Natural and Extended formulations for the Time-Dependent Traveling Salesman Problem

Maria Teresa Godinho, Luis Gouveia, Pierre Pesneau

Natural and Extended formulations for the Time-Dependent Travelling Salesman Problem

Maria Teresa Godinho [(a)], Luis Gouveia[(b)] and Pierre Pesneau [(c)]

[(a)] Centro IO & Escola Superior de Tecnologia e Gestão

Polytechnic Institute of Beja

Email: mtgodinho@ipbeja.pt

[(b)] Centro IO & Faculdade de Ciências

University of Lisbon

Email: legouveia@fc.ul.pt

[(c)] Université de Bordeaux, INRIA Bordeaux – Sud-Ouest, CNRS UMR 5251

Email: Pierre.Pesneau@math.u-bordeaux1.fr

Abstract

In this paper we present a new formulation for the Time-Dependent Travelling Salesman Problem (TDTSP). We start by reviewing well known natural formulations with some emphasis on the formulation by Picard and Queyranne (1978). The main feature of this formulation is that it uses, as a subproblem, an exact description of the n-circuit problem. Then, we present a new formulation that uses more variables and is based on using, for each node, a stronger subproblem, namely a n-circuit subproblem with the additional constraint that the corresponding node is not repeated in the circuit. Although the new model has more variables and constraints than the original PQ model, the results given from our computational experiments show that the linear programming relaxation of the new model gives, for many of the instances tested, gaps that are close to zero. Thus, the new model is worth investigating for solving TDTSP instances. We have also provided a complete characterization of the feasible set of the corresponding linear programming relaxation in the space of the variables of the PQ model. This characterization permits us to suggest alternative methods of using the proposed formulations.

1. Introduction

Consider a graph G=(V,A), where V ={1,2,..,n} and A={(i,j): i,j=1,..,n, $i \neq j$}. The Time-Dependent Travelling Salesman problem (TDTSP) is to find a minimum cost Hamiltonian circuit, starting and ending on node 1, where arc costs depend on its position in the tour. Thus, to each arc (i,j) in A and each possible position h of the arc in the tour we associate a cost c_{ij}^{h}. Clearly, an arc

(1,j) leaving the depot can be only in position 1 and an arc (i,1) entering the depot can be only in the last position. Every other arc (i,j), i,j $\neq$ 1, can be located in positions h=2,…,n-1.

The TDTSP was motivated by the following one-machine scheduling problem. Consider a set of n-1 jobs, corresponding to the nodes in the set V\{1}, to be performed on a single machine which can handle one job at a time. Transition costs c_{ij}^h occur when job i is to be processed at position h and in addition, is immediately followed by job j. We assume an idle state for the machine corresponding to the initial and final states of the machine and which will be represented by node 1. Then, we have a set up cost for any job j, given by c_{1j}^1, and a cooling cost for any job i given by c_{i1}^n. The problem is to find the cheapest sequence for performing all jobs.

Two special cases of the TDTSP are well known. The most well known case, the Asymmetric Traveling Salesman Problem (ATSP) (see, for instance, Lawler et al (1985)), is obtained when for every arc (i,j) we have $c_{ij}^h = c_{ij}$ for every h (that is, the cost of each arc does not depend on its position). The other case, the so-called Cumulative Traveling Salesman Problem (CTSP) also known as the traveling deliveryman problem, is obtained by considering $c_{ij}^h = (n-h)c_{ij}$ for every arc (i,j) and every h, where c_{ij} is a given "base" cost. The CTSP models the situation where one wants to minimize the sum of all distances from node 1 to any other node (excluding node 1). This model has applications in machine scheduling and delivery problems where one seeks to minimize the average arrival time at the customer locations. We note that in deliver applications, the cost of an arc (i,j) in position h is defined by (n-h+1) c_{ij} since in this case one wants to compute in addition the total time that the distributor is out of the depot. We note that models presented for one case are quite easily adapted for the other one and the results taken from a small computational experiment performed with the models presented in this paper, indicate that no significant difference arises when either one of the two versions is tried. Thus, we will present our models with the first cost definition mentioned above.

Exact methods for solving the CTSP are described, among others, in Lucena (1990), Fischetti et al (1992), Bianco at al (1993) and, more recently, in Bigras et al (2008), Méndez-Dias et al (2008) and Abeledo et al (2010). Lucena (1990) proposes an algorithm based on a non-linear integer programming formulation by Picard and Queyranne in which lower bounds are obtained from a Lagrangean relaxation and presents computational results for problems up to 30 nodes. A similar approach was followed by Bianco at al. (1993) which derive a Lagrangean relaxation scheme from an integer linear programming formulation also proposed by Picard and Queyranne (see next section). They solve instances with up to 35 nodes. Fischetti et al. (1992) provide a branch-and-bound algorithm based on a new integer programming formulation. The paper contains results on the cumulative matroid that are used to derive lower bounds. Problems with up to 60 nodes are solved to optimality. Bigras et al. (2008) use a branch-and-cut scheme based on a path formulation. This is equivalent to the Picard and Queyranne formulation strengthened with several classes of inequalities that are either taken from the ATSP problem (subtour elimination inequalities and 2-matching inequalities) or taken from the node packing problem. The authors apply this procedure also to the Makespan Problem and to the Total Tardiness Problem. They present results for instances taken from the literature up to 50 nodes. Méndez-Dias et al (2008) propose a new formulation which uses flow based variables as well as variables from the linear ordering problem. In the scope of a branch-and-cut algorithm they introduce several classes of valid inequalities (which are also shown to be facet defining). They produce computational results for instances with up to 40 nodes. In Abeledo et al (2010), the authors present an approach that is similar to the one presented by Bigras et al (2008) in the sense that column generation applied to a path model is also used. However, Abeledo et al (2010) use inequalities from the TDTSP. Some of these inequalities are lifted versions of inequalities from the TSP, making their method stronger in theory. They produce results taken from instances with up to 76 nodes. They also provide a polyhedral study of the TDTSP showing that one class of the inequalities used in their method are facet defining.

Several formulations for the TDTSP described in the literature (see Section 2) can be obtained by using the binary variables z_{ij}^{h} for all $(i,j) \in A$ and $h = 1,..,n$, indicating whether or not arc $(i,j) \in A$ is in the h^{th} position of the circuit. A formulation that uses only the z_{ij}^{h} variables is called a *natural* formulation. Natural formulations will be reviewed on Section 2 with some emphasis on the well known formulation by Picard and Queyranne (1978). The main feature of this formulation is that it uses, as a subproblem, an exact description of the n-circuit problem. An n-circuit is a circuit with n arcs which may repeat nodes and even arcs.

The new models discussed in this paper (see Sections 3 and 5) are built on two features: i) use a stronger subproblem, a n-circuit subproblem with the additional constraint that a given node is not repeated in the circuit and ii) combine these subproblems for all nodes. The new formulation will use extra variables (besides the z_{ij}^{h} variables) and thus, it will fall in the class of so-called *extended* formulations. Although the model has more variables and constraints than the original PQ model, the results given from our computational experiments show that the linear programming relaxation of the new model gives, for many of the instances tested, gaps that are close to zero. Thus, the new model is worth investigating for solving TDTSP instances, either by using it within available ILP packages or as the subject of determining what inequalities are implied by the linear programming relaxation of the new model and are not redundant in the linear programming relaxation of the Picard and Queyranne model. In fact, this is the topic of Section 4 and we will relate a set of such inequalities with the inequalities described in Abeledo at al. (2010). We should emphasize that our goal is not to obtain a formulation that provides fast lower bounds. The main aim is to propose a formulation that produces very tight lower bounds permitting us to get more insight on the structure of the problem (eg., projected inequalities, which subproblems are strong for a given commodity). However, in the conclusion, we will suggest some alternative ways for handling the proposed formulation."

In the following we denote the linear programming relaxation of a given model P by P_L and its linear programming bound by $v(P_L)$. We will use the designation “exact” model for a model whose linear programming relaxation only has integral extreme points. We let F(P) denote the set of feasible solutions of an integer (linear) program P. Given an integer linear programming model P defined on two sets of variables x and z, we denote by $Proj_x(F(P_L))$ the projection of the polyhedron defined by P_L into the space of the x variables, more precisely, $Proj_x(F(P_L))$ = {x: there exist z such that (x,z) is feasible for P_L}

2. Natural formulations for the TDTSP – The Picard and Queyranne Formulation

The well known formulation by Picard and Queyranne (1978), denoted by PQ in the sequel, is as follows:

$$\text{minimize} \quad \sum_{(i,j)\in A} \sum_{h=1,\dots,n} c_{ij}^h z_{ij}^h$$

$$z_{1j}^1 + \sum_{h=2,\dots,n-1} \sum_{i\in V\setminus\{1\}} z_{ij}^h = 1 \quad \text{for all } j \in V \setminus \{1\} \qquad (PQ1)$$

$$\sum_{j\in V\setminus\{1\}} z_{1j}^1 = 1 \qquad (PQ2)$$

$$\sum_{i\in V} z_{ij}^h = \sum_{i\in V} z_{ji}^{h+1} \quad \text{for all } j \in V \setminus \{1\};\ h = 1,\dots,n-1 \qquad (PQ3)$$

$$z_{ij}^h \in \{0,1\} \quad \text{for all } (i,j) \in A;\ h = 1,\dots,n \qquad (PQ4)$$

Constraints (PQ1) guarantee that each node is visited exactly once. Constraints (PQ3) state that an arc enters node j in position h if and only if another arc emanates from this node in position h + 1. Constraints (PQ2) state that one arc leaves node 1 in position 1 (similar constraints stating that one arc enters node 1 in position n are not needed). Constraints (PQ4) define the domain of the variables.

