On a Time-Dependent Formulation and an Updated Classification of ATSP Formulations

Maria Teresa Godinho, Luís Gouveia, Pierre Pesneau

On a Time-Depedent Formulation and an updated classification of ATSP formulations

M. T. Godinho[a*] **L. Gouveia**[b†] **P. 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 chapter we contextualize, in terms of the ATSP, a recent compact formulation presented in Godinho et al [11] for the time-dependent traveling salesman problem (TDTSP). The previous paper provides one way of viewing the new model, "the time-dependent TSP point of view" where it is put in evidence how to obtain the new model by tightening the linear programming relaxation of a well known formulation for the TDTSP. In this chapter, we will present the ATSP point of view and will show how to obtain the model by i) enhancing the subproblem arising in the standard multicommodity flow (MCF) model for the ATSP and then ii) by using modelling enhancement techniques. We will compare the linear programming relaxation of the new formulation with the linear programming relaxation of the three compact and non-dominated formulations presented in Oncan et al. [19]. As a result of this comparison we present an updated classification of formulations for the asymmetric traveling salesman problem (ATSP).

1 Introduction

In this chapter we present an updated classification of formulations for the asymmetric traveling salesman problem (ATSP). In the past, several papers have produced a classification of formulations for the ATSP, in terms of the associated linear programming relaxations. Among others, we may consider the papers by Gouveia and Voss [15], Langevin et al [16], Gouveia and Pires [13], Orman and Williams [20] and Oncan et al [19]. These papers fall among two classes. Either they produce new results between formulations known from the literature, or they use the fact that new formulations are also being presented in the paper in order to upgrade a classification already known from the literature. Our chapter falls in the second category in the sense that we are going to contextualize, in terms of the ATSP, a recent compact formulation presented in Godinho et al [11] for the time-dependent traveling salesman problem (TDTSP).

Let $G = (V, A)$ be a graph where $V = \{1, 2, .., n\}$ and $A = \{(i, j) : i, j = 1, .., n, i \neq j\}$. In the ATSP we consider costs c_{ij} associated to each arc (i, j) and we want to find a minimum cost Hamiltonian circuit, starting and ending on node 1. In the TDTSP, we also want to find a minimum

*Supported by FCT research project [MATH-LVT-Lisboa-152]
†Supported by FCT research project [MATH-LVT-Lisboa-152]
‡Supported by the ANR project number ANR-09-blan-0373-01

cost Hamiltonian circuit, starting and ending on node 1. However, the arc costs depend on their 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, formulations for the TDTSP can be used for the ATSP by simply considering $c_{ij}^h = c_{ij}$ for every arc (i,j) and every position h.

The paper by Godinho et al [11] provides one way of viewing the new model. We call it "the time-dependent TSP point of view" where it is put in evidence how to obtain the new model by tightening the linear programming relaxation of the Picard and Queyranne [21] formulation. In this chapter, we will present the ATSP point of view and will show how to obtain the model by enhancing the subproblem arising in the standard multicommodity flow (MCF) model for the ATSP (see Wong [27] and Claus [2]). These enhancements lead to a model with a linear programming relaxation that is tighter than the one produced by the MCF model but still weaker than the one given in Godinho et al. [11]. Then, by using variable redefinition and constraint disaggregation we make a bridge to the TDTSP, enhance the linear programming relaxation of the model and obtain the model of Godinho et al [11]. Finally, we present further enhancements on the model and show that these enhancements have a natural interpretation in terms of constraints that arise in a different class of extended formulations for the ATSP, that is, formulations with precedence variables.

We will compare the linear programming relaxation of the new formulation with the linear programming relaxation of the three compact and non-dominated formulations presented in Oncan et al. [19]. The three formulations are the one with the best linear programming relaxation presented in Fox et al. [7], the one with the best linear programming relaxation presented in Sherali and Driscoll [25] as well as the one with the best linear programming relaxation presented in Sherali et al. [26] which we denote respectively by FGG, SD and SST.

Gouveia and Voss [15] have shown that the linear programming relaxation of the Picard and Queyranne [21] formulation, denoted by PQ in the remaining of the chapter, is known to strictly dominate the linear programming relaxation of the FGG formulation. Thus, we will show that the linear programming relaxation of the new model is at least as good as the linear programming relaxation of the PQ formulation. This is the topic of Section 3. Section 4 makes a short introduction to the SD model and provides a proof that the linear programming relaxation of the new model is at least as good. In these two cases, the new model provides much better bounds for many of the instances tested. We will show that there is no dominance relationship between the linear programming relaxations of the new model and the model SST. However, we will also show how to slightly enhance the new model in order to obtain a theoretical dominance. This comparison will be discussed in Section 5. Section 6 concludes the chapter and presents numerical results on the linear programming bounds from the different models for instances coming from the TSP-lib.

2 Flow based formulations

In this section we show how to develop the formulation described in Godinho et al. [11] from an ATSP point of view. In Section 2.1 we review a standard framework for presenting formulations for the ATSP. In the remaining sections we describe the several enhancement steps to obtain the new formulation from the multicommodity flow model.

2.1 A Generic Model

Consider the binary variables x_{ij} indicating whether arc (i,j) is in the solution and the following *generic formulation* for the ATSP:

$$\text{minimize} \sum_{(i,j)\in A} c_{ij}x_{ij}$$

$$\text{subject to } x = (x_{ij}) \in \text{Assign} \qquad (1)$$

$$\{(i,j) : x_{ij} = 1\} \text{ is connected} \qquad (2)$$

with Assign denoting the feasible set of the assignment relaxation:

$$\sum_{i \in V} x_{ij} = 1 \quad \text{for all } j \in V \tag{3}$$

$$\sum_{j \in V} x_{ij} = 1 \quad \text{for all } i \in V \tag{4}$$

$$x_{ij} \in \{0,1\} \quad \text{for all } (i,j) \in A. \tag{5}$$

Many works on formulations for the ATSP start with this generic formulation and then describe different ways of expressing (2). The derived formulations may be classified into two classes: natural formulations and extended formulations. In the first class, we include formulations that use only the design variables x_{ij} and in the second class we include formulations that use linear inequalities on an extra set of variables to model the connectivity requirements. This second class may also be divided into sub-classes, according to the kind of extra variables used. In fact, in the next subsections we will review and introduce network flow based models based on paths, circuits and n-circuits, time-dependent network flow based models and we will establish a connection between these models with models using precedence variables.

Note that the definition of the variables in these models can lead to particular cases for the constraints where some terms do not appear (because they correspond to variables that are not defined). For each formulation, we will precise which variables are defined or not.

In the following we denote the linear programming relaxation of a given model P by P_L and its linear programming bound by $v(\mathrm{P}_L)$. We will use the designation "exact" model for a model whose linear programming relaxation only has integral extreme points. We let $F(\mathrm{P})$ denote the set of feasible solutions of an integer (linear) program P.

2.2 Path-based formulations

Constraints (2) can be modelled by using extra variables to express the connectivity requirements. For each node $k \in V \setminus \{1\}$, we define path variables y_{ij}^k ($(i,j) \in A$, $j \neq i$ and $i \neq k$) that indicate whether arc (i,j) is in the path from node 1 to node k. Let path(k) denote a path from node 1 to node k. Then, we can rewrite the non-explicit part (2) of the generic formulation as follows:

$$\{(i,j) : y_{ij}^k = 1\} \text{ contains a path}(k) \quad \text{for all } k \in V \setminus \{1\}, \tag{6}$$

$$y_{ij}^k \leq x_{ij} \quad \text{for all } (i,j) \in A, j \neq 1,\ k \in V \setminus \{1\}, i \neq k. \tag{7}$$

We can represent the previous condition (6) on the variables y_{ij}^k as follows:

$$\sum_{j \in V} y_{ij}^k - \sum_{j \in V} y_{ji}^k = \begin{cases} 1 & i = 1 \\ 0 & i \neq 1, k \\ -1 & i = k \end{cases} \quad \text{for all } k \in V \setminus \{1\} \tag{8}$$

$$y_{ij}^k \in \{0,1\} \quad \text{for all } (i,j) \in A, j \neq 1, \text{ for all } k \in V \setminus \{1\}, i \neq k. \tag{9}$$

By using this flow/path model together with (7) for replacing (2) in the generic formulation, we obtain the standard multi-commodity flow formulation for the ATSP (see Claus [2]), which will be denoted by P-MCF in this chapter. One important remark for the proofs that will appear later on is that for the linear programming relaxation of the P-MCF model, constraints (7) are satisfied as equalities when $k = j$ or when $i = 1$ (this follows easily from the constraints (8)-(9) together with the assignment constraints on the x_{ij} variables).

By using the max-flow min-cut theorem it is easy to show that the projection of the feasible set of the linear programming relaxation of the MCF model into the subset of the x_{ij} variables is completely described by non-negativity variables on the x_{ij} variables, (1) and the cut constraints

$$\sum_{[i \in V \setminus S, j \in S]} x_{ij} \geq 1 \quad \text{for all } S \subseteq V \setminus \{1\} \tag{10}$$

Thus, if we use the constraints (10) to express (2) we obtain a natural formulation for the ATSP that, based on the previous projection result, provides the same linear programming bound as the P-MCF model.

