ON THE SYMMETRIC TRAVELLING SALESMAN PROBLEM: THEORY AND COMPUTATION

Martin Grötschel
Universität Bonn, Bonn
and
Manfred W. Padberg
New York University, New York

The symmetric travelling salesman problem (TSP) is the problem of finding the shortest Hamiltonian cycle (or tour) in a weighted finite undirected graph without loops. This problem appears to have been formulated some 45 years ago [14] and has been a subject of intensive investigation in combinatorial optimization during the past 25 years. The interest that this problem has received is well deserved: Many practical combinatorial problems in scheduling and production management can be formulated as or shown to be equivalent to a symmetric travelling salesman problem. On the other hand, the travelling salesman problem is of theoretical interest because it is a "hard" combinatorial problem, see [10].

In this note, we review some of the recent theoretical and computational results concerning the TSP. In doing so, we focus on the oldest approach to this problem due to Dantzig, Fulkerson and Johnson [2] who formulated the problem as a linear programming problem in zero-one variables in 1954 and used a cutting plane approach to prove the optimality of a heuristically obtained solution to a 49-city problem. We review first results which concern the facial structure of the convex hull of tours of the n-city travelling salesman problem where a tour is regarded as a point in R^m with $m = n(n-1)/2$, and where n is the number of cities of TSP. The striking result of the pertinent studies [5], [6], [7] is that the number of inequalities needed to linearly describe this convex polytope grows far worse with number n of cities than previously expected [4]. We next discuss part of the experimental results of a computational study [16] which aims empirically validating the usefulness of cutting planes for the actual solution of large-scale combinatorial optimization problems. The cutting planes used in the computational study are, however, not the cutting planes used commonly in integer programming, rather they form a subset of those inequalities for which we have theoretically established that they are required in any non-redundant description of the travelling salesman polytope by way of linear inequalities. These inequalities are in a way "strongest" cutting planes in an integer programming sense and can thus be expected to perform satisfactorily in computation. The main conclusion to be drawn from the computational study is that such (facetial) inequalities are indeed of substantial computational value in the solution of this difficult combinatorial problem and, by generalization, for other "hard" combinatorial optimization problems as well.

1. NOTATION

Let $K_n = (V,E)$ be the undirected complete graph with node-set $V = \{1,\ldots,n\}$ and edge-set $E = \{(i,j) \mid i \in V,\ j \in V,\ i \neq j\}$. A tour is either a cyclic permutation of nodes $(i_1, i_2,\ldots,i_n)$ or equivalently, a set of n edges $\{(i_1,i_2),\ (i_2,i_3),\ldots,(i_{n-1},i_n),\ (i_n,i_1)\}$ which form a Hamiltonian cycle (or tour) in K_n. A cyclic permutation of r nodes with $r < n$ or its associated edge-set is called a subtour. Algebraically a tour is described by a zero-one vector x with the convention that $x_{ij} = 1$ if the edge (i,j) is in the tour and $x_{ij} = 0$ if not. As we are dealing with undirected edges the vector x has $m = n(n-1)/2$ components. For any $S \subseteq V$ and $H \subseteq E$ we use the following abbreviations:

$N(H) = \{i \in V \mid i \text{ is incident with an edge in } H\}$

$E(S) = \{(i,j) \in E \mid i \in S,\ j \in S\}$

$x(H) = \Sigma\{x_{ij} \mid (i,j) \in H\}$

$x(S) = x(E(S))$

For $x \in \mathbb{R}$, we let $\langle x \rangle = \min\{z \in Z \mid x \leq z\}$ and $[x] = \max\{z \in Z \mid x \geq z\}$, where Z are all integer numbers.

