



HAL
open science

On extended formulations for the precedence constrained asymmetric traveling salesman problem

Luís Gouveia, Pierre Pesneau

► **To cite this version:**

Luís Gouveia, Pierre Pesneau. On extended formulations for the precedence constrained asymmetric traveling salesman problem. *Networks*, 2006, 48 (2), pp.77-89. 10.1002/net.20122 . inria-00281586

HAL Id: inria-00281586

<https://inria.hal.science/inria-00281586>

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.

On extended formulations for the Precedence Constrained Asymmetric Traveling Salesman Problem*

L. Gouveia¹, P. Pesneau²

¹ CIO-DEIO, Faculdade de Ciências da Universidade de Lisboa,

² Faculdade de Ciências da Universidade de Lisboa,

^{1,2} Bloco C6, Piso 4, Campo Grande, 1749-016 Lisboa, Portugal

Emails: legouveia, ppesneau@fc.ul.pt

April 13, 2006

Abstract

In this paper we study the use of formulations with precedence relation variables for the Precedence Constrained Asymmetric Travelling Salesman (PCATS) problem. Contrary to previous papers, the emphasis of this paper is on formulations involving exponential sized sets of inequalities and on the development of a cutting plane method together with polynomial routines for separating the new inequalities. Our computational results, taken from a set of benchmark instances, show that our methods improve significantly on most of the best previously known lower bound values.

Keywords Asymmetric Traveling Salesman Problem, Precedence Constraints, Projection, Cutting Plane.

1 Introduction

Consider a loopless directed graph $G = (V, A)$ with $V = \{1, \dots, n\}$ and a cost c_{ij} associated to each arc $(i, j) \in A$. The *Asymmetric Travelling Salesman (ATS) problem* is to find a minimum-cost Hamiltonian circuit (an elementary circuit traversing all nodes) (see [17]). In some cases, more conditions are required. This is the case of the ATS problem with Precedence Constraints (PCATS problem) where some nodes are required to be visited before other nodes. To define the precedence constraints, an arbitrary node, say 1, is designated to be the start and the end of the tour. We will denote by $i \prec j$ the precedence constraint stating that node i must be visited before node j and we let $B = \{(i, j) \in V^2 : i \prec j\}$ be the so-called *precedence graph*. We assume that B is transitively closed. Then the PCATS problem is to find a minimum-cost Hamiltonian circuit such that all precedence constraints of B are satisfied.

The PCATS problem has many real-world applications, the more intuitive one concerns delivery or routing problems where a pick-up has to precede a delivery (see [25] for an example). However, other applications can be found in scheduling [2] or in the routing of a stacker crane in an automatic storage system (see [1]). Note that, if we split the starting/ending node in two nodes, say 1 and $n + 1$, the PCATS problem is equivalent to finding a minimum-cost Hamiltonian path between nodes 1 and $n + 1$ satisfying the precedence constraints of B . In this disguise, the problem is usually known as the *Sequential Ordering (SO) Problem*. In fact, the previously mentioned work by Ascheuer et al [2] addresses this version of the problem. The designation *Sequential Ordering (SO) Problem* and viewing the problem as determining an Hamiltonian path arises from a job scheduling application where some jobs have to be processed before other ones. As pointed out, the two versions are equivalent and an approach proposed for one is easily adapted for the other.

A polyhedral study of this problem on the set of natural design variables can be found in [4] while [3] discusses a Branch-and-Cut algorithm based on the previous work. A dynamic programming approach has

*Supported by research contract numbers: POCTI-ISFL-1-152 and MRTN-CT-2003-504438.

also been proposed in [5] for the PCATS problem. This approach works quite well for tightly constrained cases but as not as well for the other cases. A similar type of approach has been proposed in [20] for a generalization of the PCATS problem which also includes time window constraints.

In [11] and [12], Gouveia and Pires propose extended and compact formulations for the ATS problem by creating disaggregated versions of the well-known Miller-Tucker-Zemlin formulation [19]. The information associated to the extra variables used in these formulations (the variables indicate precedence relations on pairs of nodes) is adequate for handling the precedence constraints. Later on, Sarin et al. [24] make use of this fact and propose related compact formulations for the PCATS problem.