2.3 Circuit-based formulations

One way of attempting to improve the linear programming relaxation of the models described in subsection 2.2 is to replace the concept of "path(k)" by the concept of "circuit(k)". Let circuit(k) denote a circuit starting and ending at node 1 and passing through node k only once. Consider the circuit variables g^k_{ij} that indicate whether arc (i,j) is in a circuit(k). In contrast to the P-MCF model, because the circuit passing through node k does not terminate at that node, the new models contain both the variables g^k_{kj} and g^k_{i1}. Furthermore, since we are modeling a circuit, we can use equalities $g^k_{ij} = x_{ij}$ to relate the g^k_{ij} variables with the x_{ij} variables.

Consider the generic formulation given as follows:

$$\{(i,j) : g^k_{ij} = 1\} \text{ contains a circuit}(k) \quad \text{for all } k \in V \setminus \{1\}, \tag{11}$$

$$g^k_{ij} = x_{ij} \quad \text{for all } (i,j) \in A,\ k \in V \setminus \{1\}. \tag{12}$$

It does not seem easy to model the non-explicit part (11) of the formulation with only the g^k_{ij} variables without allowing circuits disconnected from the depot. One alternative is to follow Wong [27] and decompose the circuit subproblem into two path subproblems, for each $k \in V \setminus \{1\}$. Then, instead of using one set of circuit variables, we could use two sets of path variables yt^k_{ij} $(t = 1, 2)$, one associated with the path from node 1 to node k and the other associated with the path from node k to node 1. The variables in the first set, $y1^k_{ij}$ $(i \neq k$ and $j \neq 1)$, have the same interpretation as the variables in the previous set y^k_{ij} while the variables in the second set $y2^k_{ij}$ $(j \neq k$ and $i \neq 1)$ indicate whether arc (i,j) is in the path from node k to node 1. With these two sets of variables we can rewrite the non explicit part (11) as follows:

$$\sum_{j \in V} y1^k_{ij} - \sum_{j \in V} y1^k_{ji} = \begin{cases} 1 & i = 1 \\ 0 & i \neq 1, k \\ -1 & i = k \end{cases} \qquad \sum_{j \in V} y2^k_{ij} - \sum_{j \in V} y2^k_{ji} = \begin{cases} 1 & i = k \\ 0 & i \neq 1, k \\ -1 & i = 1 \end{cases} \tag{13}$$

$$\text{for all } k \in V \setminus \{1\}$$

$$y1^k_{ij} + y2^k_{ij} = g^k_{ij} \quad \text{for all } (i,j) \in A,\ k \in V \setminus \{1\} \tag{14}$$

$$yt^k_{ij} \in \{0,1\} \quad \text{for all } (i,j) \in A, k \in V \setminus \{1\},\ t = 1,2. \tag{15}$$

For each $k \in V \setminus \{1\}$, the constraints on the left-hand side ensure that there is a path from node 1 to node k. The constraints on the right-hand side ensure that there is a path from node k to node 1. Note also that the way the variables are defined guarantees that the cycle does not visit node k more than once. The coupling constraints $y1^k_{ij} + y2^k_{ij} = g^k_{ij}$ guarantee that, for each k, an arc is used in the circuit passing through node k if and only if it is in the path fron node 1 to node k or in the path from node k to node 1. Note however, that feasible solutions to the system (13)-(15) allow circuits disconnected from the depot. These subcircuits will be not allowed when we consider all the different circuit(k) models together with the constraints involving the x_{ij} variables.

We can obtain a formulation for the ATSP by using the previous system in the place of (11). We denote by 2P-MCF the formulation just defined. We note that by using (12) and the coupling constraints from the model above, we can eliminate the g_{ij}^k variables from the model. This formulation is exactly the formulation presented by Loulou [17] which, as shown in Langevin et al [16] is also equivalent to the P-MCF formulation and other versions of the 2P-MCF model including the original version proposed by Wong [27].

These results indicate that we have not gained anything by replacing the path subproblem by a circuit subproblem. In the next subsection we introduce models using the stronger subproblem where we add the condition "the circuit has n arcs" to circuit(k). Before, describing models based on this subproblem and as a sort of introduction to the next section, we consider the effect of adding the cardinality constraint

$$\sum_{(i,j)\in A} y1_{ij}^k + \sum_{(i,j)\in A} y2_{ij}^k = n \tag{16}$$

to the model (13)-(15). The constraint states that the number of arcs used in any feasible solution to the model (13)-(15) is equal to n. However, we observe that the inclusion of this constraint does not imply that the circuit starting and ending in node 1 has n arcs since any feasible solution to the model (13)-(15) may have circuits disconnected from node 1. Thus, what the constraint guarantees is that the total number of arcs included in the circuit passing through node 1 and node k and eventually in any other disconnected circuits that may exist in the solution is equal to n.

A different observation is that the model (13)-(15) is exact. This property is lost when the cardinality constraint (16) is added to the model.

A final observation is that although the cardinality equality (16) tightens the set of feasible solutions of the model (13)-(15), it does not tighten the linear programming relaxation of the whole 2P-MCF model. In fact, we can easily see that if for a fixed j and a fixed commodity k, we add the linking constraints $y1_{ij}^k + y2_{ij}^k = x_{ij}$ for $i = 1, \ldots, n$ and use the assignment constraints (3) for node j, we obtain $\sum_{i=1,\ldots,n} y1_{ij}^k + \sum_{i=1,\ldots,n} y2_{ij}^k = 1$ for the same j and k. Then, by adding the previous equality for all $j = 1, \ldots, n$, we obtain the equality (16).

2.4 n-Circuit-based formulations

Godinho et al. [10] have suggested a formulation for the ATSP where the subproblem associated to each node k is a circuit(k) with exactly n arcs (which we will denote by n-circuit(k)). Consider the circuit variables g_{ij}^k that indicate whether arc (i,j) is in a n-circuit(k) (for simplicity we use the same variables as in the model described in the previous subsection).

Consider the generic formulation given as follows

$$\{(i,j) : g_{ij}^k = 1\} \text{ is a } n\text{-circuit}(k) \quad \text{for all } k \in V \setminus \{1\}, \tag{17}$$

$$g_{ij}^k = x_{ij} \quad \text{for all } (i,j) \in A,\ k \in V \setminus \{1\}. \tag{18}$$

As before, it does not seem easy to model the non-explicit part (17) of the formulation with only the g_{ij}^k variables. However, this subproblem can be solved as an unconstrained shortest path in appropriate graphs, as it has been done in Godinho et al. [11] and previously by Godinho et al. [10]. The usual path equations for the unconstrained shortest path problem rewritten in these graphs give valid formulations for the n-circuit(k) subproblem. Since the graph used in Godinho et al [11] is more compact than the graph used in the previous Godinho et al [10] paper, it leads to a more compact formulation and for this reason we will describe it here and omit the previous formulation. The Godinho et al. [11] formulation 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 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

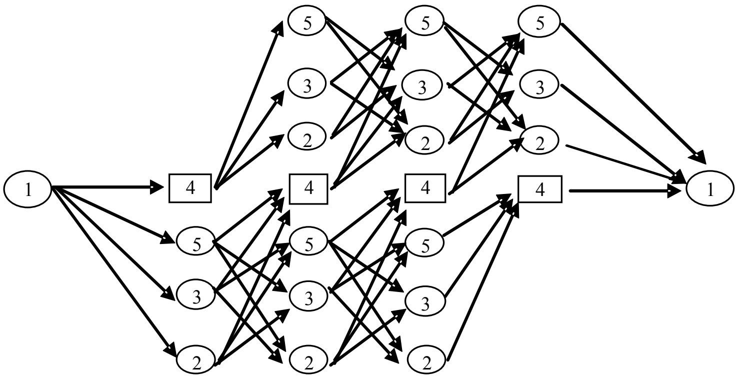

Figure 1: A two-layered graph modeling a 5-circuit(4).

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.

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 variables $z1_{ij}^{hk}$ (resp. binary variables $z2_{ij}^{hk}$) to the arcs of the subgraph 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 and is before node k. They are defined only on the following three cases:

1. $h = 1$, $i = 1$ and $j \in V \setminus \{1\}$,
2. $h = 2, \ldots, n-2$, $i \in V \setminus \{1, k\}$ and $j \in V \setminus \{1\}$, $i \neq j$,
3. $h = n-1$, $i \in V \setminus \{1, k\}$, and $j = k$.

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 and is after node k. They are defined only on the following cases:

1. $h = 2$, $i = k$ and $j \in V \setminus \{1, k\}$,
2. $h = 3, \ldots, n-1$, $i \in V \setminus \{1\}$ and $j \in V \setminus \{1, k\}$, $i \neq j$,
3. $h = n$, $i \in V \setminus \{1\}$, and $j = 1$.

