Univ. Beograd. Publ. Elektrotehn. Fak.
Ser. Mat. 1 (1990), 41–47.

TSP–SOLVER — A PROGRAMING PACKAGE FOR THE TRAVELING SALESMAN PROBLEM

Dragoš Cvetković, Mirjana Čangalović, Vladimir Dimitrijević, Laslo Kraus, Milan Milosavljević, Slobodan Simić

We report on the implementation of a programming package, called TSP–SOLVER, for the traveling salesman problem (TSP). Various variants of TSP can be treated by TSP–SOLVER: both symmetric and asymmetric cases, one- or multiple- TSP, one or first k best solutions, bandwidth limited distance matrix and others special cases, algorithms and heuristics. The system is user-friendly and offers the user, among other things, some possibilities to intervene during the solving a problem.

0. INTRODUCTION

A programming package for traveling salesman problem **(TSP)** [**19**] has been implemented at University of Belgrade, Faculty of Electrical Engineering in 1989 and 1990. The package is called **TSP–SOLVER**.

Programming package **TSP–SOLVER** is implemented on a *VAX* computer under operating system *VMS* and using programming language *FORTRAN*. A Form Management System (FMS) has been used to create menus for the communication with the user. There are about 500 subroutines linked into several executable task units (subsystems). Subsystems communicate each to other through files with data (*M*–files) on particular **TSP** instances.

Subsystems of the system **TSP–SOLVER** are GENER, SALES, STATIS and EXTER.

GENER creates, modifies and verifies *M*–files. Creation of *M*–files makes use of random number generators. *M*–files created by other programs (outside **TSP–SOLVER**) are verified before treating by SALES.

SALES contains several algorithms and heuristics for **TSP** implemented. Input data are taken from one or more *M*–files. Output is directed again to *M*–files or/and other output devices.

STATIS performs statistical treatment of data in *M*–files including results of working of SALES.

AMS Subject Classification (1980): 68R10

EXTER is a subsystem to link the system **TSP–SOLVER** to the previously developed package GRAPH [**4**], [**15**], [**16**]. EXTER transforms data from **TSP–SOLVER** into formats used in GRAPH. Results of the working of GRAPH the subsystem EXTER can put in corresponding *M*–files. A user can also via subsystem EXTER inspect the actual work of SALES by interrupting a process, inspecting intermediary results, changing modes of the work, i.e. strategy of further work.

Information flow between subsystems is given in *Fig. 1.* All presented information exchanges are realized via *M*–files except for the linkage between EXTER and GRAPH where the medium is a special file used by GRAPH.

Fig. 1.

System **TSP–SOLVER** is in spirit similar to the system TRAVEL [**1**] but has much more algorithms and heuristics implemented. Experiments with **TSP–SOLVER** will be described in other papers. Here we describe subsystems in some detail.

1. SUBSYSTEM GENER AND M–FILES

A **TSP** is defined by a weighted graph represented as the corresponding distance matrix (DM). DM is a non–negative square matrix (of the dimension n) which can contain infinite elements denoted by B. The (i, j)–entry $DM(i, j)$ is equal to B if and only if there exists no edge going from a node i to a node j of the graph. Main diagonal entries of the DM are all equal to B.

The following types of the distance matrix can be treated in package **TSP–SOLVER**.

1° asymmetric DM corresponding to the complete directed weighted graph;

2° symmetric DM corresponding to the complete undirected weighted graph;

3° band DM where there exists a positive integer w ($w < n - 1$) such that $DM(i, j)$ is finite and non–negative for $|i - j| \leq w$. All other entries of the DM are equal to B. The value w is called the bandwidth;

4° the weighted graph has a chain structure (chain–like DM). Such a graph consists of an ordered sequence of complete directed subgraphs in which each node of a subgraph is connected by an arc to each node of the immediately following subgraph;

5° DM is euclidean, i.e. satisfies the triangular inequalities $DM(i,j)+DM(j,k) \geq DM(i,k)$ for all i, j, k.

For the types 3° and 5° both symmetric and asymmetric cases are treated.

All input data of a travelling salesman problem and its solution are recorded in a special file named *M*–file.

The *M*–file consists of three sections:

Section 1 contains the dimension and the type of the distance matrix (defined by 1° − 5°), and some additional data.