2. THEORETICAL RESULTS

Denote by Q_T^n the travelling salesman polytope, i.e., the convex hull of zero-one points of $\mathbb{R}^m$ that correspond to tours in the complete graph K_n. The best-known valid inequalities for Q_T^n are the subtour-elimination constraints due to Dantzig et al.[3]. The <u>subtour-elimination constraint on a node-set W</u> is given by:

(1.1) $\quad x(W) \leq |W| - 1$

where $W \subseteq V$ satisfies $2 \leq |W| \leq n-2$. The following proposition summarizes the known properties of subtour-elimination constraints, see e.g. [4]:

Proposition 1.1: (i) <u>Every subtour-elimination constraint (1.1) defines a proper face of Q_T^n</u>. (ii) <u>The subtour-elimination constraints on W and V-W define</u>

<u>the same face of Q_T^n.</u> (iii) <u>The number of subtour-elimination constraints defining distinct faces of Q_T^n is equal to</u>

(1.2) $\quad \nu^S(n) = 2^{n-1} - n - 1.$

Since for any pair $W, W' \subseteq V$ satisfying $W \cup W' \neq V$ one can readily find a tour in K_n satisfying the subtour-elimination constraint on W with equality and the subtour-elimination constraint on W' with inequality, it follows that the subtour-elimination constraints define exactly $\nu^S(n)$ distinct faces of Q_T^n. Furthermore, for $n \geq 5$, the trivial inequalities $x_e \geq 0$, $e \in E$, define distinct faces of Q_T^n. Consequently, we have $2^{n-1} + n(n-3)/2 - 1$ inequalities defining distinct faces of Q_T^n. We discuss next a class of combinatorial inequalities whose total number apparently grows much faster with n than $O(2^{n-1})$.

Every vertex of Q_T^n satisfies the system of equations and inequalities

(1.3) $\quad Ax = 2e_n, \quad 0 \leq x \leq e_m$

where A is the node-edge incidence matrix of K_n, $m = n(n-1)/2$ and e_k is the vector with k entries equal to one. Consequently, the 2-matching constraints due to J. Edmonds [3] constitute valid inequalities for Q_T^n. V. Chvátal [1] has generalized this class of inequalities to a wider class of inequalities which he called comb inequalities. Both classes of valid inequalities for Q_T^n like the subtour-elimination constraints have coefficients of zeros and ones only (except for the right-hand side constant) and are special cases of the following general <u>comb inequality</u>, which has coefficients equal to 0,1 or 2: Let $W_i \subseteq V$ for $i = 0,1,\ldots,k$ satisfy

(1.4) $\quad |W_0 \cap W_i| \geq 1$ for $i = 1,\ldots,k$.

(1.5) $\quad |W_i - W_0| \geq 1$ for $i = 1,\ldots,k$.

(1.6) $\quad |W_i \cap W_j| = 0$ for $1 \leq i < j \leq k$.

(1.7) $\quad$ k odd.

Then we call $C = \bigcup_{i=0}^{k} E(W_i)$ a <u>comb</u> in K_n; W_0 is called the <u>handle</u> and the W_i for $i = 1,\ldots,k$ are called the <u>teeth</u> of the comb C. The <u>comb inequality</u> corresponding to a comb C in K_n is given by

(1.8) $\quad \sum_{i=0}^{k} x(W_i) \leq |W_0| + \sum_{i=1}^{k} (|W_i|-1) - \langle \frac{k}{2} \rangle.$

A comb C with $k = 1$ and $|W_0| = 1$ is a subtour-elimination constraint. A comb inequality is a 2-matching constraint [3] if the inequality in both (1.4) and (1.5) holds as an equality. A comb inequality is a <u>Chvátal-comb</u> [1] if the requirement (1.6) is dropped and the inequality (1.4) is required to hold as an equality. Chvátal [1] also permits k in (1.7) to be even, but then the inequality (1.8) does not involve any integerization and is trivially seen to be inessential for Q_T^n. In [6] we prove that requirement (1.6) does not exclude any Chvátal-combs that are essential for Q_T^n. Furthermore, the only undominated combs satisfying $k = 1$ yield subtour-elimination constraints and thus, in order to distinguish subtour-elimination constraints from comb inequalities, we assume throughout that (1.7) holds with $k \geq 3$. The next proposition summarizes the more readily proven property of (generalized) comb-inequalities:

Proposition 1.2: (i) <u>Every comb inequality (1.8) defines a proper face of</u> Q_T^n. (ii) <u>The comb inequalities (1.8) given by</u> $W_0, W_1, \ldots, W_k$ <u>and by</u> $V-W_0, W_1, \ldots, W_k$, <u>respectively, define the same face of</u> Q_T^n. (iii) <u>The number of comb inequalities defining distinct faces of</u> Q_T^n <u>is equal to</u>

(1.9) $$\nu^C(n) = \sum_{m=3}^{n-3} \tfrac{1}{2} \binom{n}{m} \sum_{j=3}^{n-m} \binom{n-m}{j} \sum_{\substack{k=3 \\ k \text{ odd}}}^{\min(j,m)} A_k^j \sum_{p=k}^{m} \frac{1}{k!} \binom{m}{p} A_k^p,$$

<u>where</u>

$$A_k^p = \sum_{j=0}^{k} (-1)^j \binom{k}{j} (k-j)^p.$$

Subtour-elimination constraints are intuitively readily understood, the logical implication of a comb inequality is more complicated. To get a better understanding, consider e.g. a comb inequality for $n = 8$ with $W_0 = \{1,2,3,4\}$, $W_1 = \{1,2,5,6\}$, $W_2 = \{3,7\}$ and $W_3 = \{4,8\}$, i.e.,

(1.10) $$2x_{12}+x_{13}+x_{14}x_{15}+x_{16}+x_{23}+x_{24}+x_{25}+x_{26}+x_{34}+x_{37}+x_{48}+x_{56} \leq 7.$$

Using the relations $\sum_{j=2}^{8} x_{1j} = 2$ and $x_{12} + \sum_{j=3}^{8} x_{2j} = 2$ to eliminate the variable x_{12} from (1.10) we obtain the equivalent constraint

(1.11) $$x_{34}+x_{37}+x_{48}+x_{56} \leq 3 + x_{17}+x_{18}+x_{27}+x_{28}.$$

This constraint now expresses quite clearly the logical implication of the comb inequality (1.10): If $x_{34} = x_{37} = x_{48} = x_{56} = 1$, i.e., if the travelling salesman travels on the chain [7,3,4,8] and includes the link [5,6] as well, then

$x_{17}+x_{18}+x_{27}+x_{28} \geq 1$ must hold, i.e. then the travelling salesman must choose one of the links [1,7], [1,8], [2,7] or [2,8], since otherwise there exists no round trip for the eight cities. It would, however, be incorrect to conclude that subtour-elimination constraints are all that are needed to linearly describe Q_T^n. Indeed, in the above example, it is not difficult to find those points of $\mathbb{R}^{28}$ that satisfy all subtour-elimination constraints with inequality while violating inequality (1.10). Clearly, there can be no such points with zero-one components only, but there are such points e.g. with components equal to 0, ½ and 1.

While in Propositions 1.1 and 1.2 it is asserted that the respective inequality (at least) <u>intersect</u> the polytope Q_T^n, so far nothing is said about the <u>dimension</u> of the face of Q_T^n that is defined by any particular such inequality. (It should be noted that cutting planes in general integer programming such as the ones commonly encountered in textbooks are not even guaranteed to intersect the convex hull of integer solutions, see [15] for a relevant illustration.) In [6], we prove that the dimension of the face of Q_T^n that is defined by any particular subtour-elimination or comb inequality is the largest possible, i.e. equal to the dimension of Q_T^n minus one. As customary in the literature, we say that the respective inequality defines a <u>facet</u> (= proper face of maximal dimension) of Q_T^n and with this notation, the following theorem is proven in [5], [6], [7]:

Theorem 1.3: (i) <u>The trivial inequalities $x_{ij} \geq 0$ define facets of Q_T^n for all unordered pairs i, j.</u>

(ii) <u>The subtour-elimination constraints (1.1) define facets of Q_T^n.</u>

(iii) <u>The comb inequalities (1.8) define facets of Q_T^n.</u>

(iv) <u>Trivial inequalities, comb inequalities and subtour-elimination constraints are pairwise non-equivalent.</u>