Other natural formulations for the TDTSP were proposed in Fox, Gavish and Graves (1980). The formulations proposed by Fox et al. have only O(n) constraints in contrast to the PQ formulation that has $O(n^2)$ constraints. However, Gouveia and Voss (1995) have shown that the linear programming relaxation of the PQ formulation is at least as good (stronger in some cases) than the linear programming relaxation of the best model presented in Fox et al. (1980). Furthermore, an empirical study given in Gouveia and Pires (1996) has shown that the linear programming bound provided by the PQ model produces reasonable improvements on the linear programming bounds produced by the best model by Fox et al. (1980).

In order to motivate the extended models defined in the next section, we look more closely to the system defined by (PQ2)-(PQ4). This system is composed by the network flow based equations defining a path in an adequate layered graph. We let $G_{PQ} = (V_{PQ}, A_{PQ})$ denote this graph where V_{PQ} contains two copies of node 1, 1^1 and 1^{n+1}, and n-1 copies j^h (h=2,…,n) of each node j (j = 2,…,n).

The arc set A_{PQ} is composed of all arcs (i^h, j^{h+1}), for h=2,…,n-1 and (i,j) in A plus the arcs $(1^1, i^2)$ and $(i^n, 1^{n+1})$ for all i in V\{1}.

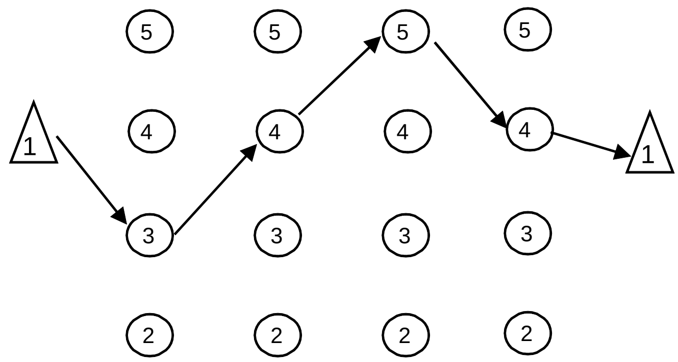

Figure 2.1 - A layered graph modeling a 5-circuit problem.

The previous remark implies that the linear programming relaxation of the model defined by (PQ2)-(PQ4) is exact since the associated constraint matrix is totally unimodular.

Any integer solution for the model defined by (PQ2)-(PQ4) is a path with n arcs (or a n-circuit in the original graph). Note, however, that the path may pass through several copies of the same original node or arc. Constraints (PQ1) impose that each node of the original graph is visited exactly once (and thus, no arc of the original graph can be used more than once) leading to a valid integer formulation for the problem. However, if we add these constraints to the system defined by (PQ2)-(PQ4), the new system is no longer integer. In fact, we lose integrality by just including a constraint (PQ1) for a single node k. In the new models proposed in this paper we will incorporate the condition “a node k is visited only once in the tour” in the n-circuit subproblem and we will also provide an exact formulation for this subproblem.

More precisely, the extended formulations discussed in this paper are based on the following two steps:

Step 1- Fixing a node k in V\{1} and strengthening the subproblem described by (PQ2)-(PQ4) in terms of the fixed node k, by requiring that this node is visited only once in the tour.

This step leads to several models with a stronger linear programming relaxation, each one based on a different node k in V\{1}. A model with a much stronger linear programming relaxation is obtained by

Step 2 - Considering together the subproblems <u>for all</u> k in V\[1}.

In the following, we denote by a n-circuit(k), a n-circuit starting and ending in node 1 and passing only once through node k.

3. Extended Formulations for the TDTSP - The Single(k) Model

For a given node k in V\{1}, let g_{ij}^{hk} be binary variables that indicate whether arc (i,j) is in position h in the circuit passing exactly once through node k. Consider the following generic extended model:

$$minimize \sum_{(i,j)\in A} \sum_{h=1,\dots,n} c_{ij}^{h} z_{ij}^{h}$$

(PQ1) and

$$\left\{(i,j):\ g_{ij}^{hk}=1\right\} \textit{ defines a } n\text{-}circuit(k) \qquad \text{(C2a)}$$

$$z_{ij}^{h}=g_{ij}^{hk} \qquad \textit{for all } (i,j)\in A, \text{ h=1,...,n.} \qquad \text{(C2b)}$$

$$z_{ij}^{h}\in\{0,1\} \qquad \textit{for all } (i,j)\in A, \text{ h=1,...,n.} \qquad \text{(PQ4)}$$

It appears to be far from easy to write an exact model only with the z_{ij}^{h} variables, for this subproblem. Later on, we will give some evidence indicating that such a model will include an exponential sized set of constraints.

However, we can write a compact exact formulation by using extra variables (an extended formulation). We propose an exact formulation that is more compact than the one presented in Godinho et al. (2008) which have also addressed this subproblem in the context of a formulation for the vehicle routing problem. The previous formulation has $O(n^4)$ variables and constraints. The new formulation has $O(n^3)$ variables and constraints (thus, we gain a decrease of one order of complexity both in the number of variables and constraints) and is based on a two-layered hop-indexed graph. The first layer represents the path before node k while the second one describes the path after node k. The two layers are linked by several copies of node k, depending on its position in the path. Figure 3.1 illustrates the adequate graph for an instance on 5 nodes and for k=4. The part below, designated by “first part of the circuit” in the remaining of the text, models the path from node 1 to node k and the part above, designated by “second part of the circuit” in the remaining of the text, models the path from node k to node 1.

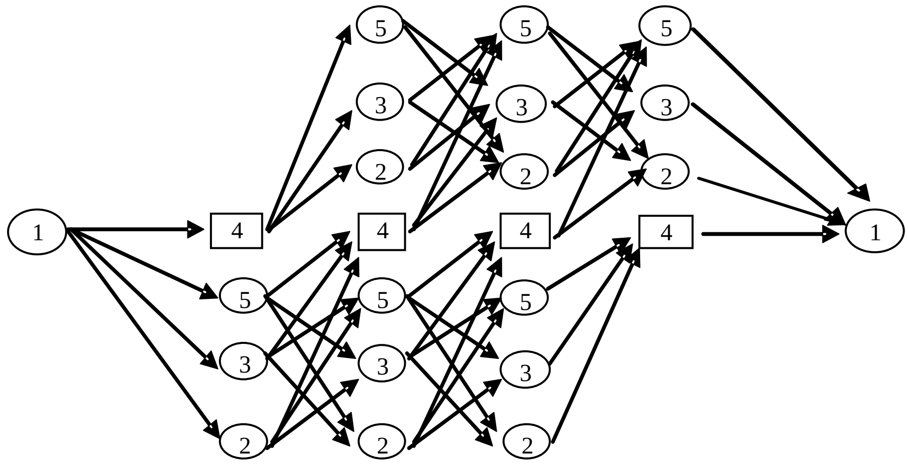

Figure 3.1 - A two-layered graph modeling a 5-circuit(4).

A straightforward shortest path reformulation based on this two-layered graph provides a compact hop-indexed (time-dependent) model for the underlying n-circuit(k) subproblem for a given node k. We associate binary $z1_{ij}^{hk}$ variables (resp. binary $z2_{ij}^{hk}$ variables) to the arcs of the sub-graph modeling a path in the first part (resp. in the second part) of the circuit. That is, the variables $z1_{ij}^{hk}$ indicate whether the arc $(i, j) \in A(j \neq 1, i \neq k)$ is in the h^{th} position in the circuit from node 1 to node 1 passing through node k <u>and is before node k</u>. They are defined only on the following three cases:

i) $h = 1, i = 1, j \in V \backslash \{1\}$

ii) $h = 2,...,n\text{-}2, i \in V \backslash \{1,k\}, j \in V \backslash \{1\}, i \neq j$

iii) $h = n\text{-}1, i \in V \backslash \{1,k\}, j = k.$

The variables $z2_{ij}^{hk}$ indicate whether arc $(i, j) \in A(j \neq k, i \neq 1)$ is in the h^{th} position in the circuit, from node 1 to node 1 passing through node k <u>and is after node k</u>. They are defined only on the following cases:

i) $h = 2, i = k, j \in V \backslash \{1,k\}$

ii) $h = 3,...,n\text{-}1, i \in V \backslash \{1\}, j \in V \backslash \{1,k\}, i \neq j$

iii) $h = n, i \in V \backslash \{1\}, j = 1.$

Using these variables, we can write the following new model for the n-circuit(k) subproblem (C2a). Note that the definition of the variables can lead to particular cases for the constraints where some terms do not appear (because they correspond to variables that are not defined). We note that similar observations hold for most of the models presented in the remainder of the text.