In the previous works by Gouveia and Pires [11], [12] and Sarin et al. [24], the emphasis was on compact formulations and on using them together with a MIP package to solve the problem or to obtain lower bounds on the value of the optimal solution. The current work focus on extended formulations that use several classes of *exponentially* sized sets of constraints on the same two sets of variables and on using polynomial routines to separate them. Our computational results will show that the inclusion of the new classes of constraints leads to models with substantially stronger linear programming relaxations.

Section 2 starts the model presentation of the current work with a formulation taken from [12] that contains an exponential sized set of constraints. Inequalities that can improve on the linear programming relaxation of this formulation will be discussed in Section 3. Section 3.1 discusses inequalities (or stronger versions of them) taken from the related Linear Ordering problem while Section 3.2 and 3.3 introduce several exponential sized sets of new cut-like and multicut-like inequalities. Section 4 systematizes the different classes of inequalities presented in the paper and suggests ways of deriving new classes of inequalities. Section 5 presents some computational results using benchmark tests and where we evaluate the strength of the linear programming relaxations of the proposed models.

We conclude this section by introducing some terminology. Let W_1 and W_2 be two disjoint nonempty node subsets of V . We will denote by $\delta(W_1, W_2)$ the set of arcs having their tail in W_1 and their head in W_2 . If W_1 and W_2 form a partition of V , then $\delta(W_1, W_2)$ is called the *cut* associated to W_1 . If S is a set of nodes, we will denote by $E(S)$ the set of arcs having their tail and head in S . Given a node i , we will denote by $G \setminus i$ the graph induced by $V \setminus \{i\}$, that is $G \setminus i = (V \setminus \{i\}, E(V \setminus \{i\}))$. For a set of arcs A and a vector x indexed on the arcs of G , we let $x(A) = \sum_{(i,j) \in A} x_{ij}$. We will denote by $P_{\mathbf{L}}$ the linear programming relaxation of the formulation P and by $F(P)$ its set of solutions.

2 Formulation

In this section we review one of the formulations presented by Gouveia and Pires [12] and we show that it can be easily adapted for the PCATS problem. We also present a complete characterization of the feasible set of the corresponding linear programming relaxation into the space of the original variables.

Consider the set of variables x_{ij} stating whether the arc (i, j) belongs to a solution of the ATS problem. Many valid formulations for the ATS problem are based on the following general scheme

$$\text{Min } \sum_{(i,j) \in A} c_{ij} x_{ij}, \tag{1}$$

$$\sum_{i \in V} x_{ij} = 1 \quad \text{for all } j \in V, \tag{2}$$

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

$$\{(i, j) : i, j = 2, \dots, n; x_{ij} = 1\} \text{ do not contain subtours,} \tag{4}$$

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

$$x_{ij} \in \{0, 1\} \quad \text{for all } (i, j) \in A. \tag{6}$$

Inequalities (2), (3) and (6) define the usual assignment relaxation of the ATS problem. Inequalities (4), that prevent subtours not containing node 1, can be written in several ways, either by using the x_{ij} variables alone or by using extra sets of variables.

The most usual way of writing such inequalities, using the variables x_{ij} alone, is given by the clique inequalities that can be written as follows

$$x(E(S)) \leq |S| - 1, \quad \text{for all } \emptyset \neq S \subset V \setminus \{1\}. \tag{7}$$

By adding assignment inequalities (3) for all nodes in S and subtracting the result from (7), we obtain the following equivalent form of inequalities (7), the so-called cut inequalities

$$x(\delta(S, V \setminus S)) \geq 1, \quad \text{for all } \emptyset \neq S \subset \{2, \dots, n\}. \quad (8)$$

As noted before, constraints (4) can also be modelled by using inequalities involving additional variables. The reader is referred to the papers [26], [16], [22] and [11] for a review of some such formulations and the relationship of the feasible set of the corresponding linear programming relaxations with the linear programming relaxations of formulations using the variables x_{ij} alone. In general, the main idea of using extra variables is that the information associated to the new variables permits us to write compact formulations (that is, formulations with a polynomial number of variables and constraints) for the ATS problem. In this work we will also focus on formulations with an extra set of variables (as they model quite adequately the additional precedence constraints) but we will "relax the condition" of focusing only on compact formulations.

