

Combining and projecting flow models for the (Precedence Constrained) Asymmetric Traveling Salesman Problem

Luis Gouveia, Pierre Pesneau, Mario Ruthmair, Daniel Santos

► **To cite this version:**

Luis Gouveia, Pierre Pesneau, Mario Ruthmair, Daniel Santos. Combining and projecting flow models for the (Precedence Constrained) Asymmetric Traveling Salesman Problem. *Networks*, Wiley, 2017, 71 (4), pp.451-465. <<https://onlinelibrary.wiley.com/doi/10.1002/net.21765>>. <10.1002/net.21765>. <hal-01655793>

HAL Id: hal-01655793

<https://hal.inria.fr/hal-01655793>

Submitted on 5 Dec 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Combining and projecting flow models for the (Precedence Constrained) Asymmetric Traveling Salesman Problem

Luis Gouveia^{1*}, Pierre Pesneau², Mario Ruthmair³, Daniel Santos¹

¹ University of Lisbon, Faculty of Sciences, Dept. of Statistics and OR, CMAF-CIO, Portugal
legouveia@fc.ul.pt, drsantos@fc.ul.pt

² University of Bordeaux, UMR 5251, INRIA Bordeaux - Sud-Ouest, France
pierre.pesneau@math.u-bordeaux.fr

³ University of Vienna, Department of Statistics and Operations Research, Austria

Abstract

There are many ways of modeling the Asymmetric Traveling Salesman Problem (ATSP) and the related Precedence Constrained ATSP (PCATSP). In this paper we present new formulations for the two problems that result from combining precedence variable based formulations with network flow based formulations. The motivation for this work is a property of the so-called GDDL inequalities (see Gouveia and Pesneau, 2006), the “disjoint sub-paths” property, that is explored to create formulations that combine two (or more) disjoint path network flow based formulations. Several sets of projected inequalities, in the space of the arc and precedence variables, and in the spirit of many inequalities presented in Gouveia and Pesneau (2006), are obtained by projecting these network flow based formulations. Computational results are given for the PCATSP and the ATSP to evaluate the quality of the new inequalities.

Keywords: traveling salesman, precedence constraints, integer linear programming, reformulation, cutting plane algorithm, valid inequalities

1 Introduction

Given a complete directed graph $G = (V, A)$ with $V = \{1, \dots, n\}$ and a cost function c associated with the arcs of A , the Asymmetric Traveling Salesman Problem (ATSP) seeks for a minimum-cost Hamiltonian circuit in G .

The precedence constrained case (see, for instance, Balas et al., 1995; Gouveia and Pesneau, 2006; Gouveia and Ruthmair, 2015; Letchford and Salazar-González, 2016) arises when we consider that the circuit starts and ends at a given node, e.g., node 1, and we introduce precedence constraints between some pairs of nodes. A precedence constraint $i \prec j$ between two distinct nodes $i, j \in V \setminus \{1\}$ states that, starting the circuit from node 1, node i should be visited before node j . Let $B \in (V \setminus \{1\}) \times (V \setminus \{1\})$ be the set of precedence constraints, that is, if $(i, j) \in B$, then $i \prec j$. Given a set B of precedence constraints, the ATSP with precedence constraints (PCATSP) consists of finding a minimum-cost Hamiltonian circuit that satisfies all precedence constraints of B .

To formulate the ATSP consider the set of binary variables x_{ij} such that $x_{ij} = 1$ if arc $(i, j) \in A$ is in the tour and $x_{ij} = 0$ otherwise. The following generic formulation for the ATSP is given in many papers

*Corresponding author.

(see, e.g., Langevin et al., 1990; Öncan et al., 2009; Godinho et al., 2011; Roberti and Toth, 2012):

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

$$\text{s.t.} \quad \sum_{j:(i,j) \in A} x_{ij} = 1, \quad \forall i \in V, \quad (1)$$

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

$$\{(i,j) \in A : x_{ij} = 1\} \text{ does not contain subtours,} \quad (3)$$

$$x_{ij} \in \{0, 1\}, \quad \forall (i,j) \in A. \quad (4)$$

Constraints (1) and (2) are assignment constraints and (3) is a generic way of representing a set of constraints that prevents the solution from containing subtours (or, equivalently, that guarantees that the solution is a single connected circuit). There are several ways of modeling (3) either by using the variables x alone (in so-called natural formulations) or by involving other variables (in so-called extended formulations). The usual way of modeling (3) with the variables x alone is to use the subtour elimination constraints

$$x(A(S)) \leq |S| - 1, \quad \forall S \subset V, S \neq \emptyset \quad (5)$$

or their equivalent cut form

$$x(V \setminus S, S) \geq 1, \quad \forall S \subset V, S \neq \emptyset. \quad (6)$$

Above, $A(S)$ is the set $\{(i,j) \in A : i, j \in S, i \neq j\}$, $x(A') = \sum_{(i,j) \in A'} x_{ij}$ denotes the sum of the x variables for arc set $A' \subseteq A$, and $x(P, Q) = \sum_{(i,j) \in A: i \in P, j \in Q} x_{ij}$ represents the sum of the x variables for all arcs going from node set P to node set Q . These notations will be used from now on.

It is well known that there are several different models that are equivalent, in terms of the corresponding linear programming (LP) bounds, for the ATSP (see, for instance, the previous references). In particular, we obtain two LP equivalent models by using either (5) or (6). It is also well known that one can obtain an extended and compact flow based formulation (with a polynomial number of constraints and variables) that is equivalent, in terms of the corresponding LP bound, to the formulation using the exponentially-sized set of cut-like inequalities (6) and, in fact, the corresponding flow based system is closely associated with the separation routine for inequalities (6). Such a model and its equivalence to the cut model was first proposed by Wong (1980) (see also Claus (1984) and the previous references). We omit this model from this paper but we observe that a similar connection will be used in this paper to derive new models for the ATSP and the PCATSP.

To model (3), Gouveia and Pires (1999) proposed several extended models using an additional set of binary variables v_i^j , $i, j \in V \setminus \{1\}$, $i \neq j$, such that $v_i^j = 1$ when node i is in the path from node 1 to node j or, equivalently, when node i precedes node j in the tour; and $v_i^j = 0$ otherwise. In one of the formulations they proposed for the ATSP, (3) is modeled by the following set of inequalities

$$x_{ij} \leq v_i^j, \quad \forall (i,j) \in A, i, j \neq 1, \quad (7)$$

$$x_{ij} + v_j^i \leq 1, \quad \forall (i,j) \in A, i, j \neq 1, \quad (8)$$

$$v_k^j + x(A(S)) \leq v_k^i + |S| - 1, \quad \forall i, j, k \in V \setminus \{1\}, \forall \{i, j\} \subseteq S \subseteq V \setminus \{1, k\}, \quad (9)$$

$$v_i^j \in \{0, 1\}, \quad \forall (i,j) \in V \setminus \{1\}, i \neq j. \quad (10)$$

Inequalities (9) are called generalized disaggregated Desrochers and Laporte (GDDL) inequalities since they generalize the particular case where $S = \{i, j\}$ (called DDL inequalities) which, in turn, can be seen as a disaggregated form of the Desrochers and Laporte inequalities (see Desrochers and Laporte, 1991) proposed in the context of the Miller-Tucker-Zemlin constraints (see Miller et al., 1960; Gouveia and Pires, 1999, 2001).

In order to obtain a valid and compact model, we can use only the DDL inequalities. In this case, however, it is preferable to use the following lifted version of the DDL inequalities, since for several instances tested by Gouveia and Pesneau (2006) the LP bounds are better:

$$v_k^j + v_j^i + x_{ij} \leq v_k^i + 1 \quad \forall \text{ sequences } (i, j, k) \text{ of 3 distinct nodes of } V \setminus \{1\}. \quad (11)$$

Observe that the GDDL constraints are meaningful only when $x(A(S)) = |S| - 1$, that is, when the nodes in the set S form an elementary path. Then, if node k (outside of S) precedes node j (inside of S), then node k must also precede node i (which is also in S). Later on, Gouveia and Pesneau (2006) showed that the LP relaxation bound of the model using (5) (or (6)) is the same as the one using (7)–(10) to model (3) for the ATSP.

The advantage of using the second system is that it also provides a valid formulation for the PCATSP, simply by adding constraints

$$v_i^j = 1, \quad \forall (i, j) \in B. \quad (12)$$

Gouveia and Pesneau (2006) also made use of the fact that the v variables can be used to model the related linear ordering problem (Grötschel et al., 1984), and “borrowed” some constraints from this problem, in particular

$$v_i^j + v_j^i = 1, \quad \forall i, j \in V \setminus \{1\}, i \neq j. \quad (13)$$

Clearly, by adding (13) to system (7)–(10), one of the constraints (7) or (8) becomes redundant. Also, by adding these inequalities, the LP bound improves for some instances, although the improvement is more noticeable for instances of the PCATSP.

Recently, Barbato et al. (2016) described a problem that can be viewed as a combination of two separate ATSPs linked by a set of loading related constraints. The authors model the problem as two separate PCATSPs linked by a set of consistency constraints. Their model and corresponding cutting plane method strongly rely on the GDDL inequalities (as well as the simple cut inequalities that shall be mentioned in the next section).

