2). Namely, except for one example ($n = 20$), the best and the average solutions are 25 % and 35 % worse, respectively, than the optimum ones.

It is worth noting that only two examples, namely for $n = 10$ and $n = 25$, satisfy the triangle inequality.

The procedures have been also tested on the examples described by Papadimitriou and Steiglitz in [7] and the results obtained showed that they are really very hard for approximation methods.

The results of our computations show that

(1) the farthest insertion method is superior to the other fast approximation algorithms (it was also verified for some random instances of the TSP, see [8]);

(2) the farthest insertion algorithm produces also comparatively best starting solutions for the Lin-Kernighan algorithm.

Added in proof. The conclusions reached in this paper have been confirmed by the results of other computational experiments published in [12]. Approximation algorithms for the asymmetric TSP and k-person TSP appeared in [11] and [10], respectively.

References

[1] A. Adrabiński, *Algorithm 45: A heuristic algorithm for the traveling-salesman problem*, Zastos. Mat. 15 (1976), p. 223-244.

[2] — and M. M. Sysło, *Computational experiments with some heuristic algorithms for the travelling salesman problem*, Report Nr N-78, Institute of Comp. Science, University of Wrocław, Wrocław 1980.

[3] M. Grötschel, *On the symmetric travelling salesman problem: solution of a 120-city problem*, Mathematical Programming Study 12 (1980), p. 61-77.

[4] A. H. Frieze, *Worst-case analysis of algorithms for travelling salesman problem*, p. 93-112 in: W. Oettli and F. Steffens (eds.), *Methods of Operations Research* 32, Verlag Anton Hain, Mannheim 1979.

[5] J. K. Lenstra and A. H. G. Rinnooy Kan, *Complexity of vehicle routing and scheduling problems*, Networks 11 (1981), p. 221-227.

[6] S. Lin and B. W. Kernighan, *An effective heuristic algorithm for the travelling-salesman problem*, Operations Res. 21 (1973), p. 498-516.

[7] Ch. Papadimitriou and K. Steiglitz, *Some examples of difficult travelling salesman problems*, ibidem 26 (1978), p. 434-443.

[8] D. J. Rosenkrantz, R. E. Stearns and P. M. Lewis II, *An analysis of several heuristics for the travelling salesman problem*, SIAM J. Comput. 6 (1977), p. 563-581.

[9] S. Sahni and T. Gonzales, *P-complete approximation problems*, J. Assoc. Comput. Mach. 23 (1976), p. 555-565.

[10] G. N. Frederickson, M. S. Hecht and C. E. Kim, *Approximation algorithms for some routing problems*, SIAM J. Comput. 7 (1978), p. 178-193.

[11] A. M. Frieze, G. Galbiati and F. Maffioli, *On the worst-case performance of some algorithms for the asymmetric travelling salesman problem*, Networks 12 (1982), p. 23-39.

[12] B. Golden, L. Bodin, T. Doyle and W. Stewart, Jr., *Approximate travelling salesman algorithms*, Operations Res. 28 (1980), p. 694-711.

INSTITUTE OF COMPUTER SCIENCE
UNIVERSITY OF WROCŁAW
51-151 WROCŁAW

Received on 10. 4. 1980