Using these variables, we can write the following new model for the n-circuit(k) subproblem (17) for a given k. As mentioned at the end of the introduction, 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).

$$\sum_{j \in V \setminus \{1\}} z1_{1j}^{1k} \tag{19}$$

$$\sum_{j \in V \setminus \{1\}} z1_{ij}^{h+1,k} - \sum_{j \in V;\, j \neq k} z1_{ji}^{hk} = 0 \quad \text{for all } i \in V \setminus \{1\}, i \neq k, h = 1, \ldots, n-2 \tag{20}$$

$$\sum_{j \in V;\, j \neq k} z2_{kj}^{h+1,k} - \sum_{j \in V;\, j \neq k} z1_{jk}^{hk} = 0 \quad \text{for } h = 1, \ldots, n-1 \tag{21}$$

$$\sum_{j \in V;\, j \neq k} z2_{ij}^{h+1,k} - \sum_{j \in V \setminus \{1\}} z2_{ji}^{hk} = 0 \quad \text{for all } i \in V \setminus \{1\}, i \neq k, h = 2, \ldots, n-1 \tag{22}$$

$$g_{ij}^{k} = \sum_{h=1,\ldots,n} (z1_{ij}^{hk} + z2_{ij}^{hk}) \quad \text{for all } (i,j) \in A \tag{23}$$

$$z1_{ij}^{hk} \in \{0,1\} \quad \text{for all } (i,j) \in A, i \neq k, j \neq 1, h = 1, \ldots, n-1 \tag{24}$$

$$z2_{ij}^{hk} \in \{0,1\} \quad \text{for all } (i,j) \in A, i \neq 1, j \neq k, h = 2, \ldots, n \tag{25}$$

$$g_{ij}^{k} \in \{0,1\} \quad \text{for all } (i,j) \in A \tag{26}$$

We can obtain a formulation for the ATSP by replacing (17) with this circuit formulation for each k. We let C-MCF denote this model. We note that when creating this model, constraints (23) enable us to rewrite the linking constraints $g_{ij}^{k} = x_{ij}$ using only the $z1_{ij}^{hk}$, $z2_{ij}^{hk}$ and x_{ij} variables, and so we can eliminate the g_{ij}^{k} variables (and constraints (23)) from the model. The transformed linking constraints become as follows:

$$\sum_{h=1,\ldots,n} (z1_{ij}^{hk} + z2_{ij}^{hk}) = x_{ij} \quad \text{for all } (i,j) \in A, \text{for all } k \in V \setminus \{1\}.$$

For the sake of simplicity, we maintain the designation (23) for these transformed equalities. Since the model (19)-(26) in the g_{ij}^{k}, $z1_{ij}^{hk}$ and $z2_{ij}^{hk}$ variables defines the path equations on the expanded network, its corresponding matrix is totally unimodular and we can conclude that its linear programming relaxation is integer. Since (19)-(26) is integral and models a toughter subproblem than the subproblem modeled by (13)-(15), we have

Propositon 2.1 *$v(\text{C-MCF}_L) \geq v(\text{2P-MCF}_L)$.*

We will show in Section 6 that the inequality can be strict for many instances.

Since $v(\text{2P-MCF}_L) = v(\text{P-MCF}_L)$ this dominance also holds with respect to the original P-MCF model and equivalent formulations such as the one obtained by using the constraints (10) in place of (2).

We observe that this model produces the same linear programming bound of a model presented in Godinho et al. [10]. The equivalence simply follows from the fact that the latter model also uses an exact, although less compact, model for the the n-circuit(k) subproblem (17).

Proposition 2.1 motivates the question of knowing which inequalities are implied by the linear programming relaxation of the C-MCF model in the space of the x_{ij} variables. Clearly, constraints (10) are included in the class of projected inequalities. We can use the previously mentioned equivalence to a model in Godinho et al. [10] to give a partial answer to this question since some such new inequalities are already described in this paper. These inequalities, called double-jumps, are related to the jump inequalities introduced by Dahl [4] and correspond to a weaker version of the lifted cycle inequalities proposed by Balas and Fischetti [1] which are known to be facet defining for the ATSP polytope. However, finding out what the remaining inequalities are seems to be much more difficult. The reason for this is the fact that hop-constrained path subproblems are included in the whole model and it is well known that finding the projection in the natural space of related polyhedra is not easy (see, for instance the paper by Dahl et al. [5].)

2.5 Enhancing the C-MCF model (using time-dependent variables)

The model described in the previous section can be enhanced by using variable redefinition and constraint disaggregation. Furthermore, with the idea suggested in this section we will make a bridge between the resulting enhanced formulation with time-dependent formulations for the ATSP (see next section).

Consider the binary variables z_{ij}^h for all $(i,j) \in A$ and $h = 1, \ldots, n$, indicating whether or not arc $(i,j) \in A$ is in the h^{th} position of the circuit. Note that when defining these variables we have $i = 1$ iff $h = 1$ (the arc leaving node 1 is in position 1) and $j = 1$ iff $h = n$ (the arc entering the depot is in position n). The original arc variables and the time-dependent variables are related as follows:

$$\sum_{h=1,\ldots,n} z_{ij}^h = x_{ij} \quad \text{for all } (i,j) \in A \tag{27}$$

Note that as before, we have special cases for these equalities depending on whether $i = 1$, $j = 1$ or $i, j \neq 1$. The interest of using these variables arises when we use the previous equality to rewrite (23) as follows

$$\sum_{h=1,\ldots,n} (z1_{ij}^{hk} + z2_{ij}^{hk}) = \sum_{h=1,\ldots,n} z_{ij}^h \quad \text{for all } (i,j) \in A.$$

Until, now, we have not tightened the previous formulation. However, the previous equalities can be disaggregated into

$$z1_{ij}^{hk} + z2_{ij}^{hk} = z_{ij}^h \quad \text{for all } (i,j) \in A, h = 1, \ldots, n. \tag{28}$$

We denote by EC-MCF the model obtained from the model C-MCF by adding the constraints (27) and by replacing (23) with (28) for all positions h. Clearly, we have that

Propositon 2.2 $v(\text{EC-MCF}_L) \geq v(\text{C-MCF}_L)$.

We will show in Section 6 that the inequality can be strict for many instances (in fact, the improvements are substantial in most cases).

We can give an intuitive and different interpretation to the disaggregated equalities (28). In the previous subsections the emphasis on modelling was set on finding good formulations for the sub-problems associated to each node k, each one seen as an independent subproblem - the subproblems are related to each other only through the linking constraints. However, when we analyse a solution for the ATSP we realize that there is a great deal of information regarding the way the sub-problems are related to each other and that might be used to derive improved formulations. In fact, what the disaggregated constraints are saying is that one arc (i,j) is the h^{th} arc of the circuit associated to a given node k if and only if it is in the same position in the other $n-2$ circuits associated to nodes $p \in V \setminus \{1\}$, $p \neq k$.

2.6 Enhancing the C-MCF model (using precedence variables)

The enhancements described in this subsection can be viewed independently of the enhancements described in the previous one. Thus, they can be used either to strengthen the C-MCF model or even to strengthen the stronger EC-MCF model.

We start by presenting a new set of equalities that relate the two sets of variables $z1$ and $z2$. Then, we will show that these equalities have a natural interpretation in terms of constraints that arise in a different class of extended formulations for the ATSP. Consider, the following equalities

$$\sum_{h=1,\ldots,n}\sum_{p\in V;\,p\neq k} z1_{pj}^{hk} = \sum_{h=1,\ldots,n}\sum_{p\in V;\,p\neq k} z2_{kp}^{hj} \quad \text{for all } j,k=2,\ldots,n, k\neq j. \tag{29}$$

The validity of these equalities is easily explained by looking at Figure 1. What the equality states, for a given pair of nodes j and k, is that node j is in the first part of the circuit through node k if and only if node k is in the second part of the circuit through node j. In fact this equality illustrates an idea explored in the previous subsection where information between different commodities is explored.

We present next an alternative motivation for these equalities. The motivation will be given in terms of extended formulations for the ATSP using binary precedence variables v_j^k indicating whether j is before k in the tour. As far as we know, the first such model was proposed by Claus [2] leading to an enhanced version of the P-MCF model (see also Section 5). Later on, Gouveia and Pires [13] present several models which can be viewed as a disaggregation (and tightening) of the well known Miller-Tucker-Zemlin [18] model. Formulations using these variables were further explored in Gouveia and Pires [14], Sarin et al. [22], Sherali et al [26], and Gouveia and Pesneau [12]. Oncan et al. [19] compare the linear programming relaxation of most of these formulations.

It is easy to see that the precedence variables can be related to the variables used in our models either as

$$v_j^k = \sum_{h=1,\ldots,n-2}\sum_{p\in V;\,p\neq k} z1_{pj}^{hk} \quad \text{for all } j,k\in V\setminus\{1\}, k\neq j \tag{30}$$

or as

$$v_j^k = \sum_{h=2,\ldots,n-1}\sum_{p\in V;\,p\neq k} z2_{kp}^{hj} \quad \text{for all } j,k\in V\setminus\{1\}, k\neq j \tag{31}$$

Note that the equalities (29) are obtained by considering (30) together with (31). Let us assume, now, that we will use only (30) to relate the two sets of variables. We will show next that adding (31) (or equivalently, adding the equalities (29)) is equivalent to adding a well known set of equalities defined on the v_j^k variables, namely the equalities

$$v_j^k + v_k^j = 1 \quad \text{for all } j,k\in V\setminus\{1\} \tag{32}$$

which state that for any given pair of nodes, one is before the other in the tour. These equalities were first used in the paper by Claus [2] and are also known from model of the linear ordering problem. As noted in the previous papers, the inclusion of these equalities lead to reasonable improvements on the lower bounds produced by most of the models using this type of variables.