$$
\begin{array}{lll}
\sum_{j\in V\setminus\{1\}} z1_{1j}^{1k} = 1 & & (H2P1_k)\\
\sum_{j\in V\setminus\{1\}} z1_{ij}^{h+1,k} - \sum_{j\in V;\, j\neq k} z1_{ji}^{hk} = 0 & \textit{for all } i\in V\setminus\{1\}, i\neq k, h=1,..,n-2 & (H2P2_k)\\
\sum_{j\in V;\, j\neq k} z2_{kj}^{h+1,k} - \sum_{j\in V;\, j\neq k} z1_{jk}^{hk} = 0 & h=1,\dots,n-1 & (H2P3_k)\\
\sum_{j\in V;\, j\neq k} z2_{ij}^{h+1,k} - \sum_{j\in V\setminus\{1\}} z2_{ji}^{hk} = 0 & \textit{for all } i\in V\setminus\{1\}, i\neq k, h=2,..,n-1 & (H2P4_k)\\
g_{ij}^{hk} = z1_{ij}^{hk} + z2_{ij}^{hk} & \textit{for all } (i,j)\in A, h=1,\dots,n & (H2P5_k)\\
z1_{ij}^{hk} \in \{0,1\} & \textit{for all } (i,j)\in A, i\neq k, j\neq 1, h=1,.., n-1 & (H2P6_k)\\
z2_{ij}^{hk} \in \{0,1\} & \textit{for all } (i,j)\in A, i\neq 1, j\neq k, h=2,..,n & (H2P7_k)\\
g_{ij}^{hk} \in \{0,1\} & \textit{for all } (i,j)\in A, h=1,..,n & (H2P8_k)
\end{array}
$$

Table 3.1 Modelling a n-circuit(k).

We can obtain a new formulation for the TDTSP by replacing (C2a) with this circuit formulation. We let Single(k) denote this model. We note that when creating this model, constraints (H2P5$_k$) permit us to rewrite the linking constraints $g_{ij}^{hk} = z_{ij}^{h}$ (C2b) from the generic model describing the Single(k) model, using only the $z1_{ij}^{hk}$, $z2_{ij}^{hk}$ and z_{ij}^{h} variables. Thus, we can eliminate the g_{ij}^{hk} variables (and constraints (C2b) from the model) and the transformed linking constraints become as follows:

$$z_{ij}^{h} = z1_{ij}^{hk} + z2_{ij}^{hk} \qquad \textit{for all } (i,j)\in A \text{ and h =1,…,n}$$

For the sake of simplicity, we maintain the designation (H2P5$_k$) for these transformed equalities. Note also that due to the definition of the variables, these constraints contain the following six particular cases that will be relevant for the Appendix.

$$
\begin{array}{ll}
z_{1j}^{1} = z1_{1j}^{1k} & \textit{for all } (1,j)\in A;\\
z_{i1}^{n} = z2_{i1}^{nk} & \textit{for all } (i,1)\in A;\\
z_{ik}^{h} = z1_{ik}^{hk} & \textit{for all } (i,k)\in A; h=2,..,n-1\\
z_{kj}^{h} = z2_{kj}^{hk} & \textit{for all } (k,j)\in A; h=2,..,n-1\\
z_{ij}^{2} = z1_{ij}^{2k} & \textit{for all } (i,j)\in A; k\neq i\\
z_{ij}^{n-1} = z2_{ij}^{n-1,k} & \textit{for all } (i,j)\in A; j\neq k
\end{array}
$$

The model Single(k) becomes

$$\text{minimize} \sum_{(i,j)\in A}\sum_{h=1,\dots,n} c_{ij}^{h} z_{ij}^{h}$$

subject to (PQ1), (H2P1$_k$)-(H2P7$_k$), (PQ4).

Since the model in the $z1_{ij}^{hk}$ and $z2_{ij}^{hk}$ variables described in Table 3.1 define the path equations on the expanded network, its corresponding matrix is totally unimodular and we can conclude that its linear programming relaxation is integer. Using this fact together with the relation between the subproblems arising in Single(k) and PQ, we can state the following result

Proposition 3.1 – Let $k \in V\backslash\{1\}$. Then, $Proj_z(F(Single(k)_L)) \subseteq F(PQ_L)$.

As a consequence, we obtain

Corollary 3.1- $v(Single(k)_L) \geq v(PQ_L)$.

We will show in Section 7 that the inequality can be strict for many instances.

The subproblem in the Single(k) model guarantees that node k is not visited more than once. This suggests that the (PQ1) constraint for the same node can be eliminated from the model without altering the validity of the model. In fact, this is obvious from an integer point of view. We show next that the same happens with respect to the corresponding linear programming relaxation.

Proposition 3.2 - Let $k \in V\backslash\{1\}$. Then, constraint (PQ1) for the same k is redundant in $Single(k)_L$.

Proof: . To see this, we start by adding up constraints ($H2P2_k$) for all i and h. After eliminating equal terms (note that many of the variables appear on each side of the resulting equality) we obtain

$$\sum_{j \in V\backslash\{1\};\, j\neq k} z1_{1j}^{1k} = \sum_{h=2,\dots,n-1} \sum_{j \in V\backslash\{1\};\, j\neq k} z1_{jk}^{hk}$$

By adding the equality ($H2P1_k$) to the previous equality and by the using ($H2P5_k$) for node k we obtain the (PQ1) equality for the same node k and thus, we can eliminate it from the Single(k) model. Δ

4. Inequalities in the space of the z_{ij}^h variables implied by the Single(k) Model

One point of interest is to know what inequalities are implied by the Single(k) model in the space of the variables of the PQ formulation. Before giving a partial answer to this question we try to put in evidence the difference between the two models. The PQ formulation contains an extended description of the polyhedron describing a circuit with n arcs. The Single(k) model contains an extended description of the polyhedron describing a circuit with n arcs but with the additional constraint that node k is not repeated in the circuit. As we shall show next, the additional requirement on the Single(k) model, namely that “node k is visited exactly once” implies an exponential sized set of constraints on the space of the z_{ij}^h variables.

Before showing this, we note that the (H2P5k) permits us to eliminate the z2 variables from the model. Consider the following formulation, denoted by $PQ_{z,z1}^+(k)$, and that is defined as follows:

$$\text{minimize} \sum_{(i,j)\in A} \sum_{h=1,\ldots,n} c_{ij}^h z_{ij}^h$$

subject to (PQ1), (PQ3),(PQ4) and

$$\sum_{j\in V\setminus\{1\}} z1_{1j}^{1k} = 1 \qquad (H2P1_k)$$

$$\sum_{j\in V\setminus\{1\}} z1_{ij}^{h+1,k} - \sum_{j\in V;\, j\neq k} z1_{ji}^{hk} = 0 \quad \text{for all } i\in V\setminus\{1\}\; i\neq k, h=1,..,n-2 \qquad (H2P2_k)$$

$$z1_{1j}^{1k} = z_{1j}^1 \quad \text{for all } (1,j)\in A; \qquad (H2P9_k)$$

$$z1_{ij}^{2k} = z_{ij}^2 \quad \text{for all } (i,j)\in A; \qquad (H2P10_k)$$

$$z1_{ij}^{hk} \leq z_{ij}^h \quad \text{for all } (i,j)\in A; i,j\neq 1, k\neq j; h=3,..,n-2 \qquad (H2P11_k)$$

$$z1_{ik}^{hk} = z_{ik}^h \quad \text{for all } (i,k)\in A; i\neq 1, h=2,..,n-1 \qquad (H2P12_k)$$

$$z1_{ij}^{hk} \in \{0,1\} \quad \text{for all } (i,j)\in A, j\neq 1, h=1,.., n-1 \qquad (H2P6_k)$$

We note that the original constraints (PQ3) from the PQ model where obtained after eliminating the z2 variables. The next result relates, in some sense, the linear programming relaxations of the two models, Single(k) and $PQ_{z,z1}^{+}(k)$. We skip the proof since a more general result will be proved in the Appendix.

Proposition 4.1 - Let k $\in$ V\{1}. Then, $Proj_{z,z1}(F(Single(k)_L)) = F(PQ_{z,z1}^{+}(k)_L)$..

As a conclusion, the linear programming bounds given by the two models, Single(k) and $PQ_{z,z1}^{+}(k)$, are equal. This new formulation, $PQ_{z,z1}^{+}(k)$, can be interpreted as a combination of i) a model for a n-circuit model defined by the PQ formulation with ii) a model for a path from node 1 to node k in V\{1} and defined on the variables $z1_{ij}^{hk}$, The linking constraints ($H2P9_k$) to ($H2P12_k$) guarantee that the path to node k uses the arcs contained in the n-circuit. The equations ($H2P1_k$) guarantee that the corresponding path starts on the depot node 1. Constraints ($H2P1_k$) and ($H2P2_k$) for all h guarantee that the path ends at a copy of node k.

We may use figure 2.1 to illustrate the difference between the linear programming relaxations of the models PQ and $PQ_{z,z1}^{+}(k)$. Assume that the value of the z variables associated to the arcs in the illustrated path, 1,3,4,5,4,1 is equal to 1/2 (the solution may be completed by considering, for instance, a 1/2 path 1,2,3,2,5,1). Clearly the first 1/2 path satisfies the constraint (PQ1) for k = 4 since 1/2 enters node 4 in position 2 and 1/2 enter node 4 in position 4. However, these values for the z variables do not allow a 1 unit of flow (defined on the z1 variables) going from node 1 to the copies of node 4.