By using adequately the assignment constraints (1), or (2), and inequalities (13), the GDDL inequalities (9) can also be written in a cut form in the following way:

$$x(S', S) \geq v_i^k + v_k^j, \quad \forall \text{ sequences } (i, j, k) \text{ of 3 distinct nodes of } V \setminus \{1\}, \quad (14) \\ \forall \text{ partitions } (S', S) \text{ of } V, 1, k \in S', i, j \in S.$$

The interest of these inequalities arises when $v_i^k = v_k^j = 1$. Since i and j are in S , and 1 and k are not, these constraints state that every directed cut that separates nodes $\{1, k\}$ from nodes $\{i, j\}$, contains at least two arcs (as variables x_{ij} are bounded by 1). Therefore, from Menger’s theorem for directed graphs (see Menger, 1927) we can conclude that there exist at least two arc-disjoint paths going from the set $\{1, k\}$ to the set $\{i, j\}$. The main point is that the variables v_i^k and v_k^j provide more information on these two paths. When the two variables are equal to 1, that is, the tour visits first node i , then node k and finally node j , we can conclude that the path corresponding to the part of the circuit going from node 1 to node i and the path corresponding to the part of the circuit going from node k to node j , must be arc-disjoint. These two paths correspond to the two arc-disjoint paths implied by inequalities (14). We will show how to explore this situation by creating a flow based formulation that results from combining two simpler flow systems, associated with each path, and by using adequate coupling constraints that explore the disjointness of the two paths. Then, by applying the reasoning above we obtain stronger flow based models that lead to new sets of cut-like inequalities involving the x_{ij} and v_i^j variables. These constraints can then be used in a cutting plane fashion to solve instances of the PCATSP (some results will be given in Section 5).

2 Recreating the GDDL inequalities

In this section we show how to recreate the GDDL inequalities by using the following 3-step procedure: i) we start with three sets of simple cut-like inequalities introduced in Gouveia and Pesneau (2006) and

obtain the corresponding LP equivalent flow-based models; ii) then, we develop new and stronger flow based systems that are obtained by deriving more complicated linking constraints that consider different flow systems simultaneously; and iii) we show that these more complicated flow models imply new sets of cut-like inequalities in the space of the x_{ij} and v_i^j variables, including the GDDL inequalities.

Two observations worth mentioning here are that this approach allows us to generate many other sets of inequalities besides the GDDL inequalities, and that the same 3-step procedure can be used to generate even more complicated inequalities if we start with more complicated sets of inequalities (as shown in the next section).

2.1 From simple cut inequalities and equivalent flow based models ...

Gouveia and Pesneau (2006) have presented several other exponentially-sized classes of cut-like inequalities that use the v variables, among them we consider the next three sets of related cut-like inequalities. They are based on the following observation.

Remark 1. *If $v_i^k = 1$ for two given nodes i and k , then*

1. *there exists a path from node 1 to node i not going through node k ,*
2. *there exists a path from node i to node k not going through node 1, and*
3. *there exists a path from node k to node 1 not going through node i .*

These three statements can be translated in terms of cut-like inequalities to get the following result.

Proposition 1 (Gouveia and Pesneau (2006)). *For all pairs of distinct nodes (i, k) of $V \setminus \{1\}$, the inequalities*

$$x(S', S) \geq v_i^k, \quad \forall \text{ partitions } (S', S) \text{ of } V \setminus \{k\}, 1 \in S', i \in S, \quad (15)$$

$$x(S', S) \geq v_i^k, \quad \forall \text{ partitions } (S', S) \text{ of } V \setminus \{1\}, i \in S', k \in S, \quad (16)$$

$$x(S', S) \geq v_i^k, \quad \forall \text{ partitions } (S', S) \text{ of } V \setminus \{i\}, k \in S', 1 \in S, \quad (17)$$

are valid for the PCATSP.

We note that these inequalities cannot play the role of the generic inequalities (3) since, alone, they do not guarantee subtour elimination. We also observe that constraints (8) or similar related inequalities are special cases of these inequalities. We obtain them by considering the special cases with single node sets either on the left-hand part of the cut or on the right-hand part, and then use the in-degree (1) or the out-degree (2) constraints. These three different families of cut-set inequalities together with the GDDL inequalities have been considered in the branch-and-cut developed by Gouveia and Pesneau (2006) and separated, in the usual way, by using max-flow computations.

Following the reasoning described in the previous section, we can translate inequalities (15)–(17) into equivalent flow systems as well. Given two nodes i and k , let $f_{pq}^{i,k}$ (resp. $g_{pq}^{i,k}$ and $h_{pq}^{i,k}$) be flow variables such that $f_{pq}^{i,k} = 1$ (resp. $g_{pq}^{i,k} = 1$ and $h_{pq}^{i,k} = 1$) when the path (or the flow) described in Remark 1.1. (resp. 1.2. and 1.3.) uses the arc $(p, q) \in A$; and $f_{pq}^{i,k} = 0$ (resp. $g_{pq}^{i,k} = 0$ and $h_{pq}^{i,k} = 0$) otherwise. For each pair of nodes $i, k \in V \setminus \{1\}$, $i \neq k$, consider the following flow systems.

$$\sum_{\substack{q:(1,q) \in A \\ q \neq 1, k}} f_{1q}^{i,k} = v_i^k, \quad (18)$$

$$\sum_{\substack{p:(p,i) \in A \\ p \neq i, k}} f_{pi}^{i,k} = v_i^k, \quad (19)$$

$$\sum_{\substack{q:(q,p) \in A \\ q \neq i, k, p}} f_{qp}^{i,k} = \sum_{\substack{q:(p,q) \in A \\ q \neq 1, k, p}} f_{pq}^{i,k}, \quad \forall p \in V \setminus \{1, i, k\}, \quad (20)$$

$$f_{pq}^{i,k} \leq x_{pq}, \quad \forall (p, q) \in A, p, q \neq k, \quad (21)$$

$$f_{pq}^{i,k} \in \{0, 1\}, \quad \forall (p, q) \in A, p, q \neq k, \quad (22)$$

$$\sum_{\substack{q:(i,q) \in A \\ q \neq 1,i}} g_{iq}^{i,k} = v_i^k, \quad (23)$$

$$\sum_{\substack{p:(p,k) \in A \\ p \neq 1,k}} g_{pk}^{i,k} = v_i^k, \quad (24)$$

$$\sum_{\substack{q:(q,p) \in A \\ q \neq 1,k,p}} g_{qp}^{i,k} = \sum_{\substack{q:(p,q) \in A \\ q \neq 1,i,p}} g_{pq}^{i,k}, \quad \forall p \in V \setminus \{1, i, k\}, \quad (25)$$

$$g_{pq}^{i,k} \leq x_{pq}, \quad \forall (p, q) \in A, p, q \neq 1, \quad (26)$$

$$g_{pq}^{i,k} \in \{0, 1\}, \quad \forall (p, q) \in A, p, q \neq 1, \quad (27)$$

$$\sum_{\substack{q:(k,q) \in A \\ q \neq i,k}} h_{kq}^{i,k} = v_i^k, \quad (28)$$

$$\sum_{\substack{p:(p,1) \in A \\ p \neq 1,i}} h_{p1}^{i,k} = v_i^k, \quad (29)$$

$$\sum_{\substack{q:(q,p) \in A \\ q \neq 1,i,p}} h_{qp}^{i,k} = \sum_{\substack{q:(p,q) \in A \\ q \neq i,k,p}} h_{pq}^{i,k}, \quad \forall p \in V \setminus \{1, i, k\}, \quad (30)$$

$$h_{pq}^{i,k} \leq x_{pq}, \quad \forall (p, q) \in A, p, q \neq i, \quad (31)$$

$$h_{pq}^{i,k} \in \{0, 1\}, \quad \forall (p, q) \in A, p, q \neq i. \quad (32)$$

In terms of the corresponding LP relaxations, and as pointed out before, by using the max-flow/min-cut theorem, these three flow systems (without the integrality constraints on the flows) are equivalent to the three sets of cut inequalities (15)–(17), respectively.

In the following subsection we show how to obtain more general linking constraints, and in particular we show how to revisit the disjointness feature of the GDDL inequalities.

2.2 ... to more complicated linking constraints and the GDDL inequalities again

In this subsection we discuss more complicated flow based models that can be obtained by using linking constraints that combine different flow systems either for the same pair of precedences or for different ones. We observe that some of the cases presented next are similar to cases presented before in the literature and adequate references will be provided later on. However, we also point out that perhaps a less obvious, and different – as far as we know – relation is also proposed which is precisely the one that leads to the GDDL inequalities, which is the main motivation of this work. We will start with this one since it is the main topic of this subsection.

2.2.1 Generating the GDDL inequalities again

The following set of linking constraints revisits the disjoint paths property mentioned for the GDDL inequalities, and its validity follows from the fact that an arc (p, q) cannot be at the same time in the path from 1 to i (without node k) and in the path starting from k to node j (without node 1) whenever v_i^k and v_k^j are both equal to 1.

$$f_{pq}^{i,k} + g_{pq}^{k,j} \leq x_{pq}, \quad \forall (p, q) \in A, \forall \text{ sequences } (i, k, j) \text{ of distinct nodes of } V \setminus \{1\}. \quad (33)$$

The model composed of the f and g flow formulations plus constraints (33) will be denoted by (f, g) -disjoint path model in the remainder of the text. We observe that we will omit the name of the precedences on the f , g and h variables whenever they are clear from the context.

Note that the (f, g) -disjoint path model implies the simple cuts (15) and (16), since the linking constraints (33) imply the linking constraints (21) and (26). We show next that the (f, g) -disjoint path model

also implies the GDDL inequalities. This type of proof may be used to generate other inequalities as we shall point out later on. It is based on the following observation.

