Hellenic Operational Research Society

(HELORS)

1963-2013

50th Anniversary

2nd International Symposium
&
24th National Conference on Operational Research

Edited by

Prof. Yannis Siskos
University of Piraeus

Prof. Nikolaos Matsatsinis
Technical University of Crete

Prof. John Psarras
National Technical University of Athens

Athens, February 2014

Different Formulations and Benders Decomposition on TSP

Georgios K.D. Saharidis
Department of Mechanical Engineering, University of Thessaly, Leoforos Athinon, Pedion Areos, 38834 Volos, Greece

George Kolomvos
Department of Industrial Engineering and Optimization, Kathikas Institute of Research and Technology, Cyprus, 4 Hatziarfanou str. Paphos, Cyprus

George Liberopoulos
Department of Mechanical Engineering, University of Thessaly, Leoforos Athinon, Pedion Areos, 38834 Volos, Greece

Abstract

In this work we consider the traveling salesman problem in a connected graph. We apply seven different formulations and we compare the results. We also apply Benders decomposition and we observe its behaviour regarding solution time. We conclude that Benders decomposition is not faster than the classical known formulations and we discuss possible reasons behind.

KEYWORDS

VRP; TSP; classical formulations; Benders decomposition

1. INTRODUCTION

Let $G = (V, A)$ be a graph where V is a set of n vertices and A is a set of arcs or edges. Let C be a cost matrix associated with A. V is the set of vertices such that $V = \{0,1,2, \dots, n\}$ and $i, j \in V$. Note that we call the first vertex $i = 0$. Edges connect vertices such that edge ij connects the vertices i and j. We denote by $x_{ij} \in \{0,1\}$ the binary variable which takes the value of 1 if the edge connecting i and j is included in the Hamiltonian cycle and 0 if not. $\mathbf{x}$ is the vector containing the values x_{ij}. Let c_{ij} be the vector of costs associated to the edge ij. The formulation of the TSP without the subtour elimination constraints (SECs) is equivalent to the assignment problem (AP) and is presented right below.

$$\min c_{ij} x_{ij}$$

subject to:

$$\sum_{\cdot} x_{ij} = 1, \forall j \in V \backslash \{i\} \quad (1)$$

$$\sum_{\cdot} x_{ij} = 1, \forall i \in V \backslash \{j\} \quad (2)$$

$$x_{ij} \in \{0,1\} \quad (3)$$

The SECs can been represented in many various ways as it will be explained in the sequel.

Solution algorithms for the TSP are divided in the literature in exact and heuristics. Heuristics can also be combined with exact solution methods yielding efficient hybrid schemes. Most modern algorithms able to tackle large instances of the TSP employ heuristics in some of the solution phases.

ISBN: 978-618-80361-1-6

For a review of approaches to solve the TSP before 1992, the reader is referred to the comprehensive work of Laport (1992). A more recent review with developments and an updated set of modern areas of applications is included in Bektas (2006) and Saharidis (2014). Both exact and heuristics approaches have been employed to tackle the TSP. In the case of exact methods, the problem is formally modelled as an integer programming problem and related techniques are applied. In the case of heuristics, a formal representation using standard mathematical programming notation is not required. It is very common in mathematical programming to associate a problem with an easier or more popular one. There are thus two perspectives to consider the TSP pertaining to the associated underlying problem.

The first perspective is to view it as an assignment problem where each vertex is assigned a descendent, coupled with a set of constraints ensuring the elimination of subtours. Taking the latter into consideration turns the problem from trivial to intractable. The modelling approaches focus on an elegant and economic formulation of the subtour elimination constraints. The work of Dantzig et al. (1954) constituted the first approach to model these constraints. The authors observe that if there was a subtour on a subset S of vertices, then this subtour would contain exactly |S| arcs and as many vertices. This observation is turned into a constraint where one forces every resulting subset of S to have contain no more than |S-1| arcs. Other such formulations emerged in the following decades inspired by the seminal work of Dantzig et al. (1954). In Miller et al. (1960), the number of constraints reduces significantly with the expense of additional variables. Other formulations called flow-based and time-staged were also mentioned presented later on in this paper.

A second perspective of viewing the TSP is as a special case of a minimum 1-spanning tree. This analogy was nicely explored by Held & Karp (1969). The idea is to carefully create an objective function such that the result of the spanning tree which is a lower bound of the TSP closely approximates the TSP. The formulation of the minimization of 1-spanning trees by default excludes subtours, so there is no reason to enforce any subtour elimination constraints. On the other hand, in a minimum spanning tree there may be nodes with a degree greater to two, that is for instance, a node with two descendants nodes, which is prohibited in the TSP.

2. SOLUTIONS METHODS FOR THE TSP

As mentioned before, one may consider the TSP as a combination of an assignment problem and a set of subtour elimination constraints. There are seven available formulations which are presented in Table 1; their difference relate to the way the SECs are represented.

