Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Discrete Mathematics 36 (2010) 351–358
Electronic Notes in DISCRETE MATHEMATICS
www.elsevier.com/locate/endm

An integer programming approach for the time-dependent TSP

Juan José Miranda-Bront [1] Isabel Méndez-Díaz [1]

Department of Computer Science, University of Buenos Aires
Buenos Aires, Argentina

Paula Zabala [1]

Department of Computer Science, University of Buenos Aires
and Consejo Nacional de Investigaciones Científicas y Técnicas
Buenos Aires, Argentina

Abstract

The Time-Dependent Travelling Salesman Problem (TDTSP) is a generalization of the traditional TSP where the travel cost between two cities depends on the moment of the day the arc is travelled. In this paper, we focus on the case where the travel time between two cities depends not only on the distance between them, but also on the position of the arc in the tour. We consider the formulations proposed in Picard and Queryanne [8] and Vander Wiel and Sahinidis [10], analyze the relationship between them and derive some valid inequalities and facets. Computational results are also presented for a Branch and Cut algorithm (B&C)that uses these inequalities, which showed to be very effective.

Keywords: TDTSP, Combinatorial Optimization, Branch and Cut.

[1] Email: {jmiranda,imendez,pzabala}@dc.uba.ar

doi:10.1016/j.endm.2010.05.045

1 Introduction

The *Time-Dependent Travelling Salesman Problem* (TDTSP) is a generalization of the classical Travelling Salesman Problem (TSP) in which the cost of the travel between two cities depends not only on the distance between the cities, but also on the time of the day the arc is travelled.

In its simplest version, TDTSP assumes that the travel time between any two cities is one time period, meaning that the cost function depends on the distance between the cities and on the position of the arc in the tour. The time-dependent cost function for each arc can be expressed as a step function with one constant value for each time period, i.e., c_{ijk}, $k = 1, \ldots, n$. This version of the TDTSP can be used to model different scheduling and assignment problems. Several formulations have been proposed for this problem in Picard and Queyranne [8], Fox et al [4] and Vander Wiel and Sahinidis [10]. To the best of our knowledge, the only exact algorithms in the literature are proposed in [8] and [10], solving instances with at the most 20 vertices, and in Bigras et al. [3] where they are able to solve scheduling instances with up to 50 vertices using a Branch and Bound algorithm.

In [8], one of the models is a three-index integer linear programming formulation. Méndez-Díaz et al [6] tested it for *Travelling Deliveryman Problem* (TDP) instances, and the results obtained are quite reasonable.

These results suggest that both models look promissory to be used in a B&C algorithm. A Branch and Bound algorithm is developed in [8] and a Branch and Cut algorithm is proposed in [10] to solve the master problem of the Benders decomposition, but they only use general purpose cuts for a restricted set of inequalities. They suggest that, as future research, it would be interesting to study the polyhedron of the TDTSP. The aim of this paper goes in that direction. We consider the models presented in [8] and [10]. Since both models are a linearization of QAP, it results that their polytopes are strongly related. We derive some families of valid inequalities and evaluate them in a B&C algorithm.

2 Models

One of the models is proposed in [8]. It uses a set of binary decision variables y_{ijk}, where $y_{ijk} = 1$ if city j is visited in time period k after city i was visited in time period $k-1$. By forcing vertex 0 to be the depot, we can remove from the formulation the variables y_{ij0} $\forall i \geq 1$, y_{ijn} $\forall j \geq 1$, y_{i0k} $\forall k \leq n-1$, y_{0jk} $\forall k \geq 1$, given that they always take value zero.

As we mentioned in the introduction, we also consider the TDTSP formulation (VW) proposed in [10]. The model is a linearization of the QAP presented in [8], and it can be seen as the problem of finding the shortest constrained path in a directed multi-partite graph. We consider this formulation in a slightly different way than the one in [10] because we force the vertex v_0 to be the depot.

The QAP formulation uses a set of binary decision variables x_{ik}, where $x_{ik} = 1$ if city i is assigned to time period k, and $x_{ik} = 0$ otherwise. This model is quadratic because of the presence of the product between x_{ik-1} and x_{ik} in the objective function for each possible combination of (i, j) and k.

In [10], variables x_{ik} are referred as the *assignment variables*. To linearize the objective function of the QAP, they define the *transition variables*, y_{ijk}, which have the same meaning as the ones defined in the previous section. Moreover, they prove (see Proposition 1 of [10] for a detailed proof) that $y_{ijk} = 1$ if and only if $x_{ik-1}x_{jk} = 1$, even when variables y_{ijk} are considered as positive continuous variables. The advantage of this linearization is that it only introduces continuous variables to the original formulation of the QAP. See [10, Section 1] for a detailed explanation and examples.