In [11] and [12], Gouveia and Pires introduced several extended formulations for the ATS problem using the "precedence relation" variables v_i^j , $i, j = 2, \dots, n$ and $i \neq j$, stating whether node i is in the path from 1 to j . In other words, if $v_i^j = 1$, then node i is before node j in the tour. Although, the focus of these works has been on compact formulations, they have also introduced an exponential sized formulation that is obtained by replacing (4) with the following inequalities

$$v_i^j + x_{ji} \leq 1, \quad \text{for } i, j = 2, \dots, n, i \neq j, \quad (9)$$

$$x_{ij} \leq v_i^j, \quad \text{for } i, j = 2, \dots, n, i \neq j, \quad (10)$$

$$v_k^i + x(E(S)) \leq v_k^j + |S| - 1 \quad \text{for all } i, j, k \in \{2, \dots, n\}, \quad (11)$$

and for all $S \in V \setminus \{1, k\}, i, j \in S$.

Inequalities (9) state that we cannot have, at the same time, an arc (i, j) and j before i in a solution. Inequalities (10) indicate that if an arc (i, j) is in the solution, then i must be before j in the solution. To see that inequalities (11) are valid, first, notice that the only situation of interest is when $x(E(S)) = |S| - 1$ (otherwise the inequality is trivially valid). However, if $x(E(S)) = |S| - 1$, then the nodes in the set S must be linked together by an elementary path. Thus, if node p (that does not belong to S) is in the path to the node i , inside S , then node p is also in the path to any other node of S and in particular, to node j . Inequalities (11) are denoted by "Generalized Disaggregated Desrochers and Laporte" (GDGL) inequalities. The reason for this designation is that, as pointed out in [11], the corresponding inequalities for $|S| = 2$ together with (9), (10) can be seen as a disaggregation of the Desrochers and Laporte inequalities (see [8]).

By adding to the formulation the set of equalities

$$v_i^j = 1, \quad \text{for all } (i, j) \in B, \quad (12)$$

$$v_j^i = 0, \quad \text{for all } (i, j) \in B, \quad (13)$$

we obtain a valid formulation, denoted by PCGDGL, for the PCATS problem. To see that (12) and (13) together with the remaining constraints of the model satisfy the precedence relations given in B , let us suppose that (\bar{x}, \bar{v}) is an integral solution of the formulation such that there exists a pair of nodes i and j with $v_i^j = 1$ ($i \prec j$) and i is not before j in the tour. If $\bar{x}_{ji} = 1$ then inequality (9) for the nodes i and j is violated, a contradiction. Suppose now there is a sequence of nodes k_1, \dots, k_q , $q \geq 1$, in the tour between j and i and let S be the set of nodes $\{j, k_1, \dots, k_q\}$. Note that $x(E(S)) = |S| - 1$. It follows that the inequality (11) for the nodes j, k_q and i and for the set S leads to $v_i^j \leq v_i^{k_q}$. As $v_i^j = 1$, we have $v_i^{k_q} = 1$ and, by inequality (9) for nodes i and k_q , we get $x_{k_q i} = 0$, a contradiction. In consequence, if $v_i^j = 1$, then i must be before j in the tour and the formulation PCGDGL is valid for the PCATS problem.

Notice that equations (12) and inequalities (9) imply that we have

$$x_{ji} = 0, \quad \text{for all } (i, j) \in B. \quad (14)$$

Clearly, the clique inequalities (7) are also valid for the PCATS problem. We can make use of the precedence constraints in order to obtain stronger forms of these inequalities. We describe next one such set of inequalities, the precedence cycle breaking inequalities (see [4]), that are valid for the PCATS problem and will appear later in the context of our models. Let S_1, \dots, S_m be $m \geq 2$ disjoint subsets of nodes of $V \setminus \{1\}$ such that for each $i = 1, \dots, m$, there is at least one precedence constraint between a node of S_i and a node of S_{i+1} (with $S_{m+1} = S_1$). Then the precedence cycle breaking inequality associated to the sets S_1, \dots, S_m is

$$\sum_{i=1}^{i=m} x(E(S_i)) \leq \sum_{i=1}^{i=m} |S_i| - m - 1. \quad (15)$$

Note that the previous inequality is stronger than the one obtained by adding the m clique inequalities (7), each one associated to a different set S_i .

We study, next, the projection of $\text{PCGDDL}_{\mathcal{L}}$ into the space of the x_{ij} variables. As observed in [12], by combining a GDDL inequality for a triple i, j and k with the similar inequality obtained by reversing the role of nodes i and j , we obtain the clique inequality (7) for the set S . The same inequality is obtained if $i \prec k$ and $j \prec k$ are precedence constraints of B .