In the above theorem, we assume that $n \geq 6$ is an arbitrary integer. Having thus established that the above inequalities <u>really matter</u> in defining the travelling salesman polytope, it is of interest to calculate the number of subtour-elimination constraints and comb inequalities. In order to get an idea about the comparative growth of the number $\nu^S(n)$ of subtour-elimination constraints and of the number $\nu^C(n)$ of comb constraints we have computed the respective numbers and tabulated them in Figure

(For n ≥ 20 we give only the order of magnitude.) As n gets large, $\nu^S(n)$ becomes marginal by comparison to $\nu^C(n)$ and for n ≥ 8 the linear system is already astronomically large - even though 8-city problems can generally be solved by inspection. To make matters worse, we know that for n ≥ 10 (and, possibly, even for n ≥ 8) the above set of linear inequalities does not completely describe Q_T^n and probably a still far greater number of linear inequalities is needed to achieve that result. For the uninitiated, these results will indicate that an algorithmic approach to TSP based on linear inequalities must fail.

3. COMPUTATIONAL RESULTS

Actual computational experience indicates strongly that a linear-programming based proceeding works very well for this difficult combinatorial optimization problem. While it is of course impossible to work with an explicit in-core or out-of-core representation of $2 \cdot 10^{179}$ inequalities during the solution of a 120-city problem, it is entirely feasible to "activate" an increasing number of inequalities as the computation proceeds. The question then becomes how many of the inequalities are eventually activated during computation and here the computational experience indicates that very few inequalities are required to solve to optimality or to establish near-optimality in a great number of travelling salesman problems. This statement is based on a recent computational study [16] in which 74 different TSP's were tried (in 44 cases

n	$\nu^S(n)$	$\nu^C(n)$
6	25	60
7	56	2100
8	119	41420
9	246	667800
10	501	8841970
15	16368	1993711339620
20	$0.5 \cdot 10^6$	$1.5 \cdot 10^{18}$
30	$0.5 \cdot 10^9$	$1.5 \cdot 10^{31}$
40	$0.5 \cdot 10^{12}$	$1.5 \cdot 10^{45}$
50	$0.5 \cdot 10^{15}$	10^{60}
59	$0.3 \cdot 10^{18}$	10^{74}
120	$0.6 \cdot 10^{36}$	$2 \cdot 10^{179}$

Figure 1: Comparative Growth of $\nu^S(n)$ and $\nu^C(n)$

optimality was established, in the remaining 30 cases excellent lower bounds on the minimum tour length were obtained).

The algorithmic proceeding is virtually identical to the one of Dantzig et al. [2] except that the man-machine interactive features have been eliminated. This means that one starts by finding a possibly optimal tour by way of a heuristic. (The heuristic used in the study is due to Lin and Kerningham [13].) This solution is used to initialize the linear program given by (1.3) and specially written sub-routines are used to activate new inequalities when they are needed to cut off a basic feasible solution to the current linear programming problem. The whole procedure works like a primal (rather than dual) cutting plane algorithm. This permits one to find better tours to the problem, if the heuristically obtained starting solution was non-optimal even though one may not terminate by actually proving optimality of a tour. This is simply due to the fact that at present no complete linear description of Q_T^n is known and that in the study it was not considered to permit cutting planes from general integer programming theory. Instead - if the sub-routines do not find a necessary next cutting plane of the types described in Section 2 - the current linear programming problem is solved to optimality yielding a lower bound on the optimum tour length.

In order to evaluate the value of inequalities (1.1) and (1.8) towards the goal of proving optimality, the linear program (1.3) is solved in a first run. Then in a second run, the same problem is run using the constraint-activating procedure described above. This yields two values: VALUE 1 is the objective function value without cuts and VALUE 2 is the objective function value with cuts. If TOUR denotes the minimum length tour of the problem, then the following ratio is a good proxy for measuring the added value of the additional work: RATIO = (VALUE 2 - VALUE 1)/(TOUR - VALUE 1). Note that RATIO is zero if no improvement is obtained (e.g. if no constraint was generated), while RATIO is one if the constraint-generation procedure terminates with the optimal tour. RATIO is, of course, always between zero and one and due to taking both differences and a ratio, the measure is invariant under scaling and translating the data. This is of particular importance since a single ratio, e.g. VALUE 2/TOUR, can be made to "look arbitrarily good" by a simple translation of the data (distances).