To prove this equivalence (that is, that adding (31) to (30) is equivalent to adding the equalities (32) to (30)) we consider the equalities (23)

$$\sum_{h=1,\ldots,n} (z1_{ij}^{hk} + z2_{ij}^{hk}) = x_{ij} \quad \text{for all } (i,j)\in A, \text{ for all } k\in V\setminus\{1\}$$

from the C-MCF model. If we fix nodes j and k and sum up for all i we obtain (the last equality follows from the assignment constraint (3) for node j)

$$\sum_{i\in V}\sum_{h=1,\ldots,n} (z1_{ij}^{hk} + z2_{ij}^{hk}) = \sum_{i\in V} x_{ij} = 1 \quad \text{for all } j,k\in V\setminus\{1\}$$

which can be rewritten as

$$v_j^k = \sum_{i \in V} \sum_{h=1,\ldots,n} z1_{ij}^{hk} = 1 - \sum_{i \in V} \sum_{h=1,\ldots,n} z2_{ij}^{hk} \quad \text{for all } j,k \in V \setminus \{1\}.$$

Now, if (31) is valid then the right-hand side term of the previous equation becomes $1 - v_k^j$ and (32) holds. The reverse holds in a similar way and we can conclude that (31) is valid if and only if (32) is, and thus, we have given an interpretation to the equalities given at the beginning of this section in terms of the v_j^k variables.

We denote by EC-MCF+ the model obtained by having (30) and (32) added to the EC-MCF model. Similarly, we denote by C-MCF+ the model obtained by adding the same constraints to the mdoel C-MCF. Our computational results will show that (30) and (32) are useful to produce further reductions on the gaps. Clearly, constraints (30) enable us to use many other inequalities that have been suggested in the referred papers using models with the v_j^k variables. One class of such inequalities will be discussed in Section 5.

3 Comparing with the Picard and Queyranne formulation

In this section we compare the linear programming relaxation of the EC-MCF model with the linear programming relaxation of the Picard and Queyranne [21], PQ, formulation that is defined on the previously presented z_{ij}^h variables:

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

$$\sum_{h=1,\ldots,n} \sum_{i \in V} z_{ij}^h = 1 \qquad \text{for all } j \in V \setminus \{1\} \tag{33}$$

$$\sum_{j \in V \setminus \{1\}} z_{1j}^1 = 1 \tag{34}$$

$$\sum_{i \in V} z_{ij}^h = \sum_{i \in V} z_{ji}^{h+1} \qquad \text{for all } j \in V \setminus \{1\}, h = 1,\ldots,n-1 \tag{35}$$

$$z_{ij}^h \in \{0,1\} \qquad \text{for all } (i,j) \in A, h = 1,\ldots,n \tag{36}$$

Constraints (33) guarantee that each node is visited exactly once. Constraints (35) state that an arc enters node j in position h if and only if another arc emanates from this node in position $h+1$. Constraint (34) states 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 (36) define the domain of the variables.

In order to compare it with the models of the previous section we note that any integer solution for the model defined by (34)-(36) is a circuit with n arcs in the original graph. Thus, the model can be rewritten as

$$\text{minimize} \sum_{(i,j) \in A} c_{ij} \sum_{h=1,\ldots,n} z_{ij}^h$$
$$(33) \text{ and}$$
$$\{(i,j) : z_{ij}^h = 1\} \text{ defines a } n\text{-circuit.}$$

Furthermore, the model defined by (34)-(36) corresponds to the path equations in a layered graph and thus, it is exact since the associated constraint matrix is totally unimodular. Thus, the PQ model is, in terms of linear programming bounds, the best model that can be built by using the generic formulation given above. Using the equations (27) that link the two sets of variables

x_{ij} and z_{ij}^h, the variables x_{ij} can be eliminated from the EC-MCF model (described in Section 2.3) leading to the following model (for simplicity, we skip this derivation from this chapter):

$$\begin{aligned}
\text{minimize } & \sum_{(i,j)\in A} c_{ij} \sum_{h=1,\dots,n} z_{ij}^h \\
& (33) \text{ and} \\
& \{(i,j) : g_{ij}^{hk} = 1\} \text{ defines a } n\text{-circuit}(k) && \text{for all } k \in V \setminus \{1\} && (17) \\
& z_{ij}^h = g_{ij}^{hk} && \text{for all } (i,j) \in A, h = 1,\dots,n, k \in V \setminus \{1\} && (37) \\
& z_{ij}^h \in \{0,1\} && \text{for all } (i,j) \in A, h = 1,\dots,n && (36)
\end{aligned}$$

where the constraints in (17) are (19)-(26) and constraints (37) are a disaggregation of the constraints (23). In fact, the new model presented in Godinho et al. [11] was given in this form. Consider, now, a weaker version of this model, where we consider (17) only for a given single $k \in V \setminus \{1\}$. The fact that the system (19)-(26) is exact and the relationship between the two subproblems enables us to show that the linear programming bound of this reduced model is at least as good as the linear programming bound of the PQ model. Thus, we have just shown that

Propositon 3.1 $v(\text{EC-MCF}_L) \geq v(PQ_L)$.

Note that this result has already been proved in different way in Godinho et al. [11]. We will show in Section 6 that the inequality can be strict for many instances. In Godinho et al. [11], the authors have given a complete description of the projection of $F(\text{EC-MCF}_L)$ into the space of the z_{ij}^h variables. This description includes a large set of cut-like inequalities defined in an appropriate layered graph. In particular, and with relevance for the follow-up of this chapter, we consider the following particular cases

$$z_{kp}^h \leq \sum_{i\in V, i\neq p,k} z_{pi}^{h+1} \quad \text{for all } p,k \neq 1, h = 1,\dots,n-2. \tag{38}$$

These constraints simply state that if arc (k,p) is in the solution in position h, then the arc in position $h+1$ cannot go to node k. As far as we know, these constraints were first proposed in the context of tree problems, see Gouveia [9], and later on in Costa et al. [3]. We denote by PQ+ the PQ model augmented with these inequalities. This model will be relevant for the dominance proof of the next section and we also have that

Propositon 3.2 $v(\text{EC-MCF}_L) \geq v(PQ+_L)$.

4 Comparing with the Sherali and Driscol Formulation

One of the earliest ideas for obtaining an extended formulation for the ATSP is to use additional variables u_i representing the position of node $i \in V$ with the generic meaning that $u_j = u_i + 1$ whenever $x_{ij} = 1$. In this way subtours are prevented. Adding constraints stating this condition to the assignment relaxation (3), (4) and (5) guarantees the extra constraints (2) since it is well known that the condition (2) is equivalent to "$\{(i,j) : x_{ij} = 1\}$ does not contain subtours". Miller, Tucker and Zemlin [18] proposed a well known set of linear constraints with the intended interpretation. Later on Desrochers and Laporte [6] have shown how to strengthen these inequalities, obtaining models with a tighter linear programming relaxation.

The development of the formulation given by Sherali and Driscoll [25] starts with a similar framework. However, they consider first a set of non-linear equalities in order to guarantee the previously stated condition "$u_j = u_i + 1$ whenever $x_{ij} = 1$". Then, they apply a specialized version of the Reformulation-Linearization Technique (RLT) of Sherali and Adams [23, 24] to this nonstandard restatement of the Miller-Tucker-Zemlin constraints [18]. After introducing new variables to linearize non-linear terms, and after some constraint manipulation they obtain a model using the same u_i variables and new variables y_{ij} representing the position of arc $(i,j) \in A$ in the circuit. As it is assumed that the circuit starts on node 1, we have $u_1 = 0$ and $y_{1j} = 0$ for any arc $(1,j)$. The constraints in the SD model corresponding to (2) are given by

$$\sum_{j \in V \setminus \{1\}} y_{ij} + (n-1)x_{i1} = u_i \qquad \text{for all } i \in V \setminus \{1\} \tag{39}$$

$$\sum_{i \in V \setminus \{1\}} y_{ij} + 1 = u_j \qquad \text{for all } j \in V \setminus \{1\} \tag{40}$$

$$x_{ij} \leq y_{ij} \leq (n-2)x_{ij} \qquad \text{for all } (i,j) \in A,\ i,j \neq 1 \tag{41}$$

$$u_j + (n-2)x_{ij} - (n-1)(1-x_{ji}) \leq y_{ij} + y_{ji} \leq u_j - (1-x_{ji}) \qquad \text{for all } (i,j) \in A,\ i,j \neq 1 \tag{42}$$

$$1 + (1-x_{1j}) + (n-3)x_{j1} \leq u_j \leq (n-1) - (n-3)x_{1j} - (1-x_{j1}) \qquad \text{for all } j \in V \setminus \{1\} \tag{43}$$

$$y_{ij} \geq 0 \qquad \text{for all } (i,j) \in A,\ i,j \neq 1 \tag{44}$$

$$u_i \geq 0 \qquad \text{for all } i \in V \setminus \{1\} \tag{45}$$

In this section we compare the linear programming relaxation of the EC-MCF model with the linear programming relaxation of the Sherali and Driscol [25] formulation, SD for short. More precisely we will show the stronger result that the linear programming relaxation of the PQ+ formulation (that is the PQ formulation plus inequalities (38)) dominates the linear programming relaxation of the SD model. Then, using the results from the previous section we conclude that the linear programming relaxation of the EC-MCF model dominates the linear programming relaxation of the SD model.