Section 2 contains finite entries of the DM recorded as a sequence column by column.

Section 3 contains all results and some of intermediate results obtained by solving the problem.

Contents of the *Section* 1 and 2 are generated by the subsystem GENER. This generating can be either manual (for weighted graphs of smaller dimensions) or automatic.

In the case of the manual generation the user manually sets the elements of the DM recorded to the *Section* 2.

For the automatic generation the finite elements of the DM are randomly generated by using a uniform random numbers generator. A lower and an upper bound of the generated entries should be presented in advance.

During the same run the GENER can automatically generate a given number of the *M*–files with various distance matrices of the same dimension and type. Such files are suitable for statistic experiments concerning the efficiency of some chosen solution methods.

GENER can verify correctness of *M*–files created by other programs. Also the content of existing *M*–files can be modified by changing finite entries of DM and by permuting, adding or deleting nodes of the weighted graph.

2. SUBSYSTEM SALES AND JOBS

The basic command which can be invoked by the subsystem SALES is captured by a notion of a job. Each job is capable to solve a concrete (some particular) **TSP** problem on data items contained in one or more *M*–files and to store its output on the appropriate place. To define a particular job the user is inquired to select:

1° job name;

2° description of a concrete **TSP** problem to be solved by the job;

3° description of a procedure (algorithm or heuristic) to be used in solving a problem as defined by 2°;

4° data items;

5° modes of execution.

Besides the above, the system is capable to keep some other information of a concrete job such as date of creation, identificator of a person who created a job, job status regarding execution, etc. For example, regarding status we can have information if the job has been already executed or not, or if the job is being executed right now, or if it is suspended at execution etc.

Job name is just an identifier of a job. One can by invoking a job by its name either edit a job or execute a job, and also get any selected information about a job. When editing a job one can either edit a new job (answer questions $1^\circ - 5^\circ$), or modify some existing (change its descriptors, again $1^\circ - 5^\circ$) or copy a job (for example, make one or more copy of some existing job for modifications) or even delete a job (to make free space for other more interesting jobs). To execute a job one needs only to invoke a job by its name. In system **TSP–SOLVER** it is possible either to select jobs for execution by giving a name sequence or by selecting in a loop particular jobs for execution. Before execution all checkings regarding data consistency are done. Jobs with nonconsistent data are, of course, rejected.

System **TSP–SOLVER** is able to solve (either exactly by algorithms, or approximately by heuristics) about 30 kind of various **TSP** problems. These variations stem from the characteristics involving the number of optimal solutions, the properties of distance matrix in chosen *M*–files, and some tour requirements. More precisely, the number of solutions to be found may be one or several; the distance matrix may be symmetric or asymmetric, band or non band, euclidean or noneuclidean, chain–like or non–chain–like, etc.; the tour may consists of a single tour (open or closed) or several tours each passing trough a fixed city or not, open or closed, with prescribed number of cities to be visited or not etc. All this requirements are extremly important when selecting an appropriate procedure (method) to be used. For most **TSP** problems there are several algorithm and/or heuristics available. If **TSP** problem and the procedure for its solving are selected, then the data (*M*–files) are to be chosen. Each job with so far descriptions fixed can operate different *M*–files depending on name pattern of corresponding *M*–files. The name of *M*–files can be given by the wildcards characters in the same way, as in operating system *VMS*, on the target machine. Finally, there is a possibility to change the job modes during execution. For instance, one can chose to have messages on screen during execution or not, to redirect the output (to store the results in *M*–files or not), interrupt the execution at some ”check points” or not, etc. All this modes options are at the disposal to the user only to make the system more friendly and to gain its efficiency.

The following algorithms and heuristics have been implemented.

1° symmetric **TSP**: branch and bound algorithm by Volgenant and Jonker [**23**] and two others based on minimal spanning tree and 1–tree as relaxations; 3–optimal and nearest neighbour heuristics.

2° asymmetric **TSP**: branch and bound algorithm by Carpaneto and Toth [**2**] and two others based on minimal assignation [**3**] and oriented rooted tree [**18**] relaxations; 3–optimal and nearest neighbour heuristics.

3° branch and bound algorithms for a given number of best suboptimal solutions for cases 1° and 2° [**5**], [**6**].