Table 1 Formulations and acronyms

Acronym	Description	Reference
DFJ	The conventional algorithm	Dantzig et al., 1954
MTZ	The sequential algorithm	Miller et al., 1960
SCF	Single commodity flow	Gavish & Graves, 1978
TCF	Two-commodity flow	Finke et al., 1984
MCF	Multi-commodity flow	Wong, 1980
TS1	Time-staged 1	Fox et al., 1980
TS2	Time-staged 2	Fox et al., 1980
TS3	Time-staged 3	Vadja, 1961

We tested and compared the above formulations and obtained the following results. We performed the experiments on a dual-core 2.2GHz processor with 3GB of usable memory. The code was on C++ (Concert Technology) and the solution was provided by the IBM ILOG CPLEX 12.4 suite. Table 2 presents the outcome of these experiments.

ISBN: 978-618-80361-1-6

Table 2. Comparison of exact formulations (solution time in s)

# nodes	DFJ	MTZ	SCF	TCF	MCF	TS1	TS2	TS3
15	2.20	0.16	0.20	0.39	1.08	3.08	3.14	0.84
17	15.27	0.90	0.50	0.88	0.50	6.02	5.86	5.45
25	-	1.19	0.89	1.48	3.77	396.36	187.36	382.20
31	-	23551.86	2.75	18.19	27.64	-	60338.47	-
43	-	26.46	5.22	5.09	188.48	-	21326.84	3287.49
50	-	33.91	15.28	40.81	573.27	-	-	-
65	-	94.23	130.86	66.42	2582.75	-	-	-
80	-	2154.37	315.22	247.86	9429.70	-	-	-
93	-	357.97	316.48	285.00	-	-	-	-
120	-	-	1345.88	12789.48	-	-	-	-

We observe that the conventional formulation DFJ that was historically the first one proposed quickly shows its limits. We cannot afford solving any problem larger than 17 cities, which is our case sounds too restrictive. Time-staged formulations also seem to quickly attain their limits. In the following we wish to seek the optimal solutions in shorter times, so we decide to test decomposition methods. The Benders decomposition method is the most popular and generic; next we will try to customise this solution in order to try it on our problem.

2.1. Benders Decomposition on TSP

We briefly recall the idea of the Benders algorithm (Benders, 1962) where we decompose the initial problem into the primal slave problem, which is a restriction of the initial problem and provides an upper bound in the case of minimisation; and the following relaxation of the initial problem, which is called the restricted master problem and provides a lower bound. At each iteration, the solution of the master is communicated to the slave and the slave returns feasibility and optimality cuts to the master. In our case, the master problem is the assignment problem and the slave problem is the SECs following each formulation.

We applied Benders decomposition on all the formulations above and compared them to the modelling and solution approach proposed in this paper. Table 3 presents the outcome of the experiments.

Table 3. Benders decomposition results (solution time in s)

Test case	MTZ	Benders on MTZ	SCF	Benders on SCF	TCF	Benders on TCF	TS1	Benders on TS1
15	0.16	0.52	0.2	0.7	0.39	1.24	3.08	11.06
17	0.90	3.16	0.5	1.67	0.88	2.93	6.02	19.95
25	1.19	4.33	0.89	2.72	1.48	5.45	396.36	1504.57
31	23551	70912.37	2.75	10.37	18.19	67.61	-	1853.95
43	26.46	101.26	5.22	16.49	5.09	19.57	-	2451.54
50	33.91	110.95	15.28	52.76	40.81	128.43	-	3432.15
93	94.23	367.19	316.48	1002.4	285	880.54	-	7705.21

Benders decomposition was shown to be slower than the initial formulation it was applied to. Typically, solution times are 2 to 3 times greater. We discuss possible reasons in the following section.

ISBN: 978-618-80361-1-6

2.2. Discussion

Benders does not seem to perform well on any type of the instances considered, the reason being the large number of iterations to convergence which essentially translates to bad quality cuts. In the MTZ formulation, the returned cuts from slave to master are low-density cuts, that is, cuts in which a low number of variables appears compared to the number of variables appearing in the master problem. In the MTZ approach the density of the feasibility cuts is in the order or magnitude $\frac{N}{N(N-1)} = \frac{1}{N-1}$. Consequently, when N is large, the number of variables included in the cuts is significantly low. It is known (for instance see: Saharidis & Ierapetritou, 2010; Saharidis et al. 2010) that in cases of low-density cuts, there is substantial room for improvement in the Benders decomposition.

Another reason of this poor behaviour is the tightness of cuts. At every iteration of the algorithm, the following actions occur:

- The master passes its optimal solution to the slave
- If the slave is infeasible, it returns feasibility cuts to master; else the solution communicated by the master is optimal.