In this paper, we show that three stronger versions of the clique inequalities can also be obtained from (9), (10) and (11) when we consider the PCATS problem and some adequate precedence constraints.

Suppose, for instance, that $k \prec i$ and $j \prec k$ are precedence constraints of B . Then $v_k^i = 1$ and $v_k^j = 0$ and the inequality can be written as

$$\begin{aligned} x(E(S)) \leq |S| - 2 \quad & \text{for all } i, j, k \in \{2, \dots, n\}, (k, i), (j, k) \in B, \\ & \text{and for all } S \in V \setminus \{1, k\}, i, j \in S. \end{aligned} \quad (16)$$

These inequalities are a special case of the precedence cycle breaking inequalities (15) with only 2 sets and with one set being a single node set, the set $\{k\}$.

Consider now the case where we have the precedence constraint $k \prec i$ but neither $j \prec k$ nor $k \prec j$ is in B . We can rewrite (11) as

$$x(E(S)) \leq v_k^j + |S| - 2.$$

By adding inequality (9) for the nodes k and j and reordering the terms, we obtain the following lifted clique inequality

$$\begin{aligned} x(E(S)) + x_{jk} \leq |S| - 1 \quad & \text{for all } i, j, k \in \{2, \dots, n\}, (k, i) \in B, \\ & \text{and for all } S \in V \setminus \{1, k\}, i, j \in S. \end{aligned} \quad (17)$$

The previous set considers a precedence constraint from a node outside the set S to a node inside S . We can derive a similar class of lifted clique inequalities by considering, in alternative, a precedence constraint from a node in S to a node outside S . If $j \prec k$ is in B and neither $k \prec i$ nor $i \prec k$ is in B , we can rewrite (11) as

$$v_k^i + x(E(S)) \leq |S| - 1.$$

By adding inequality (10) for the nodes k and i , we obtain the following lifted clique inequality

$$\begin{aligned} x(E(S)) + x_{ki} \leq |S| - 1 \quad & \text{for all } i, j, k \in \{2, \dots, n\}, (j, k) \in B, \\ & \text{and for all } S \in V \setminus \{1, k\}, i, j \in S. \end{aligned} \quad (18)$$

Clearly, inequalities (7), (16), (17) and (18) are included in the projection of $\text{F}(\text{PCGDDL}_{\mathcal{L}})$ on the space of the x_{ij} variables. The next result shows that nothing else of interest is contained in such a projection.

Let us call SPCB the formulation given by (2), (3), (7), (16)-(18), (14), (6). The following theorem, whose proof can be found in appendix, states the projection of $\text{F}(\text{PCGDDL}_{\mathcal{L}})$.

Theorem 1 $\text{F}(\text{SPCB}_{\mathcal{L}})$ is the projection of $\text{F}(\text{PCGDDL}_{\mathcal{L}})$ on the space of the x_{ij} variables.

The previous result is more general than the result conjectured in [12], when B is empty, and later proved independently in Pires PhD thesis [23] and in [21]. In this particular case, of the previously referred four sets of inequalities, only the clique inequalities are included in the projection.

Although, (11) contains too many inequalities, we note that such inequalities can be separated in polynomial time. To see this, note that by using the assignment inequalities, these constraints can be rewritten as

$$\begin{aligned} v_k^i + 1 \leq v_k^j + x(\delta(S, V \setminus S)) \quad & \text{for all } i, j, k \in \{2, \dots, n\}, \\ & \text{and for all } S \in V \setminus \{1, k\}, i, j \in S. \end{aligned} \quad (19)$$

These inequalities can then be separated in polynomial time by computing a minimum cut separating nodes $1, k$ from nodes i and j for all possible combinations of the three nodes i, j and k and checking whether the value of the cut is greater or equal to the value of the expression $v_k^i - v_k^j + 1$.

3 Strengthening the formulation

In this section, we will present several classes of valid inequalities to strengthen the linear programming relaxation of the formulation PCGDDL . The k -cycle inequalities and lifted 3-cycle inequalities have been presented before. As far as we know, the remainder of the material is new.