Oncan et al. [19] have shown that the linear programming relaxation of the SD model implies the linear programming relaxation of single commodity flow (SCF) model proposed by Gavish and Graves [8]. Here, we elaborate more on this relation by rewriting the SD model as the SCF model augmented with extra constraints. In fact, by combining (39) with (40) and introducing the variables y_{j1} such that $y_{j1} = (n-1)x_{j1}$ for all $j \in V \setminus \{1\}$, the SD model can be rewritten as follows

$$\sum_{j \in V} y_{ij} - \sum_{j \in V \setminus \{1\}} y_{ji} = 1 \qquad \text{for all } i \in V \setminus \{1\} \tag{46}$$

$$x_{ij} \leq y_{ij} \leq (n-2)x_{ij} \qquad \text{for all } (i,j) \in A,\ i,j \neq 1 \tag{47}$$

$$y_{j1} = (n-1)x_{j1} \qquad \text{for all } j \in V \setminus \{1\} \tag{48}$$

$$\sum_{i \in V \setminus \{1\}} y_{ij} + 1 = u_j \qquad \text{for all } j \in V \setminus \{1\} \tag{40}$$

$$u_j + (n-2)x_{ij} - (n-1)(1-x_{ji}) \leq y_{ij} + y_{ji} \leq u_j - (1-x_{ji}) \qquad \text{for all } (i,j) \in A,\ i,j \neq 1 \tag{42}$$

$$1 + (1-x_{1j}) + (n-3)x_{j1} \leq u_j \leq (n-1) - (n-3)x_{1j} - (1-x_{j1}) \qquad \text{for all } j \in V \setminus \{1\} \tag{43}$$

$$y_{ij} \geq 0 \qquad \text{for all } (i,j) \in A,\ i,j \neq 1 \tag{44}$$

$$u_i \geq 0 \qquad \text{for all } i \in V \setminus \{1\} \tag{45}$$

Observe that the constraints (46)-(48) and (44) define the SCF model. Thus, as stated above, we can view the SD model as the SCF augmented with constraints (40), (42), (43) and (45).

The interest of viewing the SD model in this way is due to a result in Gouveia and Voss [15] where it is shown that the linear programming relaxation of the PQ model dominates linear programming relaxation of the SCF model. More precisely, Gouveia and Voss have added the following constraints to the PQ model:

$$y_{ij} = \sum_{h=2,\dots,n-1} (h-1) z_{ij}^h \quad \text{for all } (i,j) \in A,\ i,j \neq 1 \tag{49}$$

(these constraints do not improve the linear programming bound) and then showed that the linear programming relaxation of the PQ model augmented with (49) implies all the constraints in the linear programming relaxation of the SCF model.

The z_{ij}^h variables can be related with the u_i variables in a similar way by using the following equalities:

$$u_j = \sum_{i \in V} \sum_{h=1,\dots,n-1} h z_{ij}^h \quad \text{for all } j \in V \setminus \{1\} \tag{50}$$

Again, the inclusion of (50) into PQ does not alter its LP bound. These two equalities enable us to relate the linear programming relaxations of the SD and PQ+ formulations. Based on the previous result mentioned in Gouveia and Voss [15] it is sufficient to prove that

Propositon 4.1 *The linear programming relaxation of PQ+ implies the constraints (40), (42), (43) and (45).*

Proof. Clearly, (45) is obvious. Consider the PQ+ model augmented with (49) and (50). We will show next that this augmented model implies constraints (40), (42) and (43).

1. Constraint (40) for a given node $j \in V \setminus \{1\}$.

   Consider a given node $j \in V \setminus \{1\}$ and the following trivial equality:

   $$\sum_{i \in V} \sum_{h=1,\dots,n-1} h z_{ij}^h = \sum_{i \in V \setminus \{1\}} \sum_{h=2,\dots,n-1} h z_{ij}^h + z_{1j}^1$$

   By extracting z_{1j}^1 in (33) for node j and replacing its value in the previous expression, we obtain:

   $$\begin{aligned} \sum_{i \in V} \sum_{h=1,\dots,n-1} h z_{ij}^h &= \sum_{i \in V \setminus \{1\}} \sum_{h=2,\dots,n-1} h z_{ij}^h + 1 - \sum_{i \in V \setminus \{1\}} \sum_{h=2,\dots,n-1} z_{ij}^h \\ &= \sum_{i \in V \setminus \{1\}} \sum_{h=2,\dots,n-1} (h-1) z_{ij}^h + 1. \end{aligned}$$

   Now, by using (49) on the right-hand side and (50) on the left-hand side we obtain (40) for the same node j.

2. Inequality (42) for a pair of nodes $i, j \in V \setminus \{1\}$.

   (a) Lower bound: $u_j + (n-2)x_{ij} - (n-1)(1 - x_{ji}) \leq y_{ij} + y_{ji}$

   Consider the inequalities (38) associated to arc (i,j) for $h = 2, \dots, n-2$:

   $$z_{ij}^h \leq \sum_{p \in V \setminus \{i\}} z_{jp}^{h+1} \quad h = 2, \dots, n-2.$$

   Multiplying by $(n-h+1)$ and summing up the resulting inequalities for $h = 2, \dots, n-2$, we obtain:

   $$\sum_{h=2,\dots,n-2} (n-h-1) z_{ij}^h \leq \sum_{p \in V \setminus \{i\}} \sum_{h=2,\dots,n-2} (n-(h+1)) z_{jp}^{h+1} = \sum_{p \in V \setminus \{i\}} \sum_{h=3,\dots,n-1} (n-h) z_{jp}^h$$

By considering $(n-2)z^2_{jp} \geq 0$ for all $p \in V \setminus \{1,i\}$ and observing that for $h = n-1$ we have $(n-h-1)z^h_{ij} = 0$, we obtain:

$$\sum_{h=2,\dots,n-1} (n-h-1)z^h_{ij} \leq \sum_{p\in V\setminus\{i\}} \sum_{h=2,\dots,n-1} (n-h)z^h_{jp}. \tag{51}$$

Consider now the constraints (35) for node j:

$$\sum_{p\in V} z^h_{pj} = \sum_{p\in V} z^{h+1}_{jp} \quad h = 1,\dots,n-1.$$

Multiplying both sides by h and summing up the resulting equalities for $h = 1,\dots,n-1$, we obtain:

$$\begin{aligned}\sum_{p\in V} \sum_{h=1,\dots,n-1} hz^h_{pj} &= \sum_{p\in V} \sum_{h=1,\dots,n-1} hz^{h+1}_{jp} \\ &= \sum_{p\in V} \sum_{h=2,\dots,n} (h-1)z^h_{jp} \\ &= \sum_{p\in V} \sum_{h=2,\dots,n-1} (h-1)z^h_{jp} + (n-1)z^n_{j1} \\ &= \sum_{p\in V\setminus\{i\}} \sum_{h=2,\dots,n-1} (h-1)z^h_{jp} + \sum_{h=2,\dots,n-1} (h-1)z^h_{ji} + (n-1)z^n_{j1}\end{aligned} \tag{52}$$

Now, adding (51) to (52) leads to:

$$\begin{aligned}&\sum_{h=2,\dots,n-2} (n-h-1)z^h_{ij} + \sum_{p\in V} \sum_{h=1,\dots,n-1} hz^h_{jp} \\ &\quad\leq \sum_{p\in V\setminus\{i\}} \sum_{h=2,\dots,n-1} (n-h+h-1)z^h_{jp} + \sum_{h=2,\dots,n-1} (h-1)z^h_{ji} + (n-1)z^n_{j1}.\end{aligned}$$

Now, since $n-h-1 = n-2-(h-1)$ we obtain:

$$\begin{aligned}(n-2) \sum_{h=2,\dots,n-2} z^h_{ij} + \sum_{p\in V} \sum_{h=1,\dots,n-1} hz^h_{jp} - (n-1) \sum_{p\in V\setminus\{i\}} \sum_{h=2,\dots,n-1} z^h_{jp} - (n-1)z^n_{j1} \\ \leq \sum_{h=2,\dots,n-1} (h-1)z^h_{ij} + \sum_{h=2,\dots,n-1} (h-1)z^h_{ji}.\end{aligned}$$

Now, by using (50) on the second term of the left-hand side, and using (49) on the terms on the right-hand side we obtain

$$(n-2) \sum_{h=2,\dots,n-2} z^h_{ij} + u_j - (n-1) \left[\sum_{p\in V\setminus\{i\}} \sum_{h=2,\dots,n-1} z^h_{jp} + z^n_{j1} \right] \leq y_{ij} + y_{ji}.$$

Then, by using the constraints (27) we have

$$(n-2)x_{ij} + u_j - (n-1) \sum_{p\in V\setminus\{i\}} x_{pj} \leq y_{ij} + y_{ji}.$$

Finally, using the assignment constraints on the third term of the left-hand side leads to the desired inequality.

(b) Upper bound: $y_{ij} + y_{ji} \le u_j - (1 - x_{ji})$.