In the case of Benders on theMTZ formulation, the slave problem has the following form:

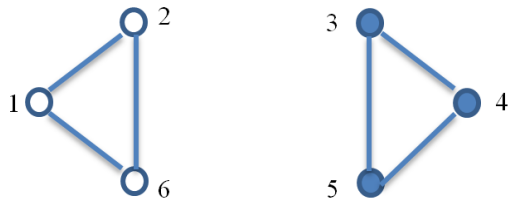
$$u_i - u_j \leq -Nx_{ij} + N - 1\,, \qquad \forall\, i,j \in V\backslash\{0\}, i \neq j \tag{4}$$

The feasibility cuts returned to the master have the following form:

$$\lambda^T(-N)x \leq -\lambda^T(N-1) \tag{5}$$

The dual value λ is non-zero only for those couples u_i and u_j for which x_{ij} was activated, i.e. $x_{ij} = 1$. Let us construct a small example and observe the form this cut takes:

Figure 1 Examples of two subtours



In the example above, the master problem returned the solution:

$$x_{12} = x_{26} = x_{16} = x_{34} = x_{45} = x_{35} = 1$$

When solved, the slave problem assigns a non-zero dual value λ_k to those constraints containing $x_{ij} = 1$ in the solution communicated by the master. The feasibility cut returned by the slave to the master is the following:

$$6x_{34} + 6x_{45} + 6x_{35} \leq 15 \text{ or } x_{34} + x_{45} + x_{35} \leq 2.5$$

Essentially, this constraint instructs the master to exclude the subtour {3-4-5} from the next solution proposed to the slave. In other words, the slave cuts does nothing more than informing the master about the subtour identified. This constraint is a simple SEC that could be manually added at each iteration, instead of having to solve a linear programme to obtain it. This idea need to be exploited further.

ISBN: 978-618-80361-1-6

3. CONCLUSIONS

In this paper we tested the seven well known exact formulations for the TSP and we compared the results among them. The MTZ, SCF and TCF were proved to be the most powerful implementations.

We also tested Benders decomposition on these formulations and observed that the performance was poor. The master problem consisted of the assignment problem while the slave problem varied across formulations. Regardless the formulation, the cuts returned from the slave to the master were of low quality. We focused on the MTZ formulation and suggested reasons of this poor behaviour. One reason was attributed to the density of the cuts that appear to be considerably low. The other reason related to the tightness of cuts that appear to be loose and at the time required to obtain a simple SEC by solving the slave problem.

A future direction of this work will be to propose and implement ways to remedy the above two obstacles. The application of the cut covering bundle and generation and maximum density cuts is being consider to tackle the issue of low-density cuts. Appending progressively a tighter type of SECs at the master simply by inspecting the (master) incumbent solution is being considered to tackle both the issues of tightness and solution time of the whole algorithm.

ACKNOWLEDGEMENT

The authors gratefully acknowledges financial support from the European Commission under the grant FP7-PEOPLE-2011-CIG, GreenRoute, 293753 and the grant EnvRouting SH3_(1234) of Action «Supporting Postdoctoral Researchers» of the Operational Program "Education and Lifelong Learning" (Action's Beneficiary: General Secretariat for Research and Technology, Greece), and is co-financed by the European Social Fund (ESF) and the Greek State.

REFERENCES

Benders, J.F., (1962). Partitioning *Procedures for Solving Mixed-Variables Programming Problems*. Numerische Mathematic. 4: p. 238-252.

Bektas T. (2006). *The multiple traveling salesman problem: an overview of formulations and solution procedures*. Omega, Vol. 34, No. 3, pp. 209-219.

Dantzig, G.B., Fulkerson, D.R. and Johnson, S.M. (1954). Solution *of a large scale traveling-salesman problem*. Operations Research, Vol. 2, pp. 393-410.

Finke, G., Clauss, A. and Gunn, E. A (1984). *Two-Commodity Network Flow Approach to the Traveling Salesman Problem.* Congressus Numerantium, Vol. 41, pp. 167-178.

Fox, K., Gavish, B., Graves, S. (1980). *An n-constraint formulation of the (time-dependent) travelling salesman problem.* Operations Research, Vol. 28, pp. 1018-1021.

Gavish, B. and Graves, S.C. (1978*). The travelling salesman problem and related problems*. Operations Research Center, MIT, Cambridge, MA. Working Paper OR-078-78.

Held, M, and Karp, R.M. (1969*). The traveling-salesman problem and minimum spanning trees*. New York : IBM Systems Research Institute.

Laporte G. (1992). *The Traveling Salesman Problem: An overview of exact and approximate algorithms*. European Journal of Operational Research, 59, 1992, pp. 231-247.

ISBN: 978-618-80361-1-6

Miller, C.E., Tucker, A.W. and Zemlin, R.A. (1960). *Integer programming formulations and traveling salesman problems*. Journal of the Association for Computing Machinery, Vol. 7, pp. 326-329.

Saharidis, G., & Ierapetritou, M. (2010). *Improving Benders decomposition using maximum feasible subsystem (MFS) cut generation strategy.* Computers & Chemical Engineering, 34(8), 1237-1245.

Saharidis, G., Minoux, M., & Ierapetritou, M. (2010). *Accelerating Benders method using covering cut bundle generation*. International Transaction in Operational Research, 17(2), 221-237.

Saharidis G.K.D. (2014). *Review of solution approaches for the symmetric traveling salesman problem*. International Journal of Information Systems and Supply Chain Management. 2014, to appear.

Vadja, S. (1961). *Mathematical Programming*. London : Addison-Wesley Publishing Company.

Wong, R. (1980). *Integer programming formulations of the travelling salesman*. In Proceedings of the IEEE Conf. on Circuits and Computers, pp. 149–152.

ISBN: 978-618-80361-1-6