We return, now, to the question of finding out what constraints are implied by the Single(k) model on the space of the z_{ij}^h variables. Clearly we will use the new model, $PQ_{z,z1}^{+}(k)$, for deriving the inequalities. It is easier to present these inequalities in terms of the layered graph G_{PQ} associated to the PQ formulation.

Let k be a fixed node of V. Consider the following inequality

$$\sum_{h=1,\ldots,n-1}\sum_{[i^h\in V_{PQ}\setminus S,\, j^{h+1}\in S]} z_{ij}^h \geq 1 \quad for\ all\ \ S \subseteq V_{PQ}\setminus\{1^1, 1^{n+1}\}\ such\ that\ \{k^2,\ldots,k^n\}\subset S \qquad (\mathrm{Cut}_k)$$

These inequalities simply state that the solution defined in the z variables must contain at least one path from node 1 to a copy of node k. Note that it is not necessary to consider these inequalities for a set S containing copies of nodes other than k in levels 2 or n since such inequalities are easily seen to be implied by the same inequality without such copies. Note also that the same applies for a (Cut_k) inequality where S = $\{k^h$: h =2,…,n$\}$ since it is dominated by the (PQ1) inequality for node k. We denote by $PQ_z^+(k)$ the PQ formulation augmented with the (Cut_k) inequalities.

Constraints ($H2P1_k$) and ($H2P2_k$) guarantee the existence of one unit flow between node 1^1 and one of the copies $\{k^h$: h =2,…,n$\}$ of node k with arc capacities given by the z variables and defined by the constraints ($H2P9_k$) to ($H2P12_k$). Thus, by the max flow / min cut theorem we can replace this system by the set (Cut_k) and vice versa. Consequently, we have just proved that

Proposition 4.2 - Let k $\in$ V\{1}. Then, $\mathrm{Proj}_{z,z1}(F(PQ_{z,z1}^+(k)_L)) = F(PQ_z^+(k)_L)$..

By combining propositions 4.1 and 4.2, we obtain

Corollary 4.1 - Let k $\in$ V\{1}. Then, $\mathrm{Proj}_z(F(\mathrm{Single}(k)_L)) = F(PQ_z^+(k)_L)$..

Two remarks should be done after these results. The first remark refers to the following normal cut constraint known from the ATSP, lifted into the space of the z_{ij}^h variables by using the linking constraints $x_{ij} = \sum_h z_{ij}^h$ between time-dependent variables and the design variables of the ATSP,

$$\sum_h \sum_{[i\in V\setminus S,\, j\in S]} z_{ij}^h \geq 1 \quad for\ all\ \ S \subseteq V\setminus\{1\} \qquad (\text{TD-Cut})$$

These constraints are (Cut_k) constraints for S' = $\{i^h$: i $\in$ S and h =2,…,n$\}$. As observed before, these constraints are dominated by other (Cut_k) constraints. We note that this situation of having inequalities in an extended space that are stronger than facet defining inequalities in a projected space is not uncommon, see, for instance the paper by Abeledo et al. (2010). Several such inequalities were also discussed in Gouveia et al. (2009) in the context of a different problem. In fact, a suitable combination of such constraints has led to new facet defining inequalities in the projected space.

The second remark follows from the derivation of the constraints (Cut_k). We could have obtained another reduced model by eliminating the z1 variables. Then, by projection we would be able to obtain the following symmetric form of the (Cut_k) inequalities

$$\sum_{h=2,\dots,n} \sum_{[i^h \in S,\, j^{h+1} \in V_{PQ} \setminus S]} z_{ij}^h \geq 1 \quad \text{for all } S \subseteq V_{PQ} \setminus \{1^1, 1^{n+1}\};\, k^h \in S \text{ for all } h = 2,\dots,n \qquad (\text{RCut}_k)$$

Since the two reduced models are equivalent to the original model, in terms of the associated linear programming relaxations, it is natural to assume that the two sets of inequalities, (Cut_k) and (RCut_k), are equivalent. In fact, a given (RCut_k) constraint for a set S can be obtained from the (Cut_k) constraint for the same set S combined with the (PQ3) constraints for all the nodes in set S. That is, the fact that the (PQ3) inequalities are implied by the model guarantees that the second set is not needed. This fact is worth mentioning because in time-dependent models with a weaker linear programming relaxation and that do not imply the (PQ3) inequalities, such as the models presented in Fox et al., this equivalence is not valid and thus the two sets should be used to tighten the linear programming relaxation (although, as we stated, it might be easier to include the (PQ3) constraints and consider only one set).

Next we relate the class of the (Cut_k) inequalities with a large class of inequalities introduced in Abeledo et al (2010), the Admissible Flow Constraints (AFC). To make easier this relation we will use, instead, an equivalent form of the (RCut_k) inequalities.

Let S be a subset of nodes such that $\{k^h\text{: } h = 2,\dots,n\} \subset S \subset V_{PQ} \setminus \{1^1, 1^{n+1}\}$ and that does not contain any node at level 2 or n except for k^2 and k^n respectively. By using the equalities (PQ3) for node k and all h we can rewrite (PQ1) for node k in a symmetric form as follows:

$$1 = z_{k1}^n + \sum_{h=2,\dots,n-1} \sum_{i^{h+1} \in S} z_{ki}^h + \sum_{h=2,\dots,n-1} \sum_{i^{h+1} \in V_{PQ} \setminus S} z_{ki}^h$$

Combining this equality with the (RCut_k) inequality for the set S we obtain

$$\sum_{h=2,\dots,n-1} \sum_{j^h \in S \setminus \{k^2,\dots,k^n\}} \sum_{i^{h+1} \in V_{PQ} \setminus S} z_{ji}^h \geq \sum_{h=2,\dots,n-1} \sum_{i^{h+1} \in S \setminus \{k^2,\dots,k^n\}} z_{ki}^h$$

Note that the given conditions on the set S, guarantee that the set $S \setminus \{ k^h\text{: } h = 2,\dots,n\}$ does not contain any nodes at level 2 and n. Thus, the previous inequality can be rewritten as follows leading to the equivalent form of the inequalities (RCut_k)

$$\sum_{h=3,\dots,n-1} \sum_{j^h \in S \setminus \{k^2,\dots,k^n\}} \sum_{i^{h+1} \in V_{PQ} / S} z_{ji}^h \geq \sum_{h=2,\dots,n-2} \sum_{i^{h+1} \in S \setminus \{k^2,\dots,k^n\}} z_{ki}^h$$
$$\text{for all } S:\ \{k^2,\dots,k^n\} \subset S \subset V_{PQ} \setminus \{1^1, 1^{n+1}\} \qquad (\text{ERCut}_k)$$

The (AFC) inequalities can be defined as follows. Consider the graph G_{PQ} that is used to define the system (PQ2)-(PQ4) in the PQ formulation. Let X a set of nodes in that graph and E a set of arcs entering that node subset. Let C(X,E) be the set of arcs leaving X with the property that if *f* is an arc in this subset, then there exists an arc *e* in E and a path P in X such that the path (f,P,e) is elementar. The AFC inequality for the subsets X and E is as follows

$$\sum_{f \in C(X,E)} z_f \geq \sum_{e \in E} z_e \tag{AFC}$$

Now, it is easy to see that the ($ERCut_k$) are AFC inequalities, we simply let X = S and E be the set of arcs entering the set X and leaving a copy of node k.

However, there are AFC inequalities that are different from ($ERCut_k$). Consider the simple inequality obtained by having X containing node j^{h+1} and another node p^{h+2} (note that we are referring to the layered graph defining the PQ equations). Let E = {(i,j)} with $\neq p$ i. Then, the corresponding AFC inequality can be written as follows

$$z_{ij}^{h} \leq \sum_{k \neq i,p} z_{jk}^{h+1} + \sum_{k \neq i,j} z_{pk}^{h+2}$$

It is easy to see that these inequalities are not implied by the linear programming relaxation of the $PQ_z^+(k)$ model. These inequalities are a special case of the r-cycle inequalities also suggested by Abeledo et al. (2010) and which are also contained in the AFC set. With exception to the case with r = 2, the r-cycle inequalities are not contained in the set defined by the ($ERCut_k$) inequalities. The inequality described above is an r-cycle inequality for r = 3. These inequalities are considered in the branch-and-cut method described in Abeledo et al (2010).

To conclude this section we note that a special case of the (Cut_k) inequalities are already known from the literature. This case is obtained by letting $S = \{k^2,\ldots,k^n\} \cup \{p^h\}$ with $p \neq 1,k$ and $h \neq 2,n$. The inequality becomes

$$\sum_{i \neq p,k} z_{pi}^{h} \geq z_{kp}^{h-1} \qquad for\ \ all\ \ p,k \neq 1, h = 3,..,n-1 \tag{$SCut_k$}$$

for the same k as used in the model Single(k). These constraints simply state that if arc (k,p) is in the solution in position h-1, then the next arc cannot go to node k. As far as we know, these constraints were first proposed in the context of tree problems, see Gouveia (1999), and later on in Costa et al. (2009). More precisely, it was their symmetric form that is proposed for these problems since the form ($SCut_k$) is valid only if the outdegree of any node is equal to one and thus, it is not valid for tree problems. Later on they were used as valid inequalities in the context of models for routing problems (see, eg., Godinho et al (2007) and Abeledo et al (2010)). The results in these two last papers indicate that these constraints are very helpful for improving the linear programming bound of formulations similar to the PQ formulation. However, these improvements are not comparable to the full strength obtained by using all the (Cut_k) inequalities.