Remark 2. Let i and j be two distinct nodes of $V \setminus \{1\}$ and S and S' be two node sets such that $x(S', S) \geq v_i^j$ defines a simple cut inequality (15) (resp. (16), and (17)). Adding equations (19) and (20) (resp. (24) and (25), and (29) and (30)) for the nodes of S and using $f_{pq}^{i,j} \geq 0$ (resp. $g_{pq}^{i,j} \geq 0$ and $h_{pq}^{i,j} \geq 0$) we get $f(S', S) \geq v_i^j$ (resp. $g(S', S) \geq v_i^j$ and $h(S', S) \geq v_i^j$), that are the simple cut inequalities written on the corresponding flow variables.

Proposition 2. The GDDL inequalities are implied by the LP relaxation of the (f, g) -disjoint path model.

Proof. Consider the partition $(\{1\}, \{k\}, \{i\}, \{j\}, T, T')$ of V . From Remark 2, we get the two constraints:

$$f(\{1\} \cup T, T' \cup \{i\} \cup \{j\}) \geq v_i^k$$

and

$$g(\{k\} \cup T, T' \cup \{j\} \cup \{i\}) \geq v_k^j.$$

By adding these two inequalities and by using (33) we obtain

$$x(\{1\} \cup \{k\} \cup T, T' \cup \{i\} \cup \{j\}) \geq v_i^k + v_k^j,$$

which is the GDDL (14) for $S = T' \cup \{i\} \cup \{j\}$. □

Observe that the generic GDDL inequality obtained in the previous proof is stronger than the inequality obtained by adding the two simple cuts that result from using (18)–(20) and (23)–(25) together with (21) and (26) respectively.

It is still open whether the (f, g) -disjoint path model is stronger than the model defined by the GDDL inequalities and the simple cut inequalities (15) and (16) or not, although we conjecture that the models are equivalent, that is, the simple cuts and the GDDL inequalities are the only interesting sets of inequalities that can be derived from the (f, g) -disjoint path model.

The fact that the (f, g) -disjoint path model implies the GDDL inequalities, guarantees that this model can be used in place of (3) to derive a valid model for the ATSP. Note, however, that such a model is far from easy to use since it contains $O(n^5)$ linking constraints.

Observe also that by using the flow system on g and h together with the linking constraints

$$g_{pq}^{i,k} + h_{pq}^{k,j} \leq x_{pq}, \quad \forall (p, q) \in A, \forall \text{ sequences } (i, k, j) \text{ of distinct nodes of } V \setminus \{1\}, \quad (34)$$

we get the reversed GDDL inequalities

$$\begin{aligned} x(S', S) &\geq v_i^k + v_k^j, \quad \forall \text{ sequences } (i, j, k) \text{ of distinct nodes of } V \setminus \{1\}, \\ &\forall \text{ partitions } (S', S) \text{ of } V, i, j \in S', 1, k \in S. \end{aligned} \quad (35)$$

These inequalities can be obtained from the GDDL inequalities (14) by applying adequately the assignment constraints (1) or (2), and thus they are redundant. However, as we will see in Section 4, this second combination of flow systems will give us a second way of strengthening the GDDL inequalities for the precedence constrained case.

2.2.2 Generating other sets of constraints

As pointed out in the introduction to this section, other linking constraints can be used in the context of the f , g and h models. In this subsection we make a brief reference to two other, perhaps more straightforward sets, and leave a more complete study for a different occasion. The first set results from the observation that whenever $v_i^k = 1$ for two given nodes i and k , then the three paths of Remark 1 should be disjoint. Thus we can replace the linking constraints (21), (26) and (31) by the following stronger set.

$$f_{pq}^{i,k} + g_{pq}^{i,k} + h_{pq}^{i,k} \leq x_{pq}, \quad \forall (p, q) \in A, \forall i, k \in V \setminus \{1\}, i \neq k. \quad (36)$$

A similar set of constraints has been presented for 2-terminal Steiner tree formulations (see, e.g., Ball et al., 1989). The new inequalities (36) permit us to derive cut-like inequalities that are stronger versions of inequalities that are obtained by adding two, or three adequate simple cuts (15)–(17). A few computational results obtained with the f , g and h model with these constraints show that, in general, the LP bound does not improve much, or more precisely, does not improve significantly to justify spending time to make the three flow model “more attractive” to use or to develop separation routines for the projected inequalities. Also, since the main purpose of this paper is to study the disjoint paths property for the two different precedences implicit in the GDDL inequalities, we omit the study of the effect of the stronger inequalities (36) from this paper.

With the g systems alone we can provide linking constraints that combine different precedences in a 2-path strategy as follows

$$g_{pq}^{i,k} + g_{pq}^{k,j} \leq x_{pq}, \quad \forall (p, q) \in A, \forall \text{ sequences } (i, k, j) \text{ of distinct nodes of } V \setminus \{1\}. \quad (37)$$

Their validity is easy to prove and is again based on the observation that the paths associated with the precedences on the left-hand side of the inequalities must be disjoint. Observe, however, that a more general inequality of the type

$$g_{pq}^{i,k} + g_{pq}^{k,j} + g_{pq}^{j,l} + \dots \leq x_{pq}, \quad \forall (p, q) \in A, \forall \text{ sequences } (i, k, j, l, \dots) \text{ of distinct nodes of } V \setminus \{1\},$$

is not necessarily valid unless the v variables associated with the several precedences are equal to one. Such path chain inequalities have been proposed in Letchford and Salazar-González (2016) and proved to be very effective for the case when there are “paths” of v_i^j variables equal to 1.

3 Creating new inequalities

In the previous section, we have shown how to recreate the GDDL inequalities from the simple cut inequalities by using a 3-step procedure. In this section we use a similar 3-step procedure but starting with a more complicated set of cut-like inequalities, the so-called 2-path inequalities. At the end of the procedure we will obtain inequalities that include more v variables on the right-hand side and that might be viewed as generalizations of the GDDL inequalities.

3.1 From 2-path cut inequalities and corresponding flow based models ...

Many other sets of cut-like inequalities were introduced in Gouveia and Pesneau (2006). For example consider the k -path cuts that generalize the simple cuts of the previous section:

Let (i_1, \dots, i_{k+1}) be a sequence of $k+1 \leq n-2$ distinct nodes of $V \setminus \{1\}$. Suppose that $i_0 = i_{k+2} = 1$. Let $r \in \{0, \dots, k+1\}$. The k -path cut inequalities are given by

$$x(S', S) \geq v_{i_1}^{i_2} + \dots + v_{i_k}^{i_{k+1}} - k + 1, \quad \forall \text{ partitions } (S', S) \text{ of } V \setminus (\{i_0, \dots, i_{r-1}\} \cup \{i_{r+2}, \dots, i_{k+1}\}), \quad (38)$$

$$i_r \in S', i_{r+1} \in S.$$

Below we describe the special case of the k -path cuts inequalities when $k = 2$, since they will be needed for the remainder of this section. Let (i, j, k) be a sequence of three distinct nodes of $V \setminus \{1\}$. The four sets of 2-path cut inequalities (depending on the choice of the node i_r) associated with these three nodes are given by

$$x(S', S) \geq v_i^j + v_j^k - 1, \quad \forall \text{ partitions } (S', S) \text{ of } V \setminus \{j, k\}, 1 \in S', i \in S, \quad (39)$$

$$x(S', S) \geq v_i^j + v_j^k - 1, \quad \forall \text{ partitions } (S', S) \text{ of } V \setminus \{k, 1\}, i \in S', j \in S, \quad (40)$$

$$x(S', S) \geq v_i^j + v_j^k - 1, \quad \forall \text{ partitions } (S', S) \text{ of } V \setminus \{1, i\}, j \in S', k \in S, \quad (41)$$

$$x(S', S) \geq v_i^j + v_j^k - 1, \quad \forall \text{ partitions } (S', S) \text{ of } V \setminus \{i, j\}, k \in S', 1 \in S. \quad (42)$$

Consider, for instance, inequalities (39). These inequalities simply state that when i is before j and j is before k , there must exist a path from 1 to i not passing through j and k . A similar interpretation holds for the remaining three 2-path inequalities.

As also noted in Gouveia and Pesneau (2006), these four sets of inequalities can be separated in polynomial time by enumerating the triplets of nodes (i, j, k) and, for each triplet with $v_i^j + v_j^k - 1$ positive, by computing a minimum cut in the adequate graph. This separation routine suggests that these sets of inequalities can be replaced by equivalent flow based systems as observed in the previous section for the other models. However, there is a slight difference between these new flow systems and the ones presented in the previous section. Here, the amount of flow sent from the origin to the destination must be equal to $\max\{0, v_i^j + v_j^k - 1\}$ since the second quantity can be negative. Thus the flow conservation constraints for the source and the destination node must be changed to “ \geq ” inequalities.