These two models are strongly related. It is quite easy to see that (PQ) is the projection of (VW) onto variables y_{ijk}, and Gouveia and Voss [5] prove that these formulations are equivalent in terms of the linear relaxation. Formulation (VW) expresses each assignment variable, x_{ik}, in terms of the transition variables y_{ijk}. Considering the results shown in Balas and Oosten [2], we know that there is a 1-1 correspondence between the faces of (PQ) and the faces of (VW). Moreover, if P_{PQ} and P_{VW} are the polytopes associated with models PQ and VW, respectively, we can also state that $dim(P_{PQ}) = dim(P_{VW})$. From Müller [7] we also know that the dimension of P_{PQ} is $n(n-1)(n-2)$. Then, if an inequality is valid for P_{PQ}, it follows that it is also valid for P_{VW} and vice versa. For the sake of notation, we will refer to both P_{PQ} and P_{VW} as P^n_{TD}.

3 Polyhedral results

In this section we present new families of inequalities that are valid for both formulations.

3.1 Polinomial-sized families

First, we introduce four families of valid inequalities that establish upper bounds on the value of variables y_{ijk} in terms of incoming and outgoing arcs. The symbol δ_{ij} stands for the Kronecker's delta.

Proposition 3.1 *For $i, j, l = 1, \dots, n$, $i \neq j \neq l$, $k = 2, \dots, n-1$, inequalities*

$$y_{ijk} \leq y_{lik-1} + \sum_{\substack{t=1 \\ t \neq k-1,k,k+1}}^{n-1} \sum_{\substack{w=1 \\ w \neq i,j,l}}^{n} y_{lwt} + (1 - \delta_{kn-1}) y_{l0n}$$

$$y_{ijk} + y_{jik} \leq y_{lik-1} + y_{ljk-1} + \sum_{\substack{t=1 \\ t \neq k-1,k,k+1}}^{n-1} \sum_{\substack{w=1 \\ w \neq i,j,l}}^{n} y_{lwt} + (1 - \delta_{kn-1}) y_{l0n}$$

are valid for P^n_{TD}.

Proposition 3.2 *For $i, j, l = 1, \dots, n$, $i \neq j \neq l$, $k = 1, \dots, n-2$, inequalities*

$$y_{ijk} \leq y_{jlk+1} + \sum_{\substack{t=1 \\ t \neq k-1,k,k+1}}^{n-1} \sum_{\substack{w=1 \\ w \neq i,j,l}}^{n} y_{wlt} + (1 - \delta_{k1}) y_{0l0}$$

$$y_{ijk} + y_{jik} \leq y_{ilk+1} + y_{jlk+1} + \sum_{\substack{t=1 \\ t \neq k-1,k,k+1}}^{n-1} \sum_{\substack{w=1 \\ w \neq i,j,l}}^{n} y_{wlt} + (1 - \delta_{k1}) y_{0l0}$$

are valid for P^n_{TD}.

3.2 Time-dependent cycle inequalities

We now introduce a new family of valid inequalities, based on the idea of the well known cycle inequalities for the asymmetric TSP. The difference is that they include the time dependency factor. As we mentioned before, cycles are not allowed by the formulations considered in this paper. However, this family can be used to cut fractional solutions by including them in a B&C algorithm. Indeed, in the next section we will reinforce these inequalities by applying a lifting procedure. We will refer to this family of inequalities as the Time-Dependent Cycle Inequalities (TDCI). For the sake of notation, we express them in terms of both variables x_{ik} and y_{ijk}.

Proposition 3.3 (TDCI) *Let $C =< v_1, v_2, \dots, v_l, v_{l+1} = v_1 >$, $l \leq n$, be a simple cycle with transitions between consecutive vertices taken in time intervals $k, k+1, \dots, k+l-1$, $k+l \leq n$. Then, inequality*

$$\sum_{i=1}^{l} y_{v_i v_{i+1} k+i-1} \leq \sum_{i=1}^{l-1} x_{v_{i+1} k+i} \tag{1}$$

is valid for P^n_{TD}.