5. Extended Formulations for the TDTSP - The All-k Model

The Single(k) model can be generalized by considering together the n-circuit(k) systems for every k in V\{1} as follows

$$minimize \quad \sum_{(i,j)\in A}\sum_{t=1,\dots,n} c_{ij}^t z_{ij}^t$$

(PQ1) and

$$\{(i,j):\ g_{ij}^{hk}=1\}\ defines\ a\ n\text{-}circuit(k) \qquad for\ all\ k\in V\setminus\{1\} \qquad \text{(C2a)}$$

$$z_{ij}^h = g_{ij}^{hk} \qquad for\ all\ (i,j)\in A,\ \text{h=1,...,n},\ k\in V\setminus\{1\}. \qquad \text{(C2b)}$$

$$z_{ij}^h \in \{0,1\} \qquad for\ all\ (i,j)\in A,\ \text{h=1,...,n}. \qquad \text{(PQ4)}$$

As before, we consider the system in Table 3.1 for the generic part and we denote by All-k the model obtained in this way. For the All-k model, we will use the designation of the constraints of Table 3.1 without the subscript k.

As it was done for the Single(k) model, the (H2P5) equalities $z_{ij}^h = z1_{ij}^{hk} + z2_{ij}^{hk}$ permits us to eliminate the entire set of the $z2_{ij}^{hk}$ variables and obtain a model with fewer variables and with a linear programming relaxation equivalent to the original All-k model. We denote by $PQ_{z,z1}^{+}$ the model defined by (PQ1), (PQ3) and (PQ4) together with the constraints

$$\sum_{j\in V\setminus\{1\}} z1_{1j}^{1k} = 1 \qquad for\ all\ k\in V\setminus\{1\} \qquad (H2P1)$$

$$\sum_{j\in V\setminus\{1\}} z1_{ij}^{h+1,k} - \sum_{j\in V;\, j\neq k} z1_{ji}^{hk} = 0 \qquad for\ all\ i,k\in V\setminus\{1\}\ i\neq k, h=1,..,n-2 \qquad (H2P2)$$

$$z1_{1j}^{1k} = z_{1j}^1 \qquad for\ all\ (1,j)\in A\ ; k\in V\setminus\{1\} \qquad (H2P9)$$

$$z1_{ij}^{2k} = z_{ij}^2 \qquad for\ all\ (i,j)\in A\ ; k\in V\setminus\{1\} \qquad (H2P10)$$

$$z1_{ij}^{hk} \le z_{ij}^h \qquad for\ all\ (i,j)\in A; i, j\neq 1, k\in V\setminus\{1,i,j\}; h=3,..,n-2 \qquad (H2P11)$$

$$z1_{ik}^{hk} = z_{ik}^h \qquad for\ all\ (i,k)\in A; i\neq 1, k\in V\setminus\{1,i\}; h=2,..,n-1 \qquad (H2P12)$$

$$z1_{ij}^{hk} \in \{0,1\} \qquad for\ all\ (i,j)\in A, i\neq k, j\neq 1, k\in V\setminus\{1\}; h=1,..,n-1 \qquad (H2P6)$$

As before we note that by using (H2P9), (H2P10) and (H2P12) we can eliminate some of the variables z1 (e.g., variables $z1_{1j}^{1k}$ and $z1_{ik}^{hk}$) and constraints (e.g., all the constraints (H2P1)) from the model. Note also that, as shown in Section 3, inequalities (PQ1) are not needed in the model (neither in its linear programming relaxation). In fact we do this in our computations, but the reduced model as it is, is easier to explain.

The new formulation has a similar interpretation to the one given in Section 3 to formulation $PQ_{z,z1}^{+}(k)$ .It can be interpreted as a combination of i) a model for a n-circuit model defined on the z_{ij}^h variables with ii) n-1 path models, one for each node k in V\{1} and defined on the variables $z1_{ij}^{hk}$. The linking constraints (H2P9) to (H2P12) guarantee that the n-1 paths use the arcs contained in the n-circuit. For each k, the equations (H2P1) guarantee that the corresponding path starts on the depot node 1 and the constraints (H2P2) guarantee that the path ends in a copy of node k.

In the Appendix we prove the following result

Proposition 5.1 - $Proj_{z,z1}(All\text{-}k_L)) = F(PQ_{z,z1}{}^{+}{}_{L})$..

We let $PQ_{z,z1}$ denote the model PQ augmented with the (Cut_k) inequalities for all k in V\{1}. In a similar way that was used for Proposition 4.2 and the subsequent Corollary, we can show the following result

Proposition 5.2 - $Proj_z(All\text{-}k_L)) = F(PQ_z{}^{+}{}_{L})$.

We observe that a model presented in Godinho et al. (2008) can be seen as an aggregated version of the All-k model. The model in Godinho et al. also uses an exact, although less compact, model for the the n-circuit(k) subproblem (C2a). However, the linking constraints result from adding in h, the constraints (C2b) for all arc (i,j) and k in V\{1}.Thus, the linear programming relaxation of the All-k model is at least as good as the linear programming relaxation of the previous model and empirical results show that a strict domination arises for many instances.

It is interesting to contextualize this formulation in terms of the particular case of the Travelling Salesman problem (TSP). The multicommodity flow model proposed by Wong (1980), for each node k, uses a commodity from node 1 to node k and another from node k to node 1. The combined systems model, for each k, a circuit passing through node k only once. However, as shown later on in Langevin et al. (1990), a simpler and standard multicommodity flow model, proposed by Claus (1984), using only a single system (either a system representing a flow from node 1 to node k or a system representing a flow from node k to node 1), for each k, produces the same linear programming bound. The All-k model, as our computational results of Section 7 will show, produces reasonable improvements on the previous bounds. The reason for this is that the All-k model uses, for each k, an exact representation of the n-circuit(k) problem. We emphasize that the model by Wong implicitly contains a constraint stating that, for each k, the two flows are routed on exactly n arcs (this constraint is obtained by adding constraints in the model). Thus, the condition "n-circuit" is not sufficient for improving the linear programming bound. As we said before, for the improvement we need to use an exact model for the n-circuit(k) subproblem, for each k. The relations between all of these models and others, are more detailed in the forthcoming Godinho et al (2011).

6. Strengthening the All-k model

Our results will show that the linear programming relaxation of the All-k model is quite strong (for many of the instances tested we have obtained linear programming bounds that were equal to the optimal integer value). In this section we present one set of valid inequalities that permit us to improve the linear programming bounds of the All-k model (when needed to be improved). As noted in Section 4, one candidate set is the one given by the r-cycle inequalities for r > 2.

Here we propose a different set of inequalities which are taken from the linear ordering problem. The connection between this problem and the TSP has already been explored in other papers (e.g., Gouveia and Pires (2001), Sarin et al. (2005) and Gouveia and Pesneau (2006)) and with the CTSP in Méndez-Dias et al (2008). The linear ordering problem uses binary precedence pair variables, v_j^k indicating whether j is before k in the tour. The precedence-pair variables can be related to the variables used in our models as follows

$$v_j^k = \sum_{h=1,..,n-2} \sum_{p \in V;\, p \neq k} z1_{pj}^{hk} \qquad for\ all\ \ j,k = 2,..,n; k \neq j \qquad \text{(LO1)}$$

Thus, any inequality known from the linear ordering problem can be added to our models by using the constraints linking the two sets of variables. In particular, the equalities $v_j^k + v_k^j = 1$ stating that for any given pair of nodes, one is before the other in the tour, have been very useful to tighten the linear programming relaxation of several models using this set of variables. Using the previous equalities that relate the two sets of variables, these constraints can be rewritten as

$$\sum_{h=1,..,n-2} \sum_{p \in V;\, p \neq k} z1_{pk}^{hj} + \sum_{h=1,..,n-2} \sum_{p \in V} z1_{pj}^{hk} = 1 \qquad for\ all\ \ j,k = 2,..,n; k \neq j \qquad \text{(LO2)}$$

and then, added to the All-k model. We denote by E-All-k the augmented model. Our computational results will show that these equalities are also useful in the context of the model presented in this paper and lead to further reductions on the gaps, without increasing significantly the corresponding CPU times for the CTSP and "pure" time-dependent ATSP instances. For the ATSP, the increase in CPU times is, however, significant.

7. Computational Results

In this section we empirically evaluate the quality of the lower bounds given by the models presented in the previous sections. For comparing the models, we use data for complete graphs taken from the papers by Bigras et al (2008) and Méndez-Dias et al (2008). The instances from Bigras et al (2008) are instances taken from the TSP library. Mendez-Dias et al (2008) created 120 test instances, grouped in sets named A, D and S. Instances in set D are Euclidian instances; instances in set S and A are randomly generated instances. The difference between these two sets lies in the fact that instances in S are symmetric (that is arc costs for (i,j) and (j,i) are equal), while instances in A are asymmetric. For either set A, D and S, five instances are generated for every value of n considered, with $20 \leq n \leq 40$. The models that we are comparing are the models PQ, All-k, and its stronger version E-All-k when needed.