Let $a_{pq}^{i,j,k}$ (resp. $b_{pq}^{i,j,k}$, $c_{pq}^{i,j,k}$, and $d_{pq}^{i,j,k}$) be flow variables such that $a_{pq}^{i,j,k} = 1$ (resp. $b_{pq}^{i,j,k} = 1$, $c_{pq}^{i,j,k} = 1$, and $d_{pq}^{i,j,k} = 1$) when the arc $(p, q) \in A$ belongs to the path from 1 to i (resp. from i to j , from j to k , and from k to 1) in the graph without nodes j and k (resp. nodes k and 1, nodes 1 and i , and nodes i and j); and $a_{pq}^{i,j,k} = 0$ (resp. $b_{pq}^{i,j,k} = 0$, $c_{pq}^{i,j,k} = 0$, and $d_{pq}^{i,j,k} = 0$) otherwise. We then get, for all triplets of distinct nodes $i, j, k \in V \setminus \{1\}$, the following four flow systems (we omit here the complete description of the flows on the b , c and d variables to lighten the paper):

$$\sum_{\substack{q:(1,q) \in A, \\ q \neq 1, j, k}} a_{1q}^{i,j,k} \geq v_i^j + v_j^k - 1, \quad (43)$$

$$\sum_{\substack{p:(p,i) \in A, \\ p \neq i, j, k}} a_{pi}^{i,j,k} \geq v_i^j + v_j^k - 1, \quad (44)$$

$$\sum_{\substack{q:(q,p) \in A, \\ q \neq i, j, k, p}} a_{qp}^{i,j,k} = \sum_{\substack{q:(p,q) \in A, \\ q \neq 1, j, k, p}} a_{pq}^{i,j,k}, \quad \forall p \in V \setminus \{1, i, j, k\}, \quad (45)$$

$$a_{pq}^{i,j,k} \leq x_{pq}, \quad \forall (p, q) \in A, p, q \notin \{j, k\}, \quad (46)$$

$$a_{pq}^{i,j,k} \in \{0, 1\}, \quad \forall (p, q) \in A, p, q \notin \{j, k\}, \quad (47)$$

$$\text{Flow system on } b \text{ sending } \max\{0, v_i^j + v_j^k - 1\} \text{ units of flow} \quad (48)$$

from node i to node j in the graph without nodes k and 1,

$$b_{pq}^{i,j,k} \leq x_{pq}, \quad \forall (p, q) \in A, p, q \notin \{1, k\}, \quad (49)$$

$$b_{pq}^{i,j,k} \in \{0, 1\}, \quad \forall (p, q) \in A, p, q \notin \{1, k\}, \quad (50)$$

$$\text{Flow system on } c \text{ sending } \max\{0, v_i^j + v_j^k - 1\} \text{ units of flow} \quad (51)$$

from node j to node k in the graph without nodes 1 and i ,

$$c_{pq}^{i,j,k} \leq x_{pq}, \quad \forall (p, q) \in A, p, q \notin \{1, i\}, \quad (52)$$

$$c_{pq}^{i,j,k} \in \{0, 1\}, \quad \forall (p, q) \in A, p, q \notin \{1, i\}, \quad (53)$$

$$\text{Flow system on } d \text{ sending } \max\{0, v_i^j + v_j^k - 1\} \text{ units of flow} \quad (54)$$

from node k to node 1 in the graph without nodes i and j ,

$$d_{pq}^{i,j,k} \leq x_{pq}, \quad \forall (p, q) \in A, p, q \notin \{i, j\}, \quad (55)$$

$$d_{pq}^{i,j,k} \in \{0, 1\}, \quad \forall (p, q) \in A, p, q \notin \{i, j\}. \quad (56)$$

3.2 ... to more complicated linking constraints and new inequalities in the space of the x and v variables

In this subsection we propose new linking constraints that combine some of these new flow systems together, or combine them with some of the flows on the variables f , g and h presented in Section 2. Then, we show

that the combined flow systems imply new sets of cut-like inequalities. We also give some intuition on how the new inequalities differ from the k -path cuts, at least for $k = 2$, $k = 3$ and $k = 4$.

Several sets of valid constraints linking two flows can be defined (as long as the precedences share some nodes). We made an extensive study of all possible combinations and have concluded that most of these combinations lead to cut-like inequalities similar to the ones presented below although redundant, since they are implied by the GDDL inequalities (for simplicity we omit the complete analysis from this paper). However, there are some interesting exceptions which will be described in the remainder of this subsection.

Consider the following set of linking constraints

$$g_{pq}^{i,j} + c_{pq}^{j,r,s} \leq x_{pq}, \quad \forall (p, q) \in A, \forall \text{ sequences } (i, j, r, s) \text{ of distinct nodes of } V \setminus \{1\}, \quad (57)$$

that combines the new flow system c with the previous flow system g . The flow on the $g^{i,j}$ variables models the path from i to j (without node 1) when $v_i^j = 1$, and the flow on variables $c^{j,r,s}$ models the path from r to s (without nodes 1 and j) when both v_r^j and v_r^s are equal to 1. Observe that when all these v variables are equal to 1, these two paths must be disjoint, thus the new linking inequalities are valid. By using a reasoning similar to the derivation used in Proposition 2, we can show that the model obtained by combining the flows g and c with the linking constraints (57), henceforth called the (g, c) -disjoint path model, implies the following set of inequalities.

Proposition 3. *For all sequences (i, j, r, s) of four distinct nodes of $V \setminus \{1\}$, inequalities*

$$x(S', S) \geq v_i^j + v_j^r + v_r^s - 1, \quad \forall \text{ partitions } (S', S) \text{ of } V \setminus \{1\}, i, r \in S', j, s \in S, \quad (58)$$

are implied by the LP relaxation of the (g, c) -disjoint path model.

Note that these inequalities correspond to cut-like inequalities defined in the graph without node 1. We observe that we could have obtained the same set of inequalities by combining the flows $b^{i,j,r}$ and $g^{r,s}$ and defining linking constraints similar to constraints (57). As for the two ‘‘symmetric’’ GDDL (14) and (35), this second way of obtaining these inequalities will show itself meaningful in Section 4. Again, these inequalities can be separated in polynomial time by computing a maximum flow in an adequate graph for each sequence of four nodes (i, j, r, s) .

Similarly, by using, for all sequences (i, j, r, s) of distinct nodes of $V \setminus \{1\}$, the following linking constraints

$$b_{pq}^{i,j,r} + h_{pq}^{r,s} \leq x_{pq}, \quad \forall (p, q) \in A, \quad (59)$$

$$f_{pq}^{i,j} + c_{pq}^{j,r,s} \leq x_{pq}, \quad \forall (p, q) \in A, \quad (60)$$

(we skip the argument for their validity but it is similar to the one used for proving the validity of the last set) and using the corresponding flow systems to define the (b, h) -disjoint path model and the (f, c) -disjoint path model, respectively, we obtain the following inequalities:

Proposition 4. *For all sequences (i, j, r, s) of four distinct nodes of $V \setminus \{1\}$, inequalities*

$$x(S', S) \geq v_i^j + v_j^r + v_r^s - 1, \quad \forall \text{ partitions } (S', S) \text{ of } V \setminus \{r\}, i, s \in S', j, 1 \in S, \quad (61)$$

and

$$x(S', S) \geq v_i^j + v_j^r + v_r^s - 1, \quad \forall \text{ partitions } (S', S) \text{ of } V \setminus \{j\}, 1, r \in S', i, s \in S, \quad (62)$$

are implied by the LP relaxations of the (b, h) -disjoint path model and the (f, c) -disjoint path model, respectively.

Finally, we consider a set of linking constraints that combines two flow systems derived from 2-path cut inequalities. In fact, our analysis shows that only a single case is of interest (the other ones lead to redundant inequalities), which is the one obtained by considering, for all sequences (i, j, k, r, s) of distinct nodes of $V \setminus \{1\}$, the following linking constraints and the corresponding flow systems:

$$b_{pq}^{i,j,k} + c_{pq}^{k,r,s} \leq x_{pq}, \quad \forall (p, q) \in A. \quad (63)$$

As before, we can show that the model defined by the b and c flows together with the linking constraints (63), that is, the (b, c) -disjoint path model, implies the following set of cut inequalities defined in a graph without two nodes.

Proposition 5. For all sequences (i, j, k, r, s) of five distinct nodes of $V \setminus \{1\}$, inequalities

$$x(S', S) \geq v_i^j + v_j^k + v_k^r + v_r^s - 2, \quad \forall \text{ partitions } (S', S) \text{ of } V \setminus \{1, k\}, i, r \in S', j, s \in S, \quad (64)$$

are implied by the LP relaxation of the (b, c) -disjoint path model.

These inequalities can still be separated in polynomial time. However, the enumeration on all the sequences of five nodes leads to huge CPU times.

The cut-like inequalities involving the x and v variables discussed in this paper can, until now, be classified in two sets. The inequalities in the first set are the k -path cut inequalities, which include the simple cuts (15)–(17) and the 2-path cut inequalities (39)–(42). These inequalities model a path going from a source node to a destination node, and not going through a given set of nodes. The inequalities in the second set are GDDL-like inequalities that model the disjointness of two paths. This second class of inequalities contains the GDDL inequalities (14) and also inequalities (58), (61), (62), and (64).

Another possible classification can be made by considering the number of v variables on the right-hand side of the inequalities (in other words, considering the length of the path induced by the precedences associated with the v variables). In this case, we can divide the inequalities into three sets:

- The 2-path cut inequalities (39)–(42) and the GDDL inequalities (14), these two sets both involving two v variables.
- The 3-path cut inequalities and inequalities (58), (61) and (62), involving three v variables.
- The 4-path cut inequalities and inequalities (64), involving four v variables.

The following table summarizes these two classifications that have been given (for the moment ignore the last line).

	1 v	2 v 's	3 v 's	4 v 's
k -path cuts	simple cuts (15)-(17) $\geq v_i^j$	2 path-cuts (39)-(42) $\geq v_i^j + v_j^k - 1$	3-path cuts $\geq v_i^j + v_j^k + v_k^r - 2$	4-path cuts $\geq v_i^j + v_j^k + v_k^r + v_r^s - 3$
GDDL-like		GDDL (14) $\geq v_i^j + v_j^k$	Inequalities (58), (61), (62) $\geq v_i^j + v_j^k + v_k^r - 1$	Inequalities (64) $\geq v_i^j + v_j^k + v_k^r + v_r^s - 2$
Stronger				Inequalities (65) $\geq v_i^j + v_j^k + v_k^r + v_r^s - 1$

An interesting observation about these three sets of inequalities (corresponding to the last three columns in the table) is concerned with the difference, inside each class, between the k -path cut inequalities and the GDDL-like inequalities. In fact, if we start from the k -path cut inequalities we observe that the GDDL-like inequalities have: i) two less nodes outside of the cut, each one is brought in on each side of the cut; and ii) as a sort of compensation, 1 unit is added to the right-hand side. As one example, with respect to inequality (41), we can “move” node 1 inside S' and node i inside S and “correct” the right-hand side by adding 1 unit and obtain the GDDL (14).