The TDCI do not define facets for P^n_{TD}. However, similarly to the cycle inequalities for the ATSP, TDCI define facets of a projection of P^n_{TD} over some specific variables. Let $C =< v_1, v_2, \ldots, v_l, v_{l+1} = v_1 >$, $l \leq n$ and k as defined in Proposition 3.3. We define the following set of variables:

- $F_1 = \{y_{v_l v_j k+j-2} = 0 \ : \ j = 2, \ldots, l-1\}$
- $F_2 = \{y_{v_l v_j k+l-1} = 0 \ : \ j = 2, \ldots, l-1\}$
- $F_3 = \{y_{v_i v_j k+j-2} = 0 \ : i = 3, \ldots, l-1, \ j = 2, \ldots, i-1\}$
- $F_4 = \{y_{v_i v_j k+i-1} = 0 \ : i = 2, \ldots, l-1, \ j = 1, \ldots, i-1\}$
- $F_5 = \{y_{v_1 v_j k+j-2} = 0 \ : \ j = 3, \ldots, l\}$,

and $P^n_{TD}(C, k) = P^n_{TD} \cap F_1 \cap F_2 \cap F_3 \cap F_4 \cap F_5$ as the projected polytope. We now state the following results.

Theorem 3.4 *Let $C =< v_1, v_2, \ldots, v_l, v_{l+1} = v_1 >$, $l \leq n$ and k as defined in Proposition 3.3. The dimension of $P^n_{TD}(C, k)$ is $n(n-1)(n-2)-(l+1)(l-2)$.*

Theorem 3.5 *Let $C =< v_1, v_2, \ldots, v_l, v_{l+1} = v_1 >$, $l \leq n$ and k as defined in Proposition 3.3. The TDCI (1) are facet-defining for $P^n_{TD}(C, k)$.*

As regards for the separation problem for the TDCI, it can be solved in polynomial time. The main idea consists on applying a straightforward implementation of a maximum-path algorithm over the multipartite graph defined in [8] for each pair of vertices in this graph, starting and ending in the same $v \in V$, but in different time periods. The following result holds.

Theorem 3.6 *The time-dependent cycle inequalities can be separated in polynomial time.*

3.3 Lifted time-dependent cycle inequalities

Based on the ideas from Balas and Fischetti [1], from Proposition 3.5 we can derive facets of P^n_{TD} applying a maximum sequential lifting over the variables present in $F_1, \ldots, F_5$. It is well known that the order in which variables are lifted generates different inequalities. Indeed, we lifted these variables in five different ways to obtain five families of valid inequalities. Due to space limitations, we only present one of them.

Proposition 3.7 *Let $C =< v_1, v_2, \ldots, v_l, v_{l+1} = v_1 >$, $l \leq n$ and k as defined in Proposition 3.3. Then, inequality*

$$\sum_{i=1}^{l} y_{v_i v_{i+1} k+i-1} + \sum_{i=3}^{l} \sum_{j=2}^{i-1} y_{v_i v_j k+j-2} + \sum_{j=3}^{l} y_{v_1 v_j k+j-2} \leq \sum_{i=1}^{l-1} x_{v_{i+1} k+i} \qquad (2)$$

defines a facet of P^n_{TD}.

4 Computational Results

In order to evaluate the strength of these inequalities, we develop a B&C algorithm for the model PQ. The cutting phase always considers inequalities from Section 3.1. Due to their similarity, each lifted TDCI inequality is tested independently from the others, and we report the results for the family that produces the best overall running time: the one from Proposition 3.7. As regards the separation of these inequalities, we develop a heuristic based on the results for the TDCI inequalities. We also incorporate a dynamic programming based primal heuristic, which generates a feasible integer solution at each node of the enumeration tree.

For the computational experiments, we consider benchmark instances from TSPLIB and the ones considered in Rubin and Ragatz [9], and we compare our B&C algorithm with the default one from CPLEX 10.1.

In Table 1 we show the computational times for the TSPLIB instances, considered as TSP and TDP instances as well. A cell filled with $(***)$ means that the instance was not solved within two hours. The main message of this table is that our B&C algorithm outperforms CPLEX in almost all the instances considered. This relies on the fact that the inequalities incorporated to the cutting phase are quite effective, especially the lifted TDCI. The best gains are obtained at the root node, where the gap with respect to the optimal solution is considerable reduced.

The most interesting results are the ones regarding instances with 29 vertices or more. CPLEX's default B&C algorithm is able to solve to optimality only two of the ten instances (bayg29, bays29, ftv33, ftv35 and ftv38; TSP and TDP version) within two hours, while our B&C algorithm solves nine of them. Although it is not reported due to space limitations, the number of tree nodes explored by CPLEX's algorithm is extremely higher than the one for our B&C algorithm.