We first note that by using (40) in the upper bounding expression of (42), we obtain

$$y_{ji} \le \sum_{p \in V \setminus \{1,i\}} y_{pj} + x_{ji}.$$

We prove, now, that the previous inequality is implied by the linear programming relaxation of PQ+. Consider constraints (38) associated to arc (j, i):

$$z_{ji}^{h+1} \le \sum_{p \in V \setminus \{i\}} z_{pj}^h \qquad h = 1, \ldots, n-2.$$

Multiplying by $(h-1)$ and adding the resulting inequalities for $h = 1, \ldots, n-2$ we obtain:

$$\sum_{h=2,\ldots,n-1} (h-2) z_{ji}^h = \sum_{h=1,\ldots,n-2} (h-1) z_{ji}^{h+1} \le \sum_{p \in V \setminus \{i\}} \sum_{h=2,\ldots,n-1} (h-1) z_{pj}^h.$$

Since for $h = 1$, we have $h - 1 = 0$, and by adding $(n-2) z_{pj}^{n-1} \ge 0$ for all $p \in V \setminus \{1, i\}$, we obtain:

$$\sum_{h=2,\ldots,n-1} (h-1) z_{ji}^h - \sum_{h=2,\ldots,n-1} z_{ji}^h \le \sum_{p \in V \setminus \{i\}} \sum_{h=2,\ldots,n-1} (h-1) z_{pj}^h.$$

Then, by using the constraints (49) and the (27) the result follows.

3. Inequality (43) for a given node $j \in V \setminus \{1\}$.

(a) Lower bound: $u_j \ge 1 + (1 - x_{1j}) - (n-3) x_{j1}$.

Since the z_{ij}^h variables are nonnegative, we have that

$$\sum_{i \in V \setminus \{1\}} \sum_{h=2,\ldots,n-2} (h-2) z_{ij}^h \ge 0$$

.

Multiplying by 2 the equality (33) for node j and adding the resulting equality to the previous inequality, we obtain:

$$2 \sum_{i \in V} \sum_{h=1,\ldots,n-1} z_{ij}^h + \sum_{i \in V \setminus \{1\}} \sum_{h=2,\ldots,n-2} (h-2) z_{ij}^h \ge 2.$$

Next, by extracting the terms for $h = 1$ and $h = n-1$ of the first summation, we obtain:

$$2 z_{1j}^1 + 2 \sum_{i \in V \setminus \{1\}} \sum_{h=2,\ldots,n-2} z_{ij}^h + 2 \sum_{i \in V \setminus \{1\}} z_{ij}^{n-1} + \sum_{i \in V \setminus \{1\}} \sum_{h=2,\ldots,n-2} (h-2) z_{ij}^h \ge 2$$

which is equivalent to

$$2 z_{1j}^1 + 2 \sum_{i \in V \setminus \{1\}} z_{ij}^{n-1} + \sum_{i \in V \setminus \{1\}} \sum_{h=2,\ldots,n-2} (2 + h - 2) z_{ij}^h \ge 2$$

which, in turn, is equivalent to

$$z_{1j}^1 + ((n-1) - (n-3)) \sum_{i \in V \setminus \{1\}} z_{ij}^{n-1} + z_{1j}^1 + \sum_{i \in V \setminus \{1\}} \sum_{h=2,\ldots,n-2} h z_{ij}^h \ge 2$$

and to

$$(n-1) \sum_{i \in V \setminus \{1\}} z_{ij}^{n-1} + \sum_{i \in V} \sum_{h=1,\ldots,n-2} h z_{ij}^h \ge 1 + (1 - z_{1j}^1) + (n-3) \sum_{i \in V \setminus \{1\}} z_{ij}^{n-1}$$

and, finally, equivalent to

$$\sum_{i \in V} \sum_{h=1,\ldots,n-1} h z_{ij}^h \geq 1 + (1 - z_{1j}^1) + (n-3) \sum_{i \in V \setminus \{1\}} z_{ij}^{n-1}.$$

Next, we use the inequalities (35) for $h = n-1$ and node j in the last term to obtain:

$$\sum_{i \in V} \sum_{h=1,\ldots,n-1} h z_{ij}^h \geq 1 + (1 - z_{1j}^1) + (n-3) z_{j1}^n.$$

Finally, we use the constraints (50) and (27) and the result follows.

(b) Upper bound: $u_j \leq (n-1) - (n-3)x_{1j} - (1 - x_{j1})$.

Since the z_{ij}^h variables are nonnegative, we have that

$$\sum_{i \in V \setminus \{1\}} \sum_{h=2,\ldots,n-2} (h + 1 - n) z_{ij}^h \leq 0.$$

Multiplying by 2 the equality (33) for node j and adding the resulting equality to the previous inequality, we obtain:

$$\sum_{i \in V} \sum_{h=1,\ldots,n-1} 2 z_{ij}^h + \sum_{i \in V \setminus \{1\}} \sum_{h=2,\ldots,n-1} (h + 1 - n) z_{ij}^h \leq 2.$$

By extracting the term for $h = 1$ of the first summation, we obtain:

$$\sum_{i \in V \setminus \{1\}} \sum_{h=2,\ldots,n-1} 2 z_{ij}^h + 2 z_{1j}^1 + \sum_{i \in V \setminus \{1\}} \sum_{h=2,\ldots,n-1} (h + 1 - n) z_{ij}^h \leq 2$$

which can be written as

$$2 z_{1j}^1 + \sum_{i \in V \setminus \{1\}} \sum_{h=2,\ldots,n-1} (2 + h + 1 - n) z_{ij}^h \leq 2$$

and as

$$z_{1j}^1 + \sum_{i \in V \setminus \{1\}} \sum_{h=2,\ldots,n-1} h z_{ij}^h - (n-3) \sum_{i \in V \setminus \{1\}} \sum_{h=2,\ldots,n-1} z_{ij}^h \leq 2 - z_{1j}^1$$

and as

$$\sum_{i \in V} \sum_{h=1,\ldots,n-1} h z_{ij}^h \leq 2 - z_{1j}^1 + (n-3) \sum_{i \in V \setminus \{1\}} \sum_{h=2,\ldots,n-1} z_{ij}^h.$$

By using the constraints (49), (50) and (27) the previous inequality can be rewritten as

$$u_j \leq 2 - x_{1j} + (n-3) \sum_{i \in V \setminus \{1\}} x_{ij}.$$

Now, use the assignment constraints on the previous inequality to obtain:

$$u_j \leq 2 - x_{1j} + (n-3)(1 - x_{ij}) = 2 - x_{1j} + (n-3) - (n-3)x_{1j} = (n-1) - (n-3)x_{1j} - x_{1j}.$$

Since $(1 - x_{j1}) \geq 0$ and $x_{1j} \geq 0$, we obtain the desired inequality.

□

Thus, we have just proved that

Propositon 4.2 $v(PQ+_L) \geq v(SD_L)$.

Combining this result with Proposition 3.2 we obtain

Propositon 4.3 $v(EC\text{-}MCF_L) \geq v(SD_L)$.

5 Comparing with the Sherali, Sarin and Tsai Formulation

Several extended formulations for the ATSP that have been proposed in the literature use the binary precedence variables v_j^k indicating whether j is before k in the tour. As noted before, the first such model was proposed by Claus [2] leading to an enhanced version of the P-MCF model (see below). However, this model uses the precedence variables as a mean to tighten the linear programming relaxation of the P-MCF model (in fact, from a strictly point of view, the precedence variables are not even needed since they are related to the flow variables by equalities; however, their inclusion make the model easier to read). As far as we know, the first models that use the two sets of variables $\{x_{ij}, v_i^j\}$ alone to describe (2) of the generic formulation were proposed by Gouveia and Pires [13]. These models can be viewed as a disaggregation (and tightening) of the Miller-Tucker-Zemlin model. Later on Gouveia and Pires [14], use the relation between flow variables of the P-MCF model and the v_i^j variables to strengthen the previous models as well as to strengthen the P-MCF model. Sarin et al. [22] improve the models of the earlier paper by including inequalities from the related linear ordering problem (see, section 2.6). Gouveia and Pesneau [12] strengthen these models by using several classes of exponential sized sets of constraints involving the two sets of variables. Max flow / min cut computations are used to separate constraints from these classes.

Sherali et al [26] present formulations for the ATSP that also use the three sets of variables y_{ij}^k, v_i^j and x_{ij} (in this sense, they are similar to the models presented in Gouveia and Pires [14]). The formulations they use are again obtained by applying the Reformulation-Linearization Technique (RLT) of Sherali and Adams [23, 24]. However, they apply it to the strongest formulation with precedence variables that has been presented in Sarin et al [22]. Thus, the procedure starts with a formulation including the two sets of variables $\{x_{ij}, v_i^j\}$ and ends up with formulation using the same two sets plus the variables y_{ij}^k as defined in section 2.2 for the P-MCF model. The variables y_{ij}^k result from linearizing nonlinear terms arising in the reformulation linearization procedure (in this sense, the approach is similar to the approach used to derive the SD formulation described in the previous section where a formulation starting with two sets of variables $\{x_{ij}, u_i\}$ leads to a formulation with three sets of variables). The main difference with the formulations in Gouveia and Pires [14] which use the same sets of variables is that Sherali et al. [26] use the linear ordering inequalities (which, as shown in Sarin et al. [22], were, in general, stronger than the ones used in [14]).

Before presenting their model with the strongest linear programming relaxation, we note that Oncan et al. [19] have related the linear programming relaxation of several of these formulations.