The computational results on a number of test problems from the literature are summarized in Table I. The heading "without cuts" refers to the solution of the (initial) linear program: TIME1 is the CPU-time in seconds, PIVOT1 the pivot count, VALUE1 the objective function. TOUR refers to the minimum length tour or the value of the best tour found by the heuristic. The heading "with cuts" refers to the constraint-generation procedure: VALUE2 is the objective function value of the linear program with cuts, the first column under PIVOT2 refers to the total number of pivots, the second column under PIVOT2 refers to the number of pivots carried out after the default in the constraint-generation procedure (i.e. the second column is counted already in the first). CUTS specifies the total number of cuts generated in the run, its second column the number of cuts that were dropped again after defaulting. TIME2 is total execution time to termination in CPU-seconds. RATIO is the value discussed in the introduction to this section. All problems were executed on the IBM 370-168 MVS of the IBM T.J. Watson Research Center in Yorktown Heights.

DAN42 is the 42-city version of the 49-city problem due to Dantzig, Fulkerson and Johnson [2]. The solution was proven to be optimal in 3.10 seconds of CPU-time after adding 9 constraints. GRO48 is a 48-city problem due to Grötschel [5] (48 cities with distances given in Shell's Roadatlas of Germany). After 9.16 seconds of CPU-time the program terminated with a lower bound of 5032 for the optimum tour; the best tour found by the heuristic has a length of 5046. HEL48 is the 48-city problem due to Held and Karp [8]. The solution was proven to be optimal in 4.30 seconds of CPU-time after adding 10 constraints. TOM57 is the 57-city problem due to Thompson and Karg [9]. After 10.40 seconds of CPU-time a lower bound of 12940 for the optimum length tour of 12955 was obtained. (Optimality was proven by Held and Karp [8]). KROL70 is a 70-city problem due to Krolak [12]. After 31.91 seconds of CPU-time a lower bound of 674 on the heuristically obtained best tour of length 675 was obtained. GRO120 is a 120-city problem due to Grötschel [5], who proved 6942 to be the minimum-length tour using the same general algorithmic approach as described here. (The problem has 120 cities with distances given in the Deutscher Generalatlas, Mairs Geographischer Verlag, Stuttgart 1967/8). In the case of this problem, the heuristic

[13] obtained in 158 CPU-seconds and 20 tries a best tour with length 6951. When this suboptimal solution was used as a starting solution to the linear programming code the program obtained after roughly 3 minutes of CPU-time a lower bound of 6929. When the optimal tour was used as a starting solution to the linear programming code, the program obtained after roughly 4 minutes a better bound of 6939. KNU121 is a supersparse 121-city problem due to Knuth [11]. The code encountered very early an "unknown" vertex and defaulted to solving the amended linear program. 7.25 seconds of CPU-time were used to obtain a lower bound of 344 on the optimum tour length of 349 published in the New York Times. LIN318 is a 318-city problem the data of which are published in [13]. The data come from an actual problem involving the routing of a numerically controlled drilling-machine through three identical sets of 105 point each plus three outliers. As the drilling is done by a pulsed laser, drilling time is negligible and the problem becomes a standard travelling salesman problem. After several runs (including runs with the heuristic) the tour of length 41349 was found. When started with this solution, the program terminated after roughly 30 minutes of CPU-time with a lower bound of 41237 for the optimum tour. Consequently, the "gap" to the best tour found is 112 and thus, this tour is at worst ¼% off the absolute optimum. Moreover, units being milliinches, the best tour found is at worst $\frac{1}{10}$ inch off the absolute optimum through the 318 points. From a practical point-of-view, this solution must be considered more than satisfactory and if the economics of this particular application demanded a true optimum solution, one would have - in view of the small remaining gap of 112 - a better than even chance to solve this problem exactly by any good branch-and-bound code.