The instances taken from Bigras et al (2008) and Méndez-Dias et al (2008) are used to show that our model produces quite good gaps (close to zero, and equal to zero for many instances) for instances suggested by others. All of these instances refer to the cumulative TSP. In fact, the instances in Méndez-Dias et al (2008) refer to the cumulative TSP with the objective (n-h)*c_{ij} while the instances in Bigras et al (2008) refer to the cumulative TSP with the objective (n-h+1)*c_{ij}. However, as pointed out before, it is quite easy to change our models for the two different cases and we will compare accordingly.

We have also created pure TDTSP instances by using the costs of the instances taken from Bigras et al. and defining the cost of arc (i,j) in the h^{th} position of the circuit as $c_{ij}^h = \lceil (n-h+1)/C \rceil * c_{ij}$ for a given constant C and where c_{ij} denotes the cost of arc (i,j) in the original instance. In our experiments, we have considered C to be equal to 2, 3, 5, $\lceil n/3 \rceil$ and $\lceil n/4 \rceil$. Notice that when C=1 we obtain the CTSP and when C = n we obtain the ATSP. Thus, with these costs, we create several TDTSP instances that could be considered as being between ATSP instances and CTSP instances.

Tables 1 to 3 present the average linear programming gaps (computed as [(Optimal Value – v(P_L))/Optimal value]*100 where P denotes the model) and the CPU times for solving the corresponding linear programming relaxations, for each group of instances. The average CPU times are given in brackets below the corresponding average gaps. The linear programming relaxations were solved using the Barrier Solver of the CPLEX 11.2 software package on a Intel CoreDuo processor at 1.33 GHz computer with 4Gb of RAM. For comparison terms, the best gaps reported in the papers by Méndez-Dias et al (2008) and. by Bigras et al (2008) will also be presented (they are given in the column “BestGap”). The Mendez- Diaz et al. (2008) computational experiments were carried out on a SUN UltraSparc III workstation with 2GB of RAM running at 1Ghz.

The optimal solutions were obtained by running the model PQ with the inequalities ($SCut_k$) for a given k (which are in a polynomial number), mentioned at the end of Section 4, within the branch-and-bound algorithm of CPLEX. To solve the Méndez-Dias et al (2008) instances we used a Intel Xeon with 16 cores (on 4 processors) at 2.8 Ghz and 24 Gb of RAM, running Linux and Parallel CPLEX 12. To solve the Bigras et al (2008) and the pure TDTSP instances, we used the Intel CoreDuo processor at 1.33 GHz computer with 4Gb of RAM mentioned above.

We start by producing results for the CTSP.

TEST	Best Gap	PQ	All -k	E-All-K	TEST	Best Gap	PQ	All -k	E-All-k	TEST	Best Gap	PQ	All -k	E-All-k
D-20	4,46 (6,35)	11,48 (0,14)	0,00 (26,57)	-	**A-20**	0,00 (3,28)	0,55 (0,14)	0,00 (25,5)	-	**S-20**	0,00 (8,69)	23,73 (0,14)	0,00 (26,82)	-
D-22	0,50 (21,08)	16,30 (0,23)	0,00 (56,54)	-	**A-22**	0,00 (4,46)	1,19 (0,22)	0,00 (48,75)	-	**S-22**	0,00 (21,42)	19,86 (0,26)	0,00 (53,80)	-
D-24	0,61 (50,57)	17,03 (0,35)	0,00 (99,18)	-	**A-24**	0,15 (16,10)	2,73 (0,34)	0,00 (89,24)	-	**S-24**	0,11 (63,17)	30,54 (0,33)	0,00 (130,33)	-
D-26	0,25 (100,00)	15,52 (0,54)	0,00 (183,58)	-	**A-26**	0,00 (14,17)	1,04 (0,49)	0,00 (157,54)	-	**S-26**	0,23 (52,99)	25,02 (0,50)	0,07 (231,54)	0,00 (381,75)
D-28	0,68 (166,94)	18,94 (0,75)	0,00 (317,88)	-	**A-28**	0,47 (64,06)	2,31 (0,86)	0,08 (329,24)	0,00 (685,72)	**S-28**	0,05 (124,1)	27,02 (0,82)	0,00 (433,36)	-
D-30	0,64 (221,39)	16,43 (1,17)	0,00 (1473,67)	-	**A-30**	0,58 (69,46)	4,25 (1,23)	0,52 (502,16)	0,00 (861,39)	**S-30**	0,22 (221,22)	28,87 (1,16)	0,00 (585,47)	-
D-35	1,29 (694,50)	20,05 (2,95)	0,00 (421,29)	-	**A-35**	0,10 (176,04)	0,77 (3,02)	0,09 (1568,55)	0,00 (3271,18)	**S-35**	0,07 (704,96)	34,61 (2,68)	0,01 (2210,27)	0,00 (3691,26)
D-40	1,54 (2090,91)	21,62 (6,27)	0,21 (5368,66)	0,00 (25845,9)	**A-40**	1,90 (1043,7)	5,49 (7,09)	1,16 (9905,29)	0,26 (25672,51)	**S-40**	1,35 (2566,65)	33,23 (6,12)	0,26 (7757,29)	0,00 (30292,88)

Table 1 - Results from the CTSP instances used in Méndez-Dias et al (2008)

TEST	Best Gap	PQ	All -K	E-All-K
gr17	0,27	16,13 (0,03)	0,00 (5,49)	-
gr21	0,00	16,29 (0,16)	0,00 (27,32)	-
gr24	0,00	14,68 (0,30)	0,00 (99,18)	-
bays29	0,70	13,77 (0,95)	0,14 (76,10)	0,00 (721,07)
bayg29	1,87	13,10 (0,76)	0,15 (346,18)	0,00 (729,24)
rbg016a	0,00	0,98 (0,06)	0,00 (5,46)	-
rbg031a	0,00	1,32 (1,84)	0,00 (5368,66)	-
rbg050b	0,00	-	0.00 (12148.10)	-

Table 2 - Results from the CTSP instances used in Bigras et al (2008) instances

	PQ					All- k					E - All- k				
TEST	C=2	C=3	C=5	$C=\left\lceil \frac{n}{2} \right\rceil$	$C=\left\lceil \frac{n}{3} \right\rceil$	C=2	C=3	C=5	$C=\left\lceil \frac{n}{2} \right\rceil$	$C=\left\lceil \frac{n}{3} \right\rceil$	C=2	C=3	C=5	$C=\left\lceil \frac{n}{2} \right\rceil$	$C=\left\lceil \frac{n}{3} \right\rceil$
gr17	13,75 (0,05)	14,87 (0,03)	11,82 (0,03)	9,52 (0,03)	8,31 (0,03)	0,00 (5,54)	0,00 (5,38)	0,00 (5,10)	0,00 (5,12)	0,00 (5,32)	-	-	-	-	-
gr21	14,87 (0,14)	13,98 (0,12)	9,86 (0,11)	11,56 (0,11)	11,91 (0,08)	0,00 (25,71)	0,00 (26,04)	0,00 (22,87)	0,00 (28,89)	0,00 (25,77)	-	-	-	-	-
gr24	15,33 (0,27)	14,29 (0,30)	15,10 (0,23)	11,72 (0,30)	14,25 (0,27)	0,00 (90,42)	0,00 (72,40)	0,08 (129,87)	0,00 (80,06)	0,15 (95,36)	-	-	0,00 (155,21)	-	0,00 (95,36)
bays29	13,42 (0,89)	14,27 (0,81)	12,95 (0,83)	12,90 (0,64)	12,97 (0,70)	0,01 (339,66)	0,65 (335,25)	0,00 (286,95)	0,57 (431,48)	0,00 (382,03)	0,00 (661,43)	0,00 (796,56)	-	0,00 (729,18)	-
bayg29	12,50 (0,86)	*	12,07 (0,84)	10,83 (0,66)	*	0,00 (339,21)	*	0,00 (351,67)	0,24 (409,55)	*	-	*	-	0,00 (687,04)	*
rbg016a	1,13 (0,05)	0,99 (0,06)	0,79 (0,05)	0,86 (0,03)	1,28 (0,05)	0,00 (5,77)	0,00 (5,41)	0,00 (5,76)	0,00 (5,76)	0,00 (6,38)	-	-	-	-	-
rbg031a	1,40 (1,62)	1,40 (1,54)	1,28 (2,31)	0,79 (1,68)	1,28 (2,43)	0,02 (655,92)	0,15 (631,48)	0,00 (920,42)	0,00 (593,54)	0,04 (593,54)	0,00 (876,41)	0,00 (1035,16)	-	-	0,00 (593,54)

Table 3 - Results for the pure TDTSP instances.