In Table 2 we show the average computational times for the scheduling instances from Rubin and Ragatz [9]. The average value is calculated over eight instances for each $n = 15, 25, 35, 45$. We slightly modify the original instances by discarding the corresponding due dates, which results in $1|s_{ij}|\sum C_j$ scheduling instances (equivalent to TDP). These results are aligned with the ones from the previous table. Our B&C performs better than CPLEX's default algorithm, both in the computational times and the number of nodes explored. It is important to note that this difference is significantly higher when $n = 45$, where the time reductions of our B&C algorithm is close to the 70%. Again, this behavior is due to the strengthening of the bounds produced

Table 1
Computational times (in seconds) for TSP and TDP instances from TSPLIB.

Instance	n	TSP		TDP	
		BC	CPLEX	BC	CPLEX
bayg29	29	1131.5	$***$	3044.92	$***$
bays29	29	1289	$***$	1265.32	4296.7
burma14	14	5.14	3.41	0.44	5.01
fri26	26	160.23	1521.31	86.67	1469.88
gr17	17	33.42	423.22	3.21	30.36
gr21	21	4.87	168	9.44	185.01
gr24	24	28.04	362.83	11.75	347.02
ulysses16	16	17.04	6.77	5.44	12.26
ulysses22	22	97.48	160.35	333.49	208.45
br17	17	101.19	34.36	101.83	40.59
ftv33	33	731.43	$***$	807.33	$***$
ftv35	35	1440.43	4418.91	965.34	$***$
ftv38	38	4756.65	$***$	$***$	$***$

by the cutting phase and the primal heuristic.

Table 2
Average computational times (in seconds) for scheduling instances.

Instances	n	BC	CPLEX
PROB40x.TXT	15	0.33	1.75
PROB50x.TXT	25	6.58	10.11
PROB60x.TXT	35	88.53	94.36
PROB70x.TXT	45	439.18	1258.8

5 Conclusions

In this paper we consider the TDTSP formulations of Picard and Queyranne [8] and Vander Wiel and Sahinidis [10]. We analyze both polytopes, and derive ten families of valid inequalities for both models. We generalize the idea of the well-known cycle inequalities for the ATSP, and derive five families of facets by applying a lifting procedure. We develop a Branch and Cut algorithm in order to evaluate these inequalities which, together with a primal heuristic, prove to be very effective. The overall approach produces good computational results over different benchmark instances with respect to CPLEX's default algorithm. As future research, it would be interesting to analyze the complexity of the separation problems for the lifted TDCI in order to improve the separation routines implemented so far. In this direction, it would also be important to develop an initial heuristic and to test different branching strategies, aiming to speed up the resolution times of the Branch and Cut algorithm.

References

[1] E. Balas and M. Fischetti. Lifted cycle inequalities for the asymmetric traveling salesman problem. *Math. Oper. Res.*, 24(2):273–292, 1999.

[2] E. Balas and M. Oosten. On the dimension of projected polyhedra. *Discrete Appl. Math.*, 87(1-3):1–9, 1998.

[3] L.P. Bigras, M. Gamache, and G. Savard. The time-dependent traveling salesman problem and single machine scheduling problems with sequence dependent setup times. *Discrete Optimization*, 5(4):685–699, 2008.

[4] K. R. Fox, B. Gavish, and S. C. Graves. An n-constraint formulation of the (time-dependent) traveling salesman problem. *Oper. Res.*, 28(4):1018–1021, 1980.

[5] L. Gouveia and S. Voss. A classification of formulations for the (time-dependent) traveling salesman problem. *European Journal of Operational Research*, 83(1):69–82, May 1995.

[6] I. Méndez-Díaz, P. Zabala, and A. Lucena. A new formulation to the Traveling Deliveryman Problem. *Discrete Appl. Math.*, 2008. In press.

[7] M. C. Müller. Das dynamische travelling salesman problem. Master's thesis, Universität Kaiserslautern, Fachbereich Mathematik, August 1996.

[8] J.C. Picard and M. Queyranne. The time-dependent traveling salesman problem and its application to the tardiness problem in one-machine scheduling. *Operations Res.*, 26(1):86–110, 1978.

[9] P. A. Rubin and G. L. Ragatz. Scheduling in a sequence dependent setup environment with genetic search. *Computers & OR*, 22(1):85–99, 1995.

[10] R. J. Vander Wiel and N. V. Sahinidis. An exact solution approach for the time-dependent traveling-salesman problem. *Naval Res. Logist.*, 43(6):797–820, 1996.