4° multiple **TSP**: symmetric and asymmetric; branch and bound algorithms with minimal forests [**6**] and oriented rooted forests [**6**] as relaxations, a heuristic from [**14**].

5° asymmetric **TSP** with band distance matrix: one ore more salesmen, a polynomial algorithm [**7**].

6° chain–like **TSP**, asymmetric, a dynamic programming algorithm [**9**] and an algorithm using suboptimal solutions for subgraphs [**20**], [**21**].

3. SUBSYSTEM STATIS

Subsystem STATIS is designed to provide statistical analysis of the data contained in the *M*–files. The functions of this subsystem are:

1° *Display M–file(s)*: This is merely an auxiliary function intended to provide individual data about one or more *M*–files. One can get general *M*–file data, distance matrix(es), general job data, and job solutions from one or more selected *M*–files.

2° *Collect statistical data*: This function represents the starting point for the statistical analysis of the solutions of the **TSP**s contained in the *M*–files. A selected set of the *M*–files is scanned and a selected data item is extracted from each of them. At a moment two data items can be selected: the path length of the best solution for the **TSP**s, or the CPU time needed to get the solution.

For a given scan only solutions provided by the same, user selected, algorithm or heuristic are examined.

The set of the obtained data is stored into one of the 10 existing working storage areas.

3° *Display statistical data*: The list of data from the selected working storage area is diplayed by this function. For every data the name of the source *M*–file and the data value is displayed. The identifying data of the list as a whole are the data item name and the applied algorithm or heuristics.

4° *Erase statistical data*: By this function the data from the selected working storage area are erased, freeing up the area for another set of data.

5° *Calculate statistical data*: This function provides the mean value, dispersion and standard deviation of the set of data contained in the selected working storage area, or the corelation factor of two sets of data contained in the two selected working storage areas.

4. SUBSYSTEM EXTER AND THE SYSTEM GRAPH

Subsystem EXTER is an interface between **TSP–SOLVER** and GRAPH.

Some (non–weighted) graphs can be created in EXTER from **TSP** instances treated with **TSP–SOLVER** (e.g. minimal spanning trees, minimal assignation graphs and other subgraphs of "short" edges, etc). These graphs can be further treated by the system GRAPH, part ALGOR for graph theoretic algorithms (e.g. we can find vertex–degrees, diameter, eigenvalues etc.). These results can be sent to the corresponding *M*–files through EXTER and further elaborated by STATIS.

Interesting "short edge" graphs can be also created from intermediary distance matrices during the work of a branch and bound algorithm. Communication between EXTER and SALES is realized via a global sherable common area. Synchronization of the access of different subsystems to this area is realized by an algorithm described in [**22**].

Facilities described in this section enable, among other things, the study of complexity indices for **TSP** [**8**], [**11**], [**12**], [**13**], [**17**] and were, in fact, motivated by such investigations.

REFERENCES

1. S. C. Boyd, W. R. Pulleyblank, G. Cornuejols: *TRAVEL – an interactive travelling salesman problem package for IBM-personal computer.* Operations Res. Letters, **6** (1987), **3**, 141-143.

2. G. Carpaneto, P. Toth: *Some new branching and bounding criteria for the asymmetric travelling salesman problem.* Management Sci., **26** (1980), 736-743.

3. G. Carpaneto, P. Toth: *Solution of the assignement problem.* ACM Trans. Math. Software, **6** (1980), 104-111.

4. D. Cvetković: *Further experiences in computer aided research in graph theory, Graphs, Hypergraphs and Applications.* Proc. Conf. Graph Theory held in Eyba, October 1984, ed. H. Sachs, Teubner, Leipzig, 1985, 27-30.

5. D. Cvetković: *Finding k $(k>1)$ shortest traveling salesman tours in a complete digraph with an asymmetric weight matrix.* (Serbo–Croatian), unpublished report, Faculty of Electrical Engineering, Belgrade, 1985.

6. D. Cvetković: *M traveling salesmen in a complete digraph with an asymmetric weight matrix.* (Serbo–Croatian), unpublished report, Faculty of Electrical Engineering, Belgrade, 1986, 1-30.

7. D. Cvetković: *M traveling salesmen in a complete digraph with an asymmetric band weight matrix.* (Serbo–Croatian), unpublished report, Faculty of Electrical Engineering, Belgrade, 1986.