In the next table we present the results produced by the linear programming relaxation of our model for known ATSP instances and we compare the given gaps with best known gaps from the literature. The results for the SD (see Sherali and Driscoll (2002)) and SST formulations were obtained by Sherali et al. (2006), using a Dell Workstation PWS 650 Double 2,5 GHZ CPU with the AMPL (Vs. 8.1) and CPLEX MIP Solver (V.s 9,0). The results for the well known multicommodity flow (MCF) formulation were obtained by Oncan et al (2009), using a Pentium IV 3GHZ CPU with the CPLEX BARRIER Solver (Vs. 9.0)

Test	**SD**	**SST**	**MCF**	**All –k**	**E-All k**
Ftv 33	4,8 (0,42)	0,0 (5540,85)	0,00 (17,2)	0,00 (1107,32)	-
Ftv 35	3,9 (0,34)	0,7 (19271,70)	1,06 (18,9)	0,88 (2314,35)	0,14 (10364,24)
Ftv 38	3,0 (0,78)	2,7 (8953,23)	1,02 (29,3)	0,79 (7364,74)	0,29 (22102,41)
Ftv44	2,4 (0,14)	1,1 (15910,00)	1,74 (56,6)	1,59 (18664,80)	1,09 (60335,60)
Ftv47	2.7 (2,1)	-	1,54 (232.0)	1,21 (64479,60)	0,72 (146812,00)

Table 4 - Results for ATSP instances.

The results obtained from our computational experiment show that the All-k model produces quite good gaps for the cumulative version as well as the pure time-dependent version of the ATSP. It should be stressed that for many of the instances tested, the gaps given by the linear programming relaxation of the model are equal to zero. The results also indicate that these bounds are significantly better than the ones given by the previous known models including the PQ model.

For the ATSP, the results are not as good, but the reported gaps are, as far as we know, the best gaps produced by known compact models, for the reported instances taken from TSP Lib.

In terms of solution times, the reported CPU times (including the solution times for solving the integer problem) indicate that the model may be difficult to use in practice for larger sized instances. However, the reported results suggest that the model can be used solely as a lower bounding method for larger instances.

Several alternative ways of using the proposed formulations are suggested next. First, by taking advantage of the projection result given in Proposition 5.1, the (Cut_k) inequalities, that are implied by the extended model on the z variables, can be separated in polynomial time, using a maximum-flow algorithm. In consequence, the linear programming relaxation of the projected formulation can be computed in polynomial time in the framework of a cutting plane algorithm. An extension of such algorithm in a Branch-and-Cut scheme should permit to solve larger instances. Note that a similar approach of separating cut inequalities defined in a layered graph has already been suggested by Gouveia et al. (2009) in the context of a different problem.

Second, the structure of the models allows us to suggest alternative decomposition techniques based on Lagrangean relaxation for obtaining the linear programming bound. Consider, for instance, the original All-k model and a method based on relaxing the constraints (H2P5), solving a simple problem on the z variables, n-1 enhanced n-circuit(k) subproblems on the z1 and z2 variables, and updating the multipliers by using subgradient optimization. A similar method can be devised from the reduced model. In this case, after relaxing the linking constraints, we obtain a n-circuit subproblem on the z variables and n-1 shortest path subproblems on the z1 variables.

Third, in order to try to overcome the size of the All-k model we can devise an iterative method where we start with the Single(k) model for a given k, and iteratively add different systems ($H2P1_k$)-($H2P7_k$) for different values of k. The main idea is to reach a model with a smaller size than the All-k model but with a linear programming bound that is close to the bound given by All-k. This idea follows closely the iterative method suggested in Van Vyve and Wolsey (2010) where a multicommodity flow model is used to strengthen a model based on the Miller-Tucker and Zemlin (MTZ) constraints (1960) to solve ATSP instances. That is, the authors start with a MTZ based model and iteratively add the commodities and corresponding systems of a multicommodity flow model. Note, however, that the first model in the iterative approach proposed by us is already valid for the problem while for the model given in Van Vyve and Wolsey (2010) this is true only because the MTZ constraints are already included in the model. That is, the multicommodity flow model alone is not valid unless we consider all the commodities. In both cases, a max-flow/min cut problem needs to be solved to decide how to add a new commodity. In our case, however, we need to solve the problem in a more complicated graph as specified in Section 4.

8. Conclusion

In this paper we have presented a new formulation for the Time-Dependent Travelling Salesman problem (TDTSP). The main feature of this formulation is that it uses, as a subproblem, an exact description of a slightly stronger version of the n-circuit subproblem arising on the well known Picard and Queyranne formulation. Although the new model has more variables and constraints than the original PQ model, the results given from our computational experiments show that the linear programming relaxation of the new model gives, for many of the instances tested and known from the literature, gaps that are close to zero. The results indicate that the new model is worth investigating for solving TDTSP instances, either by using it within available ILP packages or as the subject of further investigation in terms of new inequalities implied on the space of the variables of the PQ model. In a companion work we plan to provide a theoretical proof of the reported dominances for the ATSP.

References

H. Abeledo, R. Fukasawa,.A. Pessoa and E. Uchoa, “The Time Dependent Traveling Salesman Problem: Polyhedra and Branch-Cut-and-Price Algorithm” In: SEA 2010, 2010, Nápoles. Lecture Notes in Computer Science. Berlim : Springer, 2010. v. 6049, 202-213, 2010.

L. Bianco, A. Mingozzi and S. Ricciardelli, “The Traveling Salesman Problem with Cumulative Costs”, Networks, Vol 23, 81-91, 1993.

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, Vol 5, 685-699, 2008.

A. Claus “A new formulation for the travelling salesman problem” SIAM Journal on Algebraic and Discrete Methods, 5:21–5, 1984.

A. Costa, J. F. Cordeau and G. Laporte, “Models and Branch-and-Cut Algorithms for the Steiner Tree Problem with Revenues, Budget and Hop Constraints, Networks, 53, 141-159, 2009.

M. Fischetti, G. Laporte and S. Martello, “The Delivery Man Problem and Cumulative Matroids”, Operations Research, Vol 41, pp 1055, 1064, 1992.

L.R. Fox, B. Gavish and S.C. Graves ‘A N-Constrained formulation of the (Time-dependent) Traveling Salesman Problem’, Operations Research, 28,1018-1021,1980.

M. T. Godinho, L. Gouveia, and T. Magnanti, “Combined Route Capacity and Route Length Models for Unit Demand Vehicle Routing Problems”, Discrete Optimization, Vol 5, pp 350-372, 2008 (special issue in honor of George Dantzig).

M. T Godinho, L. Gouveia, T. Magnanti, P. Pesneau, and J. Pires, “On Time-Dependent Models for Unit Demand Vehicle Routing Problems”, Proceedings of the INOC2007 Conference, Spa 2007.

M. T Godinho, L. Gouveia, T. Magnanti, P. Pesneau, and J. Pires, “Formulations for the ATSP: contextualizing a strong time dependent flow based formulation (in preparation).

L. Gouveia, “Using Hop-indexed Models for Constrained Spanning and Steiner Tree Models”, in Telecommunications Network Planning, Edited by B. Samso and P. Soriano, pp 21-32, 1999

L. Gouveia and P. Pesneau, “On extended formulations for the precedence constrained asymmetric Travelling Salesman Problem”, Networks, 48, 77-89,2006.

L. Gouveia and J.M Pires, ‘Uma análise comparativa de formulações para o Problema do Caixeiro Viajante’, Investigação Operacional, 16, 89-114, 1996.

L. Gouveia and J.M. Pires (2001) The Asymmetric Travelling Salesman Problem: on Generalisations of Disaggregated Miller-Tucker-Zemlin Constraints, Discrete Applied Mathematics, 112(1-3), 129-145.

L. Gouveia, L. Simonetti and E. Uchoa, ”Modelling Hop-Constrained and Diameter-Constrained Minimum Spanning Tree Problems as Steiner Tree Problems over Layered Graphs”, to appear in Mathematical Programming, appeared online in July 2009.

L. Gouveia and S.Voss, ‘A classification of formulations for the time-dependent Travelling Salesman Problem, European Journal of Operations Research, 83, 69-82,1995

A. Langevin, F. Soumis and J. Desrosiers , “Classification of travelling salesman formulations”, Operations Research Letters, 9, 127-132, 1990.

E. Lawler, J. Lenstra, A. Rinnooy Kan, and D. Shmoys, “The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization,” WiLey, New York, 1985.

C. Miller, A. Tucker and R. Zemlin, "Integer programming formulations and traveling salesman problems", Journal of ACM, 7, 326-329, 1960.

A. Lucena, “The Time-Dependent Traveling Salesman Problem – The Deliveryman Case”, Networks, 20, 753-763, 1990.

I. Méndez-Díaz, P. Zabala and A. Lucena, “A New Formulation for the Traveling Deliveryman Problem”, Discrete Applied Mathematics, 156, 3223-3237, 2008.

T. Oncan, I. K. Atinel and G. Laporte, “A Comparative analysis of several asymmetric traveling salesman problem formulations”, Computers and Operations Research, 36(3), 637-654, 2009.

J.C. Picard and M. Queyranne, ‘The Time-dependent Traveling salesman problem and its application to the tardiness in one-machine scheduling’, Operations Research 26, 86–110 1978.

S. C. Sarin, H. D. Sherali and A. Bhootra, “New Tighter Polynomial Length Formulations for the Asymmetric Travelling Salesman Problem with and without Precedence Constraints”, Operations Research Letters, 33, 62–70, 2005.

H. D. Sherali and P. J. Driscoll, “On Tightening the Relaxations of Miller-Tucker-Zemlin Formulations for Asymmetric Traveling Salesman Problems”, Operations Research, 50, 656-669, 2002.