Clearly, the k -path cut inequalities and the GDDL-like inequalities inside each class are unrelated but, as we will see in Section 5, from a computational point of view, the set of GDDL-like inequalities is stronger. In fact, the GDDL inequalities (14) have a much greater effect on the LP bounds than the 2-path cut inequalities. We observe the same when comparing the 3-path cuts with inequalities (58), (61), and (62), since adding the new inequalities (58), (61), and (62) leads to a significant improvement of the LP bound, while adding the 3-path cut inequalities only leads to a negligible improvement of the LP bound. For this reason we have not considered the 4-path cut inequalities in the computational comparison.

An analysis, by column, of the sets of inequalities in the table combined with the fact that the inequalities with more nodes inside the cut (even with the correction on the right-hand side) lead to a more effective set of inequalities, motivated us to suggest the following new set of inequalities (shown in the last line of the table). This new set of inequalities also appears to be a straightforward generalization of the GDDL inequalities. We conjecture that they may be obtained by combining the flow associated with a GDDL with the flow associated with a 2-path inequality and guaranteeing that the three induced paths should be disjoint, but the validity of the linking constraints is not obvious. They are defined as follows.

Proposition 6. For all sequences (i, j, k, r, s) of five distinct nodes of $V \setminus \{1\}$, inequalities

$$x(S', S) \geq v_i^j + v_j^k + v_k^r + v_r^s - 1, \quad \forall \text{ partitions } (S', S) \text{ of } V, 1, j, r \in S', i, k, s \in S. \quad (65)$$

are valid for the PCATSP.

Proof. The validity of the inequalities is easy to see by considering all cases: If all the v variables $v_i^j, v_j^k, v_k^r, v_r^s$ are equal to 1, the three paths from 1 to i , from j to k and from r to s should be disjoint; If at most one of these variables is equal to 1, the inequalities become trivial; If exactly two of these variables are equal to 1, the inequalities correspond to cut constraints (6).

Thus, suppose that exactly three of the variables $v_i^j, v_j^k, v_k^r, v_r^s$ are equal to 1 and the fourth one is equal to 0. Suppose that v_i^j or v_j^k (resp. v_k^r or v_r^s) is equal to 0. Then $v_k^r = v_r^s = 1$ (resp. $v_i^j = v_j^k = 1$) and the paths from 1 to k , and from r to s (resp. from 1 to i , and from j to k) should be disjoint. \square

Again we observe that the “difference” between these inequalities and the inequalities (64) is similar (but not identical) to the difference between inequalities (64) and the 4-path inequalities. Computational results taken from a few instances show that this last set of inequalities may lead to substantial improvements on the lower bounds, however, the CPU times are prohibitively high.

4 Rules for strengthening inequalities according to the precedence constraints

The inequalities introduced in the previous sections were described without assuming any knowledge about the precedence constraints that are given in B . However, as pointed out in Gouveia and Ruthmair (2015) and Letchford and Salazar-González (2016), significant improvements can be obtained in terms of LP bounds if such information is used. In this section we show how to use such information to strengthen the inequalities presented.

As an example, let us first consider the simple cut inequalities (16). If i and j are two distinct nodes of $V \setminus \{1\}$, these inequalities state that there must exist a path from i to j not going through node 1 when $v_i^j = 1$. Suppose that there exists a node q such that $(j, q) \in B$, meaning that j must be before q in the tour. Consequently, q cannot belong to the path from i to j and can be removed from the sets defining the corresponding cut. Similarly, every node p such that $(p, i) \in B$ can also be removed from the cut since p has to be before i in the tour.

Let us denote by $N_B^+(i)$ the set of nodes $q \in V \setminus \{1\}$ such that $(i, q) \in B$ and by $N_B^-(i)$ the set of nodes $p \in V \setminus \{1\}$ such that $(p, i) \in B$. Following the previous observation, we can strengthen inequalities (16) in the following way.

Proposition 7. *Inequalities*

$$x(S', S) \geq v_i^k, \quad \forall i, k \in V \setminus \{1\}, i \neq k, \quad (66)$$

$$\forall \text{ partitions } (S', S) \text{ of } V \setminus (\{1\} \cup N_B^-(i) \cup N_B^+(k)), i \in S', k \in S,$$

are valid for the PCATSP.

Similarly, inequalities (15) and (17) can be strengthened as follows.

Proposition 8. *Inequalities*

$$x(S', S) \geq v_i^k, \quad \forall i, k \in V \setminus \{1\}, i \neq k, \quad (67)$$

$$\forall \text{ partitions } (S', S) \text{ of } V \setminus (\{k\} \cup N_B^+(i) \cup N_B^-(k)), 1 \in S', i \in S,$$

$$x(S', S) \geq v_i^k, \quad \forall i, k \in V \setminus \{1\}, i \neq k, \quad (68)$$

$$\forall \text{ partitions } (S', S) \text{ of } V \setminus (\{i\} \cup N_B^-(i) \cup N_B^-(k)), k \in S', 1 \in S,$$

are valid for the PCATSP.

Clearly, the separation of these strengthened inequalities is still polynomial, since it can be done by performing maximum flow computations in the graph obtained by removing the node sets described in the previous two propositions.

Consider, now, the case of the 2-path cut inequalities (39)–(42). A similar reasoning leads to the following strengthening.

Proposition 9. *For all sequences (i, j, k) of distinct nodes of $V \setminus \{1\}$, inequalities*

$$x(S', S) \geq v_i^j + v_j^k - 1, \quad \forall \text{ partitions } (S', S) \text{ of } V \setminus (\{j, k\} \cup N_B^+(i) \cup N_B^+(j) \cup N_B^+(k)), \quad 1 \in S', i \in S, \quad (69)$$

$$x(S', S) \geq v_i^j + v_j^k - 1, \quad \forall \text{ partitions } (S', S) \text{ of } V \setminus (\{1, k\} \cup N_B^-(i) \cup N_B^+(j) \cup N_B^+(k)), \quad i \in S', j \in S, \quad (70)$$

$$x(S', S) \geq v_i^j + v_j^k - 1, \quad \forall \text{ partitions } (S', S) \text{ of } V \setminus (\{1, i\} \cup N_B^-(i) \cup N_B^-(j) \cup N_B^+(k)), \quad j \in S', k \in S, \quad (71)$$

$$x(S', S) \geq v_i^j + v_j^k - 1, \quad \forall \text{ partitions } (S', S) \text{ of } V \setminus (\{i, j\} \cup N_B^-(i) \cup N_B^-(j) \cup N_B^-(k)), \quad k \in S', 1 \in S, \quad (72)$$

are valid for the PCATSP.

As it was the case for the simple cut inequalities, these enhanced versions of the 2-path inequalities can also be separated in a similar manner. The question, now, is whether we could do similar enhancements for the GDDL inequalities (14), inequalities (58), (61) and (62) and inequalities (64). In such cases, it is not clear how to use a direct reasoning as above to eliminate nodes from the subsets defining the cuts.

Now, we present a rule that is based on the procedure for creating these inequalities from the flow based systems and which allows us to eliminate nodes from the sets defining the corresponding cuts.

We illustrate this rule with the GDDL inequalities (14). Let i, k and j be three distinct nodes of $V \setminus \{1\}$. As we have seen in Section 2.2.1, these inequalities can be obtained by combining the f flow system for precedence (i, k) and the g flow system for precedence (k, j) . Also, as seen before, these two flow systems correspond to the feasibility problem associated with the separation of the simple cut (15) for precedence (i, k) and the simple cut (16) for precedence (k, j) .

Instead of considering the flow systems associated with the original simple cut inequalities, we consider, now, the flow systems associated with the strengthened versions that we have just presented before. The flow system corresponding to inequalities (67) for precedence (i, k) is defined on the graph where the nodes in $\{k\} \cup N_B^+(i) \cup N_B^+(k)$ are removed. Similarly the flow system corresponding to inequalities (66) for precedence (k, j) is defined on the graph where the nodes in $\{1\} \cup N_B^-(k) \cup N_B^+(j)$ are removed.

Consider, now, that $v_i^k = v_k^j = 1$. This means that the paths from 1 to i and from k to j must be disjoint. A node that has been removed in both previous flow systems cannot belong to any of these two paths, and thus, any arc incident to such a node can be removed from the GDDL inequality. Consequently, the cut (S', S) defining the GDDL inequalities (14) can be defined on the graph where the nodes of $(\{k\} \cup N_B^+(i) \cup N_B^+(k)) \cap (\{1\} \cup N_B^-(k) \cup N_B^+(j))$ are removed.

That is, the following strengthened GDDL inequalities can be directly obtained from the combination of the two flows associated with the strengthened simple cuts (66) and (67) with the linking constraints (33). To simplify, let $N_B^f(i, k)$ be the set of nodes to be removed from the cut-set in inequalities (67), that is, $N_B^f(i, k) = \{k\} \cup N_B^+(i) \cup N_B^+(k)$. Similarly, define $N_B^g(i, k)$ and $N_b^h(i, k)$ for inequalities (66) and (68), respectively.