	WITHOUT CUTS				WITH CUTS						
PROBLEM	TIME1	PIVOT1	VALUE1	TOUR	VALUE2	PIVOT2		CUTS		TIME2	RATIO
DAN 42	2.57	30	641	699	699	37	0	9	0	3.10	1.0
GRO 48	4.09	33	4769	5046	$5031^1/16$	83	9	32	8	9.16	0.95
HEL 48	3.69	33	11197	11461	11461	38	0	10	0	4.30	1.0
TOM 57	7.79	44	$12633^1/2$	12955	12940	61	4	22	1	10.40	0.95
KROL 70	16.33	53	$623^1/2$	675	$673^1/4$	120	8	44	10	31.91	0.98
GRO 120	111.20	69	$6662^1/2$	6942	$6938^7/26$	243	4	75	17	221.50	0.99
				6951	$6928^1/4$	166	17	49	4	171.44	0.92
KNU 121	4.54	45	328	349	$343^1/2$	74	13	10	1	7.25	0.76
LIN 318	670.8	251	$38765^1/2$	41349	$41236\frac{112}{240}$	578	70	171	64	1751.46	0.96

TABLE 1

<u>References</u>:

[1] Chvátal, V.: "Edmonds Polytopes and Weakly Hamiltonian Graphs", Mathematical Programming, 5 (1973), 29-40.

[2] Dantzig, G.B., D.R. Fulkerson and S.M. Johnson: "Solution of a Large-Scale Travelling Salesman Problem", Operations Research, 2 (1954), 393-410.

[3] Edmonds, J.: "Maximum Matching and a Polyhedron with 0,1 Vertices", Journal of Research of the National Bureau of Standards, Sect. B 69 (1965), 125-130.

[4] Gomory, R.E.: "The Travelling Salesman Problem", in Proceedings of the IBM Scientific Computing Symposium on Combinatorial Problems, IBM Data Processing Division, White Plains, New York, 1964.

[5] Grötschel, M.: "Polyedrische Charakterisierungen Kombinatorischer Optimierungsprobleme", Dissertation, Universität Bonn, 1977.

[6] Grötschel, M. and M.W. Padberg: "On the Symmetric Travelling Salesman Problem I: Inequalities", T.J. Watson Research Report, IBM Research Center, Yorktown Heights, New York, 1977.

[7] Grötschel, M. and M.W. Padberg: "On the Symmetric Travelling Salesman Problem II: Lifting Theorems and Facets", T.J. Watson Research Report, IBM Research Center, Yorktown Heights, New York, 1977.

[8] Held, M. and R.M. Karp: "The Travelling Salesman Problem and Minimum Spanning Trees, Part I", Operations Research 18 (1970), 1138-1162; "Part II', Mathematical Programming 1 (1971), 6-26.

[9] R.L. Karg and G.L. Thompson: "A Heuristic Approach to Solving Travelling-Salesman Problems", Management Science, 10 (1964), 225-247.

[10] R.M. Karp: "Reducibility among Combinatorial Problems" in R.E. Miller and J.W. Thatcher (eds.): Complexity of Computer Computations, Plenum Press, New York, 1972, 85-103.

[11] Knuth, D.: "The Travelling Salesman Problem", illustrative example in W. Sullivan: "Frontiers of Science, from Microcosm to Macrocosm", The New York Times, February 24, 1976, p. 18.

[12] Krolak, P., Felts, W. and G. Marble: "A Man-Machine Approach towards Solving the Travelling Salesman Problem", CACM 14 (1971), 327-334.

[13] Lin, S. and B.W. Kernighan: "An Effective Heuristic Algorithm for the Travelling-Salesman Problem", Operations Research 21 (1973), 498-516.

[14] Menger, K.: "Botenproblem", in K. Menger (ed.): Ergebnisse eines mathematischen Kolloquiums, Leipzig 1932, Heft 2, 11-12.

[15] Padberg, M.W.: "On the Facial Structure of Set Packing Polyhedra", Mathematical Programming, 5 (1973), 199-215.

[16] Padberg, M.W. and S. Hong: "On the Symmetric Travelling Salesman Problem: A Computational Study", T.J. Watson Research Report, IBM Research Center, Yorktown Heights, New York, 1977.