H. D. Sherali, S. C. Sarin, and P. F. Tsai, “A class of lifted path and flow-based formulations for the asymmetric traveling salesman problem with and without precedence constraints”, Discrete Optimization, 3, 20–32, 2006.

M. Van Vyve and L. A. Wolsey: Approximate extended formulations, Math. Program., Ser. B 105, 501-522, 2006

R. T. Wong “Integer programming formulations of the traveling salesman problem”. Proceedings of the IEEE international conference of circuits and computers, 149–52, 1980..

Appendix

In this appendix we show how to eliminate the entire set of the z2 variables and obtain a model with fewer variables and with a linear programming relaxation equivalent to the original All(k) model. Consider the All-k model rewritten explicitly as follows:

$$
\begin{array}{lll}
\textit{minimize} \quad \sum_{(i,j)\in A}\sum_{h=1,\dots,n} c_{ij}^{h} z_{ij}^{h} & & \\
\sum_{h=2,\dots,n}\sum_{i\in V\setminus\{1\}} z_{ij}^{h} = 1 & \textit{for all } j \in V\setminus\{1\} & (PQ1) \\
\sum_{j\in V\setminus\{1\}} z1_{1j}^{1k} = 1 & \textit{for all } k \in V\setminus\{1\} & (H2P1) \\
\sum_{j\in V\setminus\{1\}} z1_{ij}^{2k} - z1_{1i}^{1k} = 0 & \textit{for all } i,k \in V\setminus\{1\}, i \neq k & (H2P2^{1}) \\
\sum_{j\in V\setminus\{1\}} z1_{ij}^{h+1,k} - \sum_{j\in V\setminus\{1\};\, j\neq k} z1_{ji}^{hk} = 0 & \textit{for all } i,k \in V\setminus\{1\}, i \neq k, h = 2,..,n-2 & (H2P2^{h}) \\
\sum_{j\in V\setminus\{1\};\, j\neq k} z2_{kj}^{2k} - z1_{1k}^{1k} = 0 & & (H2P3^{1}) \\
\sum_{j\in V\setminus\{1\};\, j\neq k} z2_{kj}^{h+1,k} - \sum_{j\in V\setminus\{1\};\, j\neq k} z1_{jk}^{hk} = 0 & h = 2,\dots,n-2 & (H2P3^{h}) \\
z2_{k1}^{nk} - \sum_{j\in V\setminus\{1\};\, j\neq k} z1_{jk}^{n-1,k} = 0 & & (H2P3^{n-1}) \\
\sum_{j\in V\setminus\{1\};\, j\neq k} z2_{ij}^{h+1,k} - \sum_{j\in V\setminus\{1\}} z2_{ji}^{hk} = 0 & \textit{for all } i,k \in V\setminus\{1\}, i \neq k, h = 2,..,n-2 & (H2P4^{h}) \\
z2_{i1}^{nk} - \sum_{j\in V\setminus\{1\}} z2_{ji}^{n-1k} = 0 & \textit{for all } i,k \in V\setminus\{1\}, i \neq k & (H2P4^{n-1}) \\
z_{ij}^{h} = z1_{ij}^{hk} + z2_{ij}^{hk} & \textit{for all } (i,j) \in A, h = 2,\dots,n-1 & (H2P5) \\
z1_{ij}^{hk} \in \{0,1\} & \textit{for all } (i,j) \in A, i \neq k, j \neq 1, h = 1,..,n-1 & (H2P6) \\
z2_{ij}^{hk} \in \{0,1\} & \textit{for all } (i,j) \in A, i \neq 1, j \neq k, h = 2,..,n & (H2P7) \\
z_{ij}^{h} \in \{0,1\} & \textit{for all } (i,j) \in A, h = 1,..,n & (PQ4)
\end{array}
$$

As we noted in section 4 the linking constraints (H2P5) permit us to obtain the following linking constraints

$$z_{ij}^{h} = z1_{ij}^{hj} = z1_{ij}^{hk} + z2_{ij}^{hk} = z2_{ij}^{hi} \qquad \textit{for all } i,j,k \in V\setminus\{1,i,j\};\, h = 3,..,n-2 \qquad \text{(A)}$$

$$z_{1j}^{1} = z1_{1j}^{1k} \qquad \textit{for all } i,j,k \in V\setminus\{1,j\} \qquad \text{(B)}$$

$$z_{ij}^{2} = z1_{ij}^{2k} \qquad \textit{for all } i,j,k \in V\setminus\{1,i\} \qquad \text{(C)}$$

$$z_{ij}^{n-1} = z2_{ij}^{n-1,k} \qquad \textit{for all } i,j,k \in V \setminus \{1,j\} \qquad \text{(D)}$$

$$z_{i1}^{n} = z2_{i1}^{nk} \qquad \textit{for all } (i,1) \in A; \qquad \text{(E)}$$

We show next how to use equalities (A) to (E) to eliminate the entire set of the z2 variables and obtain a model with fewer variables and with a linear programming relaxation equivalent to the original All(k) model. This transformation is described next.

i) Constraints (H2P1):

Constraints (H2P1) and (H2P2) remain unchanged.

iii) Constraints ($H2P3_k$):

Now, note that by (A) we obtain $z_{ij}^{h} = z1_{ij}^{hj} = z2_{ij}^{hi}$ for all $(i,j) \in A, i,j \neq 1$ and $h = 2,..,n-1$. Thus, by doing a simple substitution, the inequalities (H2P3) for h=1,…,n-2 become

$$\sum_{i \in V} z_{ik}^{h} - \sum_{i \in V} z1_{ki}^{h+1} = 0 \qquad h=1,..,n-2,\ k \in V \setminus \{1\} \qquad \text{(PQ3)}$$

iii) Constraints (H2P4):

We show next that the inequalities that are obtained by using (A)-(E) in (H2P4) lead to redundant inequalities in the enhanced model.

Consider constraints (H2P4) for a given i,k such that $i,k \in V \setminus \{1\}$, $i \neq k$

$$\sum_{j \in V} z2_{ij}^{h+1,k} - z2_{ki}^{hk} - \sum_{j \in V-\{k\}} z2_{ji}^{hk} = 0 \qquad \textit{for } h = 2,..,n-2 \qquad (*)$$

For $2 \leq h \leq n-3$ we use (A) in (*) to obtain:

$$\sum_{j \in V} (z_{ij}^{h+1} - z1_{ij}^{h+1,k}) - z_{ki}^{h} - \sum_{j \in V-\{k\}} (z_{ij}^{h} - z1_{ji}^{hk}) = 0$$

Rearranging both hand sides, we obtain finally:

$$\sum_{j \in V} z_{ij}^{h+1} - \sum_{j \in V} z_{ij}^{h} = \sum_{j \in V} z1_{ij}^{h+1,k} - \sum_{j \in V} z1_{ij}^{hk}$$

that can easily be seen to be redundant in the presence of (H2P2) and (PQ3).

For h = n-2 we use (D) and (A) in (*) to obtain:

$$\sum_{j\in V-\{k\}} z_{ij}^{n-1} - \sum_{j\in V-\{k\}} z2_{ji}^{n-2,k} + z2_{ki}^{n-2,k} = 0$$

Rearranging, we obtain:

$$\sum_{j\in V-\{k\}} z_{ij}^{n-1} - \sum_{j\in V} z_{ij}^{n-2} = \sum_{j\in V} z1_{ji}^{n-2,k}$$

Finally, adding z_{ik}^{n-1} to both hand sides of the equation, we obtain

$$\sum_{j\in V} z_{ij}^{n-1} - \sum_{j\in V} z_{ij}^{n-2} = z_{ik}^{n-1} - \sum_{j\in V} z1_{ji}^{n-2,k}$$

which can also be seen to be redundant in the presence of (H2P2) and (PQ3).

As for h = n-1 variables $z2_{ij}^{nk}$ are not defined for $j \neq 1$, we can write (H2P4) as follows:

$$z2_{i1}^{nk} - \sum_{j\in V} z2_{ji}^{n-1,k} = 0$$

By using (E) and (D) in the previous inequality we obtain:

$$z_{i1}^{n} - \sum_{j\in V} z_{ji}^{n-1} = 0$$

which is one of the (PQ3) equalities and, therefore, redundant in the reduced model.

Thus, we have shown that the inequalities that are obtained by using (A)-(E) in (H2P4) lead to redundant inequalities in the reduced model.

iv) Linking Inequalities (H2P5):

By using (A) together with $z2_{ij}^{hk} \geq 0$ for h=3,..,n-2 we obtain the projected inequalities

$z_{ij}^{h} \geq z1_{ij}^{hk}$ $\quad$ $(i,j) \in$ A,i,j≠1;k=2,..,n; k≠i,j; h=3,..,n-2 $\quad$ (H2P10)

By using (A) together with $z2_{ij}^{hk} \leq 1$ we obtain the inequalities

$z_{ij}^{h} \leq z1_{ij}^{hk} + 1$ $\quad$ $(i,j) \in$ A,i,j≠1;k=2,..,n;k≠i,j; h=3,..,n-2

However, these inequalities are redundant in the presence of (PQ1), and thus we omit them from the reduced model.

This terminates the determination of the reduced model. We could go further by eliminating some of the z1 variables by using (B) and (C). However, for the sake of simplicity and to have some intuition on the new model we kept them in the model.