The part of their strongest model corresponding to (2) of the generic model given in subsection 2.1 is as follows:

$$v_j^k \geq x_{1j} \qquad \text{for all } j,k \in V \setminus \{1\} \tag{53}$$

$$v_k^i \geq x_{i1} \qquad \text{for all } k,i \in V \setminus \{1\} \tag{54}$$

$$x_{1i} + \sum_{p \in V \setminus \{1,j\}} y_{pi}^j = v_i^j \qquad \text{for all } i,j \in V \setminus \{1\} \tag{55}$$

$$\sum_{p \in V \setminus \{1,j\}} y_{ip}^j + x_{ij} = v_i^j \qquad \text{for all } i,j \in V \setminus \{1\} \tag{56}$$

$$y_{ij}^k \leq x_{ij} \qquad \text{for all } i,j,k \in V \setminus \{1\}, i,j \neq k \tag{57}$$

$$v_i^k + v_k^i = 1 \qquad \text{for all } i,k \in V \setminus \{1\} \tag{58}$$

$$(v_i^j + x_{ji}) + v_j^k + v_k^i \leq 2 \qquad \text{for all } i,j,k \in V \setminus \{1\} \tag{59}$$

$$y_{ij}^k \in \{0,1\} \qquad \text{for all } (i,j) \in A, j \neq 1, \text{ for all } k \in V \setminus \{1\}, i \neq k \tag{60}$$

$$v_i^j \in \{0,1\} \qquad \text{for all } i,j \in V \setminus \{1\} \tag{61}$$

In this section we compare the linear programming relaxation of the EC-MCF+ model with the linear programming relaxation of the Sherali, Sarin and Tsai [26] formulation, SST for short. We will first consider a variant of the SST that is obtained by removing one set of constraints from the SST model namely the constraints (59) (we will denote this variant by W-SST). The constraints (59) belong to the class of constraints mentioned at the end of section 2.6. As far as we know, they were first proposed by Letchford (see also Sarin et al. [22] and Gouveia and Pesneau [12]). These inequalities are a lifted version (obtained by considering the extra term $+x_{ji}$ on the left-hand side) of the 3-cycle inequalities known from the linear ordering polytope.

We will show that:

1. The linear programming relaxation of the W-SST model is equivalent to the linear programming relaxation of an enhanced version of the P-MCF model which has already been presented by Claus [2]. For this reason we will denote this variant by P-MCF+.
2. The linear programming relaxation of the EC-MCF+ model dominates the linear programming relaxation of the P-MCF+ model and as a consequence, dominates the linear programming relaxation of the W-SST model.
3. Similarly the linear programming relaxation of the C-MCF+ model dominates the linear programming relaxation of the P-MCF+ model and again we can conclude that it dominates the linear programming relaxation of the W-SST model.
4. Then, we will reintroduce the missing constraints in SST. In particular we point out that by adding these missing constraints to EC-MCF+ or to the C-MCF+ model we will obtain a strict dominance over SST. And, we will make some comments about the effect of using computationally these constraints in the SST and in the EC-MCF+ models. In particular we will point out that these constraints are very effective in SST but are not in EC-MCF+ (although they are not dominated).

Consider the P-MCF model presented in section 2.2. Claus [2] has introduced an enhanced version of this formulation by adding constraints linking the y_{ij}^k and v_i^j variables and by adding the linear ordering equalities (32) discussed in subsection 2.6 (for simplicity, we will relabel the equalities). As noted before, we denote by P-MCF+ model this enhanced version.

$$\sum_{j \in V\setminus\{1\}} y_{ij}^k - \sum_{j\in V} y_{ji}^k = \begin{cases} 1 & i=1 \\ 0 & i \neq 1,k \\ -1 & i=k \end{cases} \quad \text{for all } k \in V\setminus\{1\} \tag{62}$$

$$y_{ij}^k \leq x_{ij} \quad \text{for all } (i,j)\in A, j\neq 1, i\neq k \tag{63}$$

$$v_i^j = \sum_{p\in V} y_{pi}^j \quad \text{for all } i,j \in V\setminus\{1\} \tag{64}$$

$$v_i^j + v_j^i = 1 \quad \text{for all } i,j\in V\setminus\{1\} \tag{65}$$

$$y_{ij}^k \in \{0,1\} \quad \text{for all } (i,j)\in A, j\neq 1, \text{ for all } k\in V\setminus\{1\}, i\neq k \tag{66}$$

$$v_i^j \in \{0,1\} \quad \text{for all } i,j\in V\setminus\{1\}. \tag{67}$$

First, note that as noted in Section 2.2, the constraints (63) are satisfied as equalities when $k=j$ and when $i=1$. Thus we will use them explicitly as

$$y_{1j}^k = x_{1j} \quad \text{for all } j,k\in V\setminus\{1\} \tag{68}$$

or

$$y_{ij}^j = x_{ij} \quad \text{for all } (i,j)\in A, j\neq 1. \tag{69}$$

Propositon 5.1 $v(P\text{-}MCF+_L) = v(W\text{-}SST_L)$.

Proof. We prove first that $v(\text{P-MCF}+_L) \geq v(\text{W-SST}_L)$. It is sufficient to show that (53)-(56) are redundant in the linear programming relaxation of P-MCF+.

Obtaining (53) - Consider the inequality (68) for a pair $j,k\in V\setminus\{1\}$, $y_{1j}^k = x_{1j}$. By the non-negativity constraints of the y_{ij}^k variables we obtain:

$$\sum_{p\in V\setminus\{1,k\}} y_{pj}^k + y_{1j}^k \geq x_{1j}$$

Now use constraint (64) on the left-hand side to obtain (53) for the same pair j,k.

Obtaining (54) - Let $i,k\in V\setminus\{1\}$. By adding constraint (63) for $j=2,\ldots,n$, $j\neq i$, we obtain:

$$\sum_{j\in V\setminus\{1\}} y_{ij}^k \leq \sum_{j\in V\setminus\{1\}} x_{ij}.$$

Using (62) for $j\neq 1,k$ in the previous constraint we obtain

$$\sum_{j\in V} y_{ji}^k \leq \sum_{j\in V\setminus\{1\}} x_{ij}.$$

By using (64) on the previous inequality, we obtain:

$$v_i^k \leq \sum_{j\in V\setminus\{1\}} x_{ij}.$$

Now, by using the assignment constraints (4) associated to node i, and rewriting we have

$$v_i^k \leq 1 - x_{i1}.$$

By using (65) we obtain finally (54) for the same pair of nodes.

Obtaining (55) and (56) - Let $i,j\in V\setminus\{1\}$ and consider constraint (64) for this pair of nodes

$$v_i^j = \sum_{p \in V} y_{pi}^j.$$

Observe that by extracting the term for $p = 1$ from the summation and using (68), we obtain (55) for the same pair of nodes.

On the other hand by using (62) with (64), we obtain

$$v_i^j = \sum_{p \in V \setminus \{1\}} y_{ip}^j.$$

Extracting the term for $p = j$ from the summation and using (69), we obtain (56) for the same pair of nodes.

And thus (53)-(56) are implied by the linear programming relaxation of the P-MCF+ model.

We prove now that $v(\text{P-MCF+}_L) \le v(\text{W-SST}_L)$. It is sufficient to show that (62), (64), (68) and (69) are redundant in the linear programming relaxation of W-SST.

First, we note that the W-SST model does not include variables y_{1j}^k for all $j, k \in V \setminus \{1\}$ and variables y_{ij}^j for all $(i, j) \in A$, $i, j \neq 1$. Thus we introduce those variables in the model and we relate them to variables x_{ij} as follows:

$$y_{1j}^k = x_{1j} \quad \text{for all } j, k \in V \setminus \{1\}$$

and

$$y_{ij}^j = x_{ij} \quad \text{for all } (i, j) \in A, j \neq 1.$$

and obtaining, respectively (68) and (69). Note that in the context of W-SST, these two sets of equalities only define the new variables and do not modify its linear programming bound.

Obtaining (64) - Let $i, j \in V \setminus \{1\}$ and consider constraint (55). By using (68) in (55), we obtain (64) for the same pair of nodes.

Obtaining (62) - Let $i, k \in V \setminus \{1\}$. By using (69) in (56) we obtain

$$v_i^j = \sum_{p \in V \setminus \{1,j\}} y_{ip}^j + y_{ij}^j = \sum_{p \in V \setminus \{1\}} y_{ip}^j$$

Combining with (64) we obtain constraint (62) for $i \in V \setminus \{1, k\}$.

By considering a given node $k \in V \setminus \{1\}$ and summing up equality (68) for all $j \in V \setminus \{1\}$, we obtain:

$$\sum_{j \in V \setminus \{1\}} y_{1j}^k = \sum_{j \in V \setminus \{1\}} x_{1j}.$$

Using the assignment constraint for node 1 on the right-hand side, we obtain (62) for $i = 1$.

Finally, by summing up equality (69) for all $j \in V \setminus \{1\}$, we obtain:

$$\sum_{j \in V \setminus \{1\}} y_{jk}^k = \sum_{j \in V \setminus \{1\}} x_{jk}.$$

Adding equality (68) written for node k to the previous equality, we obtain:

$$\sum_{j \in V} y_{jk}^k = \sum_{j \in V} x_{jk}.$$

Using the assignment constraint on the right-hand side, we obtain (62) for $i = k$.

And thus (62), (64), (68) and (69) are implied by the linear programming relaxation of the W-SST model. □

TEST	SD	PQ+	P-MCF	P-MCF+	SST	EC-MCF	EC-MCF+
ftv33	4.78	4.51	0.00	0.00	0.00	0.00	0.00
ftv35	3.90	3.84	1.06	0.87	0.65	0.85	0.39
ftv38	3.26	3.11	1.02	0.84	0.64	0.83	0.52
ftv44	2.43	2.43	1.74	1.38	1.10	1.59	1.09

Table 1: Results for ATSP instances.