Proposition 10. *For all sequences (i, j, k) of distinct nodes of $V \setminus \{1\}$, inequalities*

$$x(S', S) \geq v_i^k + v_k^j, \quad \forall \text{ partitions } (S', S) \text{ of } V \setminus (N_B^f(i, k) \cap N_B^g(k, j)), \quad 1, k \in S', i, j \in S, \quad (73)$$

are valid for the PCATSP.

In Section 2.2.1 we have observed that the GDDL inequalities can also be derived by combining the g flows with the h flows. More precisely, the reversed GDDL inequalities (35) can be obtained by combining

the appropriate g and h flows and can be shown to be equivalent, under the assignment constraints, to the GDDL inequalities derived by combining f with g . However, for the PCATSP, the rule given above with the g and h flows leads to inequalities that are, in general, not equivalent to the inequalities derived by combining f with g , and thus we will consider the two sets in our computational results.

This “intersection” argument, that is, intersecting the subsets of nodes associated with each flow used to derive a new inequality can easily be seen to be valid to obtain stronger versions of the inequalities (58), (61), (62), and (64). Let $N_B^a(i, j, k)$, $N_B^b(i, j, k)$, $N_B^c(i, j, k)$, and $N_B^d(i, j, k)$ be the sets of nodes to be removed from the cut-set in inequalities (39)–(42), respectively. We have the following result:

Proposition 11.

a) For all sequences (i, j, r, s) of distinct nodes of $V \setminus \{1\}$, inequalities

$$x(S', S) \geq v_i^j + v_j^r + v_r^s - 1, \quad \forall \text{ partitions } (S', S) \text{ of } V \setminus (N_B^g(i, j) \cap N_B^c(j, r, s)), \quad i, r \in S', \quad j, s \in S, \quad (74)$$

$$x(S', S) \geq v_i^j + v_j^r + v_r^s - 1, \quad \forall \text{ partitions } (S', S) \text{ of } V \setminus (N_B^b(i, j, r) \cap N_B^h(r, s)), \quad i, s \in S', \quad j, 1 \in S, \quad (75)$$

$$x(S', S) \geq v_i^j + v_j^r + v_r^s - 1, \quad \forall \text{ partitions } (S', S) \text{ of } V \setminus (N_B^f(i, j) \cap N_B^c(j, r, s)), \quad 1, r \in S', \quad i, s \in S, \quad (76)$$

are valid for the PCATSP.

b) For all sequences (i, j, k, r, s) of distinct nodes of $V \setminus \{1\}$, inequalities

$$x(S', S) \geq v_i^j + v_j^k + v_k^r + v_r^s - 2, \quad \forall \text{ partitions } (S', S) \text{ of } V \setminus (N_B^b(i, j, k) \cap N_B^c(k, r, s)), \quad i, r \in S', \quad j, s \in S, \quad (77)$$

are valid for the PCATSP.

For simplicity we omit the proof of the validity of these enhanced versions of inequalities from this paper.

We observe that, as noted in Section 3.2, inequalities (58) can also be obtained from combining appropriate b and g flow systems, which, under this new rule, define a different set of inequalities than (74), similarly to the relation between the GDDL and the reversed GDDL inequalities.

We conclude this section by emphasizing again the relevance of the procedure described to eliminate nodes from the subsets defining the cuts. That is, even if the new inequalities were known before (and their validity also proved), the indirect way of producing them by combining flows allows us to eliminate nodes, something that would not have been clear by using a direct reasoning (as in the case of the GDDL inequalities). Our computational results will show the benefits of eliminating nodes from the inequalities. In fact, since inequalities (65) have not been generated in this way, it is not clear how to find a corresponding set of nodes to eliminate. In our opinion, significant improvements and also reductions in the computation times would be obtained by providing such a rule for this last set of inequalities.

5 Computational results

In this section we provide some insight on how the new inequalities, and the framework for deriving them, might prove to be useful for solving the PCATSP. We observe that this study is exploratory and not conclusive in the sense that in our cutting plane method, although the separation for each inequality is done in a fairly good standard way, the inequalities are added to the LP models in a straightforward way with no study, yet, on dominance between several different sets neither on heuristics to separate only a subset of them.

The computational study was performed on an Intel(R) Core(TM) i7-4790 @3.60GHz, 8GB RAM, and using CPLEX 12.6.1’s Concert Technology for C++. To compute the LP bounds we use a single thread

and deactivate CPLEX cuts, preprocessing, and heuristics. All other CPLEX settings are default. In this study we have used some instances for the sequential ordering problem from the TSPLIB library, namely the instances p43.1-4 and ry48p.1-4, which are instances with asymmetric costs. With exception for the instances ry48p.2 and ry48p.3, all of them have been solved by state-of-the-art methods (Cire and van Hoes, 2013; Gouveia and Ruthmair, 2015). We have also randomly generated four additional sets of instances for the sequential ordering problem, with 25 or 35 nodes and symmetric random or Euclidean costs, based on the method used to generate the instances in Letchford and Salazar-González (2016): we randomly place n points in a 500×500 grid and set the costs c_{ij} to the Euclidean distance between i and j (instance set ejj) or to a random number in $[0, 500]$ (instance set rjj). Precedence relations B are created by iteratively choosing random nodes i and j such that $1 < i < j < n$. In case a new relation leads to a cycle of precedence relations, we skip it and create a new precedence pair. The instance sets were named rjj25.1-6 and rjj35.1-5 in case of random symmetric costs, and ejj25.1-6 and ejj35.1-5 in the case of Euclidean costs.

The separation of any of the inequalities described throughout this work can be done in polynomial time, since the separation algorithms are straightforward adaptations of the max-flow/min-cut separation procedure used to separate subtour elimination constraints. For example, in order to separate the GDDL inequalities (14), we compute a max-flow (and hence a min-cut) for every possible triplet of distinct nodes $i, j, k \in V \setminus \{1\}$, such that 1 and k are fixed to the subset S' and i and j are fixed to the subset S , by connecting them with large-capacity arcs to an artificial flow source and sink, respectively. This max-flow must be computed in an adequate graph where some nodes are eliminated depending on the considered inequalities. With respect to the other more “complicated” inequalities, the procedure differs only in the number of nodes to be fixed on each side of the cut, and also on the nodes to be removed from the graph before computing the max-flow. Note that the strengthened inequalities from the previous section are separated in the same way, but now more nodes are to be removed from the auxiliary graph. These separation procedures are used in a cutting plane algorithm which follows an incremental approach in the sense that if the separation procedure for a set A of inequalities is less computationally expensive than the separation procedure for a set B of inequalities, then the inequalities in set A are separated first (e.g., inequalities (39)–(42) are separated before inequalities (58) + (61) + (62)).

Tables 1 and 2 provide the LP relaxation bounds produced by several of the models described in the paper for the p43, ry48p, rjj25, rjj35, ejj25 and ejj35 instances. The entries correspond to:

- Name: name of the instance
- n : number of nodes in the instance
- $|B|$: number of precedence constraints in the instance
- OPT: optimal integer value OR best known upper and lower bounds
- M_1 : base model (1), (2), (4) + (7), (10), (13) + lifted DDL inequalities (11)
- M_2 : M_1 + GDDL cuts (14) + simple cuts (15)–(17)
- M_3 : M_2 + 2-path cuts (39)–(42)
- M_4 : M_3 + $3v$ GDDL-like inequalities (58) + (61) + (62)
- M_5 : M_4 + $4v$ GDDL-like inequalities (64) + (65)
- LPB: LP relaxation bound of the corresponding model
- t : time, in seconds, taken to solve the LP relaxation of the corresponding model

The values in the column OPT were taken from Gouveia and Ruthmair (2015) in the case of the instances p43 and ry48p, and computed by model M_4 in the case of the instances rjj and ejj. In both tables, the LP bound values coinciding with the optimal integer values are highlighted in boldface.

Given that the cutting-plane algorithm we use follows an incremental approach, whenever the LP relaxation bound provided by a given model coincides with the optimal integer value, we do not attempt

to solve the same instance with a more complex model. Note that the LP relaxation bound obtained by a more complex model would be the same and the computational time to find such bound would increase very marginally due to the incremental approach mentioned before. We observe again that the main focus of our study is in evaluating the quality of the LP relaxation bounds provided by the various valid inequalities presented and not in finding the best strategies to use these inequalities in a branch-and-cut algorithm to solve the PCATSP.