8. D. Cvetković: *Adaptive solving of the traveling salesman problem.* (Serbo–Croatian), unpublished report, Faculty of Electrical Engineering, Belgrade, 1987, 1-48.

9. D. Cvetković: *Connection between global and local solutions of the traveling salesman problem.* (Serbo–Croatian), unpublished report, Faculty of Electrical Engineering, Belgrade, 1987, 1-11.

10. D. Cvetković: *Finding of shortest rooted forest.* (Serbo–Croatian), unpublished report, Faculty of Electrical Engineering, Belgrade, 1987, 1-20.

11. D. Cvetković, V. Dimitrijević, M. Milosavljević: *A spectral traveling salesman problem complexity index.* (Serbo–Croatian), XI bosansko-hercegovački simpozijum iz informatike, Jahorina '87, 30. 3. – 3. 4. 1987., Zbornik radova, Knjiga **2**, 2761-2765.

12. D. Cvetković, V. Dimitrijević, M. Milosavljević: *Traveling salesman problem complexity indices based on minimal spanning trees.* Graph Theory, Proc. Eighth Yugoslav Sem. Graph Theory, Novi Sad 1987, Univ. Novi Sad, Institute of Mathematics, Novi Sad 1989, 43-51.

13. D. Cvetković, V. Dimitrijević, M. Milosavljević: *A traveling salesman problem comlexity index based on linear factors.* (Serbo–Croatian), SYM-OP-IS '87, Hercegnovi 6. – 9. X 1987., Institut za ekonomiku industrije, Beograd, 1987, 95-99.

14. D. Cvetković, V. Dimitrijević, M. Milosavljević: *A class of algorithms for suboptimal solving of the M traveling salesmen problem.* (Serbo–Croatian), SYM-OP-IS '87, Hercegnovi 6. – 9. X 1987., Institut za ekonomiku industrije, Beograd, 1987, 101-106.

15. D. Cvetković, L. Kraus: *"Graph" an expert system for the classification and extension of the knowledge in the field of graph theory, User's manual.* Elektrotehnički fakultet Beograd, 1983.

16. D. Cvetković, Z Radosavljević, S. Simić: *Using expert programming system in investigations in graph theory.* (Serbo–Croatian), SYM-OP-IS '89, 165-168.

17. V. Dimitrijević: *Adaptive decision in a class of pattern recognition systems.* (Serbo–Croatian), Master Thesis, Faculty of Electrical Engineering, Belgrade, 1988.

18. M. Gondran, M. Minoux: *Graphs and algorithms.* John Wiley and Sons, Chichester – New York – Brisbane – Toronto – Singapore, 1984.

19. E. L. Lawler, J. K. Lenstra, A. H. G. Rinnoy Kan, D. B. Shmoys, (editors): *The traveling salesman problem.* John Wiley and Sons, Chichester – New York – Brisbane – Toronto – Singapore, 1985.

20. M. Milosavljević, M. Obradović: *An algorithm for finding the shortest path on a class of large dimensional graphs.* (Serbo–Croatian), Naučno–tehnički pregled, **35** (1985), **6**.

21. M. Milosavljević, M. Obradović: *An algorithm for finding the k shortest paths on a class of large dimensional graphs.* (Serbo–Croatian), X bosansko-hercegovački simpozijum iz informatike, Jahorina '86.

22. J. L. Peterson, A. Silberschatz: *Operating system concepts.* Addison-Wisley Publishing Company, Reading, Mass., 1983.

23. T. Volgenant, R. Jonker: *Branch and bound algorithm for the symmetric travelling salesman problem based on 1–tree relaxation.* European J. Op. Res., **9** (1982), 83-89.

(Received October 31, 1990)

Faculty of Electrical Engineering,
University of Belgrade,
P. O. Box 816, 11001 Belgrade,
Yugoslavia
(Cvetković, Kraus, Milosavljević, Simić)

Faculty of Organizational Sciences,
University of Belgrade,
Jove Ilića 154, 11000 Belgrade,
Yugoslavia
(Čangalović)

Institute for Applied Mathematics and Electronics,
Kneza Miloša 37, 11000 Belgrade,
Yugoslavia
(Dimitrijević)