As a consequence of Proposition 2.1 and of the observations made in Section 2.4 we conclude that

Propositon 5.2 $v(EC\text{-}MCF+_L) \geq v(W\text{-}SST_L)$.

Propositon 5.3 $v(C\text{-}MCF+_L) \geq v(W\text{-}SST_L)$.

Clearly, if we use (59) in the two models EC-MCF+ and C-MCF+, (let us denote by EC-MCF++ and C-MCF++ the two models obtained in this way) we obtain the stronger results

Propositon 5.4 $v(EC\text{-}MCF{+}{+}_L) \geq v(SST_L)$.

Propositon 5.5 $v(C\text{-}MCF{+}{+}_L) \geq v(SST_L)$.

6 Computational Results and Conclusions

In this section we empirically evaluate the quality of the lower bounds given by the models discussed in the previous sections. As noted before, in this chapter we are concerned with the strength of the linear programming relaxations and not with the time needed to solve it or the corresponding integer program. For comparing the models, we use the known ATSP instances taken from TSP Lib (`http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/`).

In this chapter we compare the linear programming relaxation of the following models: SD, PQ+, P-MCF, P-MCF+, SST, EC-MCF and EC-MCF+. The linear programming relaxations where solved with the barrier solver CPLEX 12.

The next table gives the linear programming relaxation gaps of these models. The gaps are computed as [(Optimal Value - $v(\mathrm{P}_L)$)/Optimal value] * 100 (where P denotes the model) for each instance.

Several comments are interesting. First, the results obtained from our computational experiment show that the EC-MCF+ model produces, as far as we know, the best gaps produced by known compact models, for the reported instances.

Second, it is also worth to compare the effect of the linear ordering equalities LO1 in the P-MCF model (leading to P-MCF+ which is one of the models presented by Claus) and in EC-MCF (leading to EC-MCF+). We can conclude that these equalities lead to reasonable improvements when added either to the P-MCF or EC-MCF model. In particular in the later case, they lead to the best bounds.

Third, it is interesting to analyze the effect of adding the lifted 3-cycle inequalities (59) in the models. The addition of (59) to the model P-MCF+ results in the SST model and thus, as can be seen in Table 1, leads to some improvements. Adding, (59) to EC-MCF+ also may lead to further reduction on the gaps, for instance, for ftv35, the addition of these constraints to EC-MCF+ leads to a gap of 0.17 which is a reduction of more than 50% of the gap obtained without including these constraints). However, the EC-MCF+ model with constraints (59) is difficult to use for larger instances.

We also note that the gaps from SD and PQ+ have been included only for the sake of completeness, however, they are substantially weaker than the gaps given by the other formulations.

In terms of comparing ideas from these formulations, it is worth noting the two parallel and different strategic approaches leading to SST and to EC-MCF:

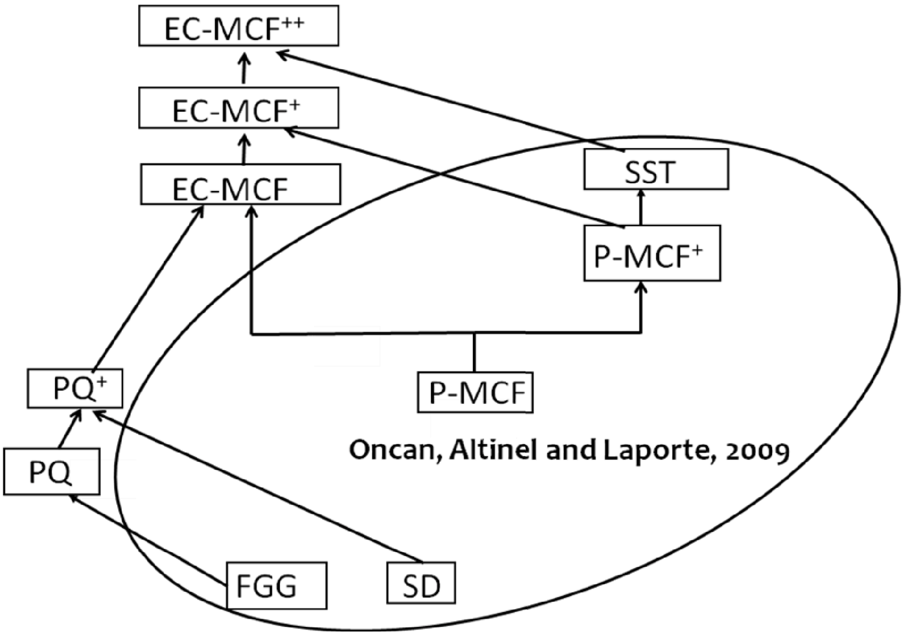

Figure 2: Relative strength of the corresponding linear programming relaxations

1. the one leading to SST that basically is nothing else than intersecting the multicommodity flow reformulation (here represented by the P-MCF model) with some inequalities from the linear ordering problem (here represented by the inequalities (32) and (59) - note that as stated before, the (59) inequalities are stronger than the original inequalities from the linear ordering problem);
2. the one leading to the EC-MCF and EC-MCF+ models that can be seen as moving from a path subproblem to a n-circuit subproblem combined with the disaggregation specified in section 2.3 (or alternatively, moving from a path subproblem to a n-circuit subproblem in the space of the time-dependent variables as originally done in Godinho et al [11]).

It is also interesting to note that both approaches are also very relevant (or even more relevant) for versions of the ATSP that capture information given by the extra variables. As shown in Godinho et al [11], the approach 2. leads to very tight models for the time-dependent TSP and the cumulative TSP. As shown in Sarin et al. [22] and Gouveia and Pesneau [12], approach 1. leads to tight models for the Precedence Constrained TSP problem. Furthermore, Gouveia and Pesneau [12] also propose several ideas worth exploring in order to strengthen this class of models. Finally, equalities (30) also enable us to use the models of approach 2. for Precedence Constrained TSP.

We conclude this chapter by presenting a table which gives a relative relation of the strength of the linear programming relaxations of the relevant models mentioned in this chapter. We have also contextualized this table in terms of the one from Oncan et al [19] taking from here the relevant models for the new comparison.

References

[1] E. Balas, and M. Fischetti. “On the monotonization of polyhedra”, Math. Program. 77, 59-84, 1997.

[2] A. Claus. “A new formulation for the traveling salesman problem”, SIAM Journal on Algebraic and Discrete Methods, 5, 21-5, 1984.

[3] 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.

[4] G. Dahl. “Notes on polyhedra associated with hop-constrained paths”, O. R. Letter, 25, 97-100, 1999.

[5] G. Dahl, N. Foldnes, and L. Gouveia.”A note on hop-constrained walk polytopes”, O.R. Letters, 32, 345-349, 2004.

[6] M. Desrochers and G. Laporte. “Improvements to the Miller-Tucker-Zemlin subtour elimination constraints”, Operation Research Letters, 10, 27-36, 1991.

[7] 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.

[8] B. Gavish, and S.C. Graves. “The traveling salesman problem and related problems”, Working Paper OR-078-78, Operations Research Center, MIT, Cambridge, MA, 1978.

[9] L. Gouveia. “Using Hop-Indexed Models for Constrained Spanning and Steiner Tree Problems” Telecommunications Network Planning, B. Samsò and P. Soriano, Eds., 21-32, 1999.

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

[11] M. T Godinho, L. Gouveia, and P. Pesneau. “Natural and Extended formulations for the Time-Dependent Traveling Salesman Problem”, Working paper do CIO, 8-2010, 2010.

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

[13] L. Gouveia and J.M. Pires. “The Asymmetric Traveling Salesman Problem and a Reformulation of the Miller-Tucker-Zemlin Constraints”, European Journal of Operational Research, 112, 134-146, 1999.

[14] L. Gouveia and J.M. Pires. “The Asymmetric Traveling Salesman Problem: on Generalisations of Disaggregated Miller-Tucker-Zemlin Constraints”, Discrete Applied Mathematics, 112, 129-145, 2001.

[15] L. Gouveia and S.Voss. “A classification of formulations for the time-dependent Traveling Salesman Problem”, European Journal of Operations Research, 83, 69-82, 1995.

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

[17] R. Loulou. “On multicomodity flow formulations for the TSP”, Working Paper, McGill University, Montreal, 1988.

[18] C. Miller, A. Tucker and R. Zemlin. “Integer programming formulations and traveling salesman problems”, Journal of ACM, 7, 326-329, 1960.

[19] 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.

[20] A. Orman and H. Williams. “A Survey of Different Integer Programming Formulations of the Traveling Salesman Problem”, in Optimisation, Econometric and Financial Analysis, Advances in Computational Management Science, Vol 9, Part I, 91-104, Springer, 2007.

[21] 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.

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

[23] H. D. Sherali, and W. P. Adams. “A Hierarchy of Relaxations Between the Continuous and Convex Hull Representations for Zero-One Programming Problems”, SIAM J. Discrete Math. 3, 411-430, 1990.

[24] H. D. Sherali, and W. P. Adams. "A Hierarchy of Relaxations and Convex Hull Characterizations for Mixed-integer Zero-one Programming Problems", Discrete Applied Mathematics, 52, 83-106, 1994.

[25] 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.

[26] 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.

[27] R. T. Wong. "Integer programming formulations of the traveling salesman problem", Proceedings of the IEEE international conference of circuits and computers, 149-52, 1980.