Name	n	$ B $	OPT	M_1		M_2		M_3		M_4		M_5	
				t	LPB	t	LPB	t	LPB	t	LPB	t	LPB
p43.1	44	9	28140	14	920	4673	28138.3	10030	28138.7	11889	28140	-	-
p43.2	44	20	28480	12	1064	6603	28397	18657	28401	86001	28425.7	-	-
p43.3	44	37	28835	23	1449.11	3127	28643.2	6173	28653	45384	28707	-	-
p43.4	44	58	83005	1	56000.8	71	82874.6	172	82876.9	7888	82919.8	-	-
ry48p.1	49	11	15805	51	13889.6	3213	15331.6	5417	15344.7	60460	15420.2	-	-
ry48p.2	49	23	[16074, 16666]	68	14060.6	2263	15711.9	3936	15725.5	61439	15864.9	-	-
ry48p.3	49	42	[19490, 19894]	98	15907.6	773	17718.2	1836	17742.5	28658	18256.2	-	-
ry48p.4	49	58	31446	6	25124.8	171	27397.6	401	27499.9	10982	28377.2	-	-
rjj25.1	25	10	1221	0	1155.69	2	1193.77	6	1196.48	90	1204.08	848	1205.36
rjj25.2	25	20	1317	0	1240.2	2	1288.18	4	1292.21	43	1298.86	1323	1305.71
rjj25.3	25	30	1427	0	1370.83	1	1420.27	2	1420.85	58	1426.98	66	1427
rjj25.4	25	40	1788	0	1590.56	1	1676	1	1676	64	1704.92	392	1705.18
rjj25.5	25	50	1863	0	1669.57	1	1788.33	1	1788.33	13	1808.75	141	1809
rjj25.6	25	60	1863	0	1669.57	1	1788.33	1	1788.33	13	1808.75	139	1809
ejj25.1	25	10	2753	0	2366.52	5	2702.5	12	2716.43	74	2753	-	-
ejj25.2	25	20	3102	0	2735.51	5	3048.23	9	3066.47	53	3102	-	-
ejj25.3	25	30	3703	0	3205.1	4	3600.46	7	3615.14	51	3703	-	-
ejj25.4	25	40	4042	0	3436.1	3	3903.12	8	3917.31	158	4029.24	381	4042
ejj25.5	25	50	4129	0	3490.86	3	3913.1	10	3932.28	159	4073.34	1767	4113.42
ejj25.6	25	60	4129	0	3546.31	3	3948.94	7	3974.72	140	4118.01	365	4129
rjj35.1	35	20	1403	3	1243.75	18	1275.82	42	1276.56	819	1285.3	24012	1303.52
rjj35.2	35	40	1830	3	1630.28	11	1726.12	18	1726.12	781	1737.12	13357	1741.17
rjj35.3	35	60	2352	1	2080.91	7	2126.99	15	2130.14	594	2147.25	9657	2150.81
rjj35.4	35	80	2756	1	2537.8	5	2594.43	14	2598.63	601	2660.69	7718	2664.65
rjj35.5	35	100	3219	0	2901.08	8	3158.38	16	3163.22	757	3183.12	11970	3187.26
ejj35.1	35	20	3202	3	2644.84	91	2942.53	255	2949.87	4307	3039.8	130223	3072.99
ejj35.2	35	40	3640	1	2838.24	36	3197.64	100	3200.54	1881	3444.02	61161	3482.49
ejj35.3	35	60	4395	2	3657.43	18	3848.48	56	3866.95	1354	4231.91	29513	4265.54
ejj35.4	35	80	4638	1	3982.16	19	4294.61	72	4312.1	1621	4633.72	6491	4638
ejj35.5	35	100	5165	0	4452.73	12	4916.02	35	4927.49	38	5165	-	-

Table 1: Results for the PCATSP instances with formulations without the extra node removals

Table 1 shows that, in the p43 instances, the new $3v$ GDDL-like inequalities, on the one hand, substantially improve the LP relaxation bound when compared to the models with only GDDL cuts, simple cuts, and 2-path cuts. On the other hand, the models with these new inequalities also take substantially longer to solve. Observe that for the instance p43.1, the LP relaxation bound when the $3v$ GDDL-like inequalities are added coincides with the optimal integer solution value.

With respect to the ry48p instances the conclusions are fairly similar. Once again, the new $3v$ GDDL-like inequalities are able to considerably improve the LP relaxation bounds, although the bounds are still far from the corresponding optimal integer values.

Finally, we observe that we do not have results with the $4v$ GDDL-like inequalities in the case of the p43 and ry48p instances since their separation takes too long.

Regarding the self-generated instances, we can conclude that with Euclidean distances the models behave a lot better when compared to random symmetric costs. In fact, for the Euclidean instances, we were able to obtain the optimal solution value of 5 out of 6 instances with 25 nodes and 2 out of 6 instances with 35 nodes. Of these 7 optimal values obtained, 4 were obtained with model M_4 while the other 3 only with model M_5 . This reinforces the significance of the $3v$ and $4v$ inequalities. In the random symmetric cost instances we were only able to obtain one optimal value (instance rjj25.3). We can see that the $3v$ inequalities are able to substantially improve the LP relaxation bounds at the cost of increased

computational time. Compared to the $4v$ inequalities, where the computational times can be prohibitively long (even though the LP relaxation bounds are better), the increase in computational time due to the addition of the $3v$ GDDL-like inequalities is not that large, thus, with a “smarter” way of using these inequalities we can hope to incorporate them in a successful branch-and-cut procedure. We have also seen that the improvement made by the $4v$ GDDL-like inequalities is mostly due to inequalities (65), although this is not shown in this table.

We have also made a similar study where node elimination as described in Section 4 has been incorporated in the inequalities. These results are described in Table 2, where the entries correspond to the same models as described before. We observe, however, that the M_2 model also includes the reversed GDDL inequalities (35) since they now differ from the regular strengthened GDDL inequalities (73) in terms of the nodes to be removed, as explained in Section 4.

Name	n	$ B $	OPT	M_1		M_2		M_3		M_4		M_5	
				t	LPB	t	LPB	t	LPB	t	LPB	t	LPB
p43.1	44	9	28140	14	920	5379	28138.8	12215	28138.8	13959	28140	-	-
p43.2	44	20	28480	12	1064	10848	28429.5	29686	28432	125780	28443.4	-	-
p43.3	44	37	28835	23	1449.11	5708	28733.4	9926	28740.7	36861	28785.7	-	-
p43.4	44	58	83005	1	56000.8	270	82991.7	585	82993.6	1642	83005	-	-
ry48p.1	49	11	15805	42	13889.6	3155	15386.1	5507	15403.8	103330	15477.4	-	-
ry48p.2	49	23	[16074, 16666]	83	14060.6	2502	15806	4275	15844.7	93498	15938.1	-	-
ry48p.3	49	42	[19490, 19894]	84	15907.6	1463	18352.1	4373	18506.7	39339	18704.9	-	-
ry48p.4	49	58	31446	3	25124.8	453	30465	912	30599.7	15651	30919.8	-	-
rjj25.1	25	10	1221	0	1155.69	3	1202.93	8	1208.13	162	1213.08	1109	1214.46
rjj25.2	25	20	1317	0	1240.2	4	1308.86	8	1312.64	59	1317	-	-
rjj25.3	25	30	1427	0	1370.83	2	1427	-	-	-	-	-	-
rjj25.4	25	40	1788	0	1590.56	1	1676	1	1676	102	1712.91	659	1713.51
rjj25.5	25	50	1863	0	1669.57	1	1788.33	1	1788.33	14	1808.75	139	1809
rjj25.6	25	60	1863	0	1669.57	0	1788.33	1	1788.33	16	1808.75	147	1809
ejj25.1	25	10	2753	0	2366.52	4	2753	-	-	-	-	-	-
ejj25.2	25	20	3102	0	2735.51	2	3102	-	-	-	-	-	-
ejj25.3	25	30	3703	0	3205.1	4	3703	-	-	-	-	-	-
ejj25.4	25	40	4042	0	3436.1	6	4042	-	-	-	-	-	-
ejj25.5	25	50	4129	0	3490.86	6	4129	-	-	-	-	-	-
ejj25.6	25	60	4129	0	3546.31	3	4129	-	-	-	-	-	-
rjj35.1	35	20	1403	3	1243.75	33	1290.12	74	1299.75	1084	1305.15	25739	1313.07
rjj35.2	35	40	1830	3	1630.28	36	1766.95	76	1774.11	842	1774.83	12946	1778.52
rjj35.3	35	60	2352	1	2080.91	22	2243.11	46	2264.14	907	2268.14	8025	2270.51
rjj35.4	35	80	2756	1	2537.8	16	2707.82	45	2750.91	48	2756	-	-
rjj35.5	35	100	3219	0	2901.08	8	3219	-	-	-	-	-	-
ejj35.1	35	20	3202	3	2644.84	92	2978.28	292	3000.36	5399	3047.97	124059	3073.15
ejj35.2	35	40	3640	1	2838.24	85	3332.26	224	3368.53	2507	3446.75	54463	3484.28
ejj35.3	35	60	4395	2	3657.43	52	4113.05	183	4174.83	2094	4239.93	33137	4273.67
ejj35.4	35	80	4638	1	3982.16	58	4570.94	156	4610.55	1450	4636	2173	4638
ejj35.5	35	100	5165	0	4452.73	23	5165	-	-	-	-	-	-

Table 2: Results for the PCATSP instances with formulations with the extra node removals

Our first observation is that, if we compare the results for the p43 instances from the previous table to these new results, the strengthened inequalities provide a considerable improvement when compared with the non-strengthened inequalities. Also, the models with the strengthened inequalities do not take longer to solve. We were able to obtain the optimal value of instance p43.4 with strengthened model M_4 and, regarding instances p43.2 and p43.3, the LP relaxation bounds are significantly better. Furthermore, for these four instances, we can see that the LP relaxation bounds of models with fewer sets of strengthened inequalities provide significantly better bounds than non-strengthened models with more sets of inequalities. For instance, regarding the instance p43.3, the LP relaxation bound given by the strengthened M_2 model is stronger than the one obtained by the M_4 model.

Our second observation, which is related to the previous one, is that the “simpler” inequalities benefit more from the extra node eliminations. This is not only clear in terms of results (see the examples in the previous paragraph), but also theoretically since the node elimination sets contain many more nodes,

especially in instances with many precedence relations. The more “complicated” inequalities have node elimination sets which are tighter, thus, when compared to the “simpler” inequalities, they contain very few nodes or are even empty in some cases. Even so, for the p43 instances which have more precedence relations, namely p43.3 and p43.4, our study indicates that these “complicated” inequalities can benefit from the node elimination sets since there are several such sets which are non-empty.

Regarding the ry48p instances, we were able to obtain some interesting results. The LP relaxation bounds that our models obtained for the instances ry48p.2 and ry48p.3 are quite close to the best known lower bounds (note that the best known lower bounds were obtained with branch-and-cut methods, therefore they are not “pure” LP relaxation bounds). As for the instances ry48p.1 and ry48p.4, although we were unable to obtain the optimal value, once again our LP relaxation bounds are quite close. Once again we can see that the LP relaxation bounds of models with fewer sets of strengthened inequalities provide significantly better bounds than non-strengthened models with more sets of inequalities, as was the case with the p43 instances. Furthermore, in instances ry48p.3 and ry48p.4 we can see that the strengthened model M_2 provides a better LP relaxation bound value than the non-strengthened model M_4 and, in the case of instance ry48p.4, this improvement is considerable.

All the remarks made so far still apply to the self-generated instances. We can see that the strengthened inequalities are considerably better than the non-strengthened ones since, in the case of Euclidean distances, we are now able to obtain 6 out of 6 optimal values for the instances ejj25.1-6 by using “only” the strengthened model M_2 . Regarding instances ejj35.1-5, we do not obtain any new optimal values but, once more, instance ejj35.5 is now solved by strengthened model M_2 . As for instances with random symmetric costs, where we previously could only obtain one optimal value out of 11 instances, we are now able to obtain 4 out of 11 optimal integer values and instances rjj25.3 and rjj35.5 are now solved by the strengthened model M_2 . Finally, all of the unsolved instances have greatly improved LP relaxation bounds.

Regarding both tables, we notice that, in general, as the number of precedence constraints increases, the time to obtain the LP relaxation bounds decreases. This is probably due to the fact that if a precedence is given in the instance, for example $(i, j) \in B$, then variables x_{ji} , v_i^j and v_j^i do not exist in the model, since we can replace them by 0, 1 and 0 in every constraint where they were to appear.

Our main conclusion from this preliminary computational study is that the new inequalities that we studied in this paper, although slow to separate, provide interesting LP relaxation bounds, especially if we incorporate the information obtained from precedence relations to create node elimination sets in order to further strengthen the inequalities.

We conclude this section by providing some information on the value of the LP relaxation bounds of the proposed models for a set of ATSP instances, namely instances ftv33, ftv35, ftv38, ftv44, ftv47, ftv53 and ftv55. The results obtained are presented in Table 3, which follows the same format as the previous two tables. Since these instances do not have any precedence relations, the column that indicates the number of precedence relations, $|B|$, is removed and, in addition, recall that the strengthened inequalities presented in Section 4 are the same as the non-strengthened ones.

Name	n	OPT	M_1		M_2		M_3		M_4		M_5	
			t	LPB	t	LPB	t	LPB	t	LPB	t	LPB
ftv33	34	1286	2	1229.08	9	1286	-	-	-	-	-	-
ftv35	36	1473	7	1425.5	29	1457.33	33	1457.33	298	1458.13	3358	1458.13
ftv38	39	1530	10	1485.44	33	1514.33	38	1514.33	465	1515.12	5820	1515.12
ftv44	45	1613	60	1580.88	177	1584.87	297	1584.87	5691	1588.98	-	-
ftv47	48	1776	85	1734	434	1748.61	526	1748.61	4411	1753.1	-	-
ftv53	53	6905	160	6061.93	1071	6905	-	-	-	-	-	-
ftv55	56	1608	271	1541	1254	1584	873	1584	63042	1588.81	-	-

Table 3: Results for the ATSP instances

Gouveia and Pesneau (2006) showed that the GDDL model provides, for ATSP instances, LP relaxation bounds that are equal to the bounds provided by the formulation obtained by replacing (3) with (5) or (6). In theory, model M_2 provides bounds that are at least as good, but for all the instances tested the

bounds can be seen to be the same. For the *ftv44*, *ftv47* and *ftv55* instances, the results provided by the model M_4 are interesting in the sense that they may lead to further developments, such as investigating which kind of ATSP inequalities are implied by the new $3v$ GDDL-like inequalities (and the $4v$ GDDL-like inequalities as a consequence). As before, we did not attempt to solve the LP relaxation of the model M_5 for the four larger instances.

6 Conclusion

In this paper we have discussed flow based models involving precedence variables and exploiting the disjoint path property for different pairs of precedences. We have shown how to find flow based systems that imply the GDDL inequalities, in particular one such system that results from adequately combining the linking constraints associated with two flow based systems corresponding to the separation of simple cut inequalities. This derivation is useful for several reasons: i) it gives us a procedure to eliminate nodes from the cut sets for the case of the PCATSP and ii) it provides us with a generic idea to generate more complicated inequalities, either by using the flow based systems associated with the simple cut inequalities or by using a similar procedure but starting with even more complicated flow systems as shown in Section 3.2.

The computational results we have conducted indicate that these more general inequalities, although harder and slower to use from a practical point of view, may significantly bridge the LP gap when compared with models using the original set of inequalities (although for several cases, this original set is already quite good).

As mentioned before, this work is exploratory in the sense that there is no study on making the cutting plane algorithm more practical, that is, there is no study on dominance between several different sets of inequalities neither on heuristics to separate only a subset of them. Such a study could make the inequalities here presented even more attractive to use when solving PCATSP instances with a cutting plane approach.

We conclude by pointing out that the 3-step procedure here proposed could be used to obtain even more general inequalities by starting with other k -path ($k \geq 3$) inequalities. However, it seems clear that this would lead to inequalities that may not be easy to use since they would have at least $4v$ variables on the right-hand side and, as we have seen, the separation of the $4v$ GDDL-like inequalities already takes too long.

7 Acknowledgements

We thank the three reviewers whose comments have helped to improve the paper. We also thank Juan-José Salazar-González for providing us with the instance generator used in Letchford and Salazar-González (2016). L. Gouveia and D. Santos were supported by Portuguese National Funding from Fundação para a Ciência e a Tecnologia, under projects UID/MAT/04561/2013 and PTDC/MAT NAN/2196/2014. M. Ruthmair has been funded by the Vienna Science and Technology Fund (WWTF) through project ICT15-014. D. Santos was also supported by the PhD grant “Programa de Bolsas de Doutorado da Universidade de Lisboa”.

References

- E. Balas, M. Fischetti, and W. R. Pulleyblank. The precedence-constrained asymmetric traveling salesman polytope. *Mathematical Programming*, 68:241–265, 1995.
- M. O. Ball, W.-G. Liu, and W. R. Pulleyblank. Two terminal steiner tree polyhedra. In B. Cornet and H. Tulkens, editors, *Contributions to operations research and economics*, pages 251–284. MIT Press, 1989.
- M. Barbato, R. Grappe, M. Lacroix, and R. Wolfer Calvo. A set covering approach for the double traveling salesman problem with multiple stacks. In R. Cerulli, S. Fujishige, and A. R. Mahjoub, editors,

Combinatorial Optimization: 4th International Symposium, ISCO 2016, Vietri sul Mare, Italy, May 16-18, 2016, Revised Selected Papers, pages 260–272. Springer International Publishing, 2016.

- A. A. Cire and W.-J. van Hoeve. Multivalued decision diagrams for sequencing problems. *Operations Research*, 61(6):1411–1428, 2013.
- A. Claus. A new formulation for the travelling salesman problem. *SIAM J. Algebraic Discrete Methods*, 5(1):21–25, 1984.
- M. Desrochers and G. Laporte. Improvements to the Miller-Tucker-Zemlin subtour elimination constraints. *Operations Research Letters*, 10:27–36, 1991.
- M. T. Godinho, L. Gouveia, and P. Pesneau. On a time-dependent formulation and an updated classification of ATSP formulations. In A. R. Mahjoub, editor, *Progress in Combinatorial Optimization*. Wiley, 2011.
- L. Gouveia and P. Pesneau. On extended formulations for the precedence constrained asymmetric traveling salesman problem. *Networks*, 48(2):77–89, 2006.
- L. Gouveia and J. M. Pires. The asymmetric travelling salesman problem and a reformulation of the Miller-Tucker-Zemlin constraints. *European Journal of Operational Research*, 112:134–146, 1999.
- L. Gouveia and J. M. Pires. The asymmetric travelling salesman problem: On generalizations of disaggregated Miller-Tucker-Zemlin constraints. *Discrete Applied Mathematics*, 112:129–145, 2001.
- L. Gouveia and M. Ruthmair. Load-dependent and precedence-based models for pickup and delivery problems. *Computers & Operations Research*, 63:56–71, 2015.
- M. Grötschel, M. Jünger, and G. Reinelt. A cutting plane algorithm for the linear ordering problem. *Operations Research*, 32:1195–1220, 1984.
- A. Langevin, F. Soumis, and J. Desrosiers. Classification of travelling salesman problem formulations. *Operations Research Letters*, 9:127–132, 1990.
- A. N. Letchford and J.-J. Salazar-González. Stronger multi-commodity flow formulations of the (capacitated) sequential ordering problem. *European Journal of Operational Research*, 251(1):74–84, 2016.
- K. Menger. Zur allgemeinen kurventheorie. *Fundamenta Mathematicae*, 10:96–115, 1927.
- C. Miller, A. Tucker, and R. Zemlin. Integer programming formulations and the travelling salesman problems. *J. ACM*, 7:326–329, 1960.
- T. Öncan, İ. K. Altinel, and G. Laporte. A comparative analysis of several asymmetric traveling salesman problem formulations. *Computers & Operations Research*, 36(3):637–654, 2009.
- R. Roberti and P. Toth. Models and algorithms for the asymmetric traveling salesman problem: an experimental comparison. *EURO Journal on Transportation and Logistics*, 1(1-2):113–133, 2012.
- R. Wong. Integer programming formulations of the travelling salesman problem. In *Proceedings of the IEEE International Conference of Circuits and Computers*, pages 149–152, 1980.