3.1 Inequalities from the Linear Ordering problem

Following an observation by Letchford [18], if we replace (1) by an objective function on the v_i^j variables alone (that is, we are switching the role of the two sets of variables in these formulations, the original variables with the extra variables) we would obtain valid formulations for a precedence constrained version of the so-called linear ordering (LO) problem [13]. The polytope associated to the latter problem is described by the 2-cycle and 3-cycle inequalities, that will be presented later on, together with the integrality constraints on the v variables. In fact, any tour gives an ordering of the nodes in the set $V \setminus \{1\}$. Thus, it would make sense to use these two set of inequalities from the linear ordering problem in order to improve the linear programming relaxation of the PCGDDL model.

3.1.1 2-Cycle Equations

The first class of constraints is quite intuitive. Constraints (12) and (13) show that the equation

$$v_i^j + v_j^i = 1$$

is valid whenever (i, j) or (j, i) is in B . However, it is not difficult to see that the equation is valid for any pair of nodes i and j , as in the tour either i is before j or j is before i . Thus, the "2-cycle" constraints

$$v_i^j + v_j^i = 1 \quad \text{for all } i, j \in V \setminus \{1\}, i \neq j, \quad (20)$$

are valid. These constraints have already been used by Sarin et al. [24] in their extended formulations. As our computational results will show, these equations are quite effective to improve the linear programming relaxation of the previous formulation PCGDDL.

Note that, for two nodes i and j , either inequality (9) or inequality (10) becomes redundant under the presence of (20) and thus, one of them can be removed (in the sequel, we assume that inequalities (10) are removed).

Unfortunately, with respect to formulations involving constraints (20), it is much more difficult to study the projection of the corresponding linear programming feasible set in the space of the x_{ij} variables. The reason for this is that the dual problem arising in the Farka's lemma in the proof of Theorem 1, becomes much more complicated and loses the Network Flow structure that was so useful in the proof of Theorem 1 and also in the proofs of the results given in [11].

However, this does not mean that we cannot find new inequalities in the x_{ij} variables that are implied by the linear programming relaxation of the stronger model. Let i, j, p and q be 4 distinct nodes of $V \setminus \{1\}$ and let S_1 and S_2 be two distinct sets of $V \setminus \{1\}$ with $i, j \in S_1$ and $p, q \in S_2$. Note that p is not in S_1 and i is not in S_2 because S_1 and S_2 are disjoint. By adding the two inequalities (11), one obtained with S_1 and the nodes i, j and p , the other obtained with S_2 and the nodes q, p , and j , we get

$$v_p^i + v_j^q + x(E(S_1)) + x(E(S_2)) \leq v_p^j + v_j^p + |S_1| + |S_2| - 2.$$

Now, by adding constraint (20) for the nodes p and j , we obtain the following constraint

$$v_p^i + v_j^q + x(E(S_1)) + x(E(S_2)) \leq |S_1| + |S_2| - 1. \quad (21)$$

Following the reasoning given before for generating projected inequalities from constraint (11), we can now consider two different cases. Suppose, first, that both $j \prec q$ and $p \prec i$ are precedence constraints of B . In this case, we have $v_j^q = v_p^i = 1$ and then

$$\begin{aligned} x(E(S_1)) + x(E(S_2)) &\leq |S_1| + |S_2| - 3 \quad \text{for all } i, j, p, q \in \{2, \dots, n\}, (j, q), (p, i) \in B, \\ &\text{and for all } S_1, S_2 \in V \setminus \{1\}, i, j \in S_1, p, q \in S_2. \end{aligned} \quad (22)$$

These inequalities are precedence cycle breaking inequalities (15) for two non-single node sets S_1 and S_2 .

Suppose, now, that $j \prec q$ is a precedence constraint and neither is $p \prec i$ nor $i \prec p$ (the case where $p \prec i$ is a precedence constraint and neither is $j \prec q$ nor $q \prec j$ leads to the same type of inequality). Thus, we have $v_j^q = 1$. By replacing this in (21) and adding $x_{pi} \leq v_p^i$ we obtain

$$\begin{aligned} x(E(S_1)) + x(E(S_2)) + x_{pi} &\leq |S_1| + |S_2| - 2 \quad \text{for all } i, j, p, q \in \{2, \dots, n\}, (j, q) \in B, \\ &\text{and for all } S_1, S_2 \in V \setminus \{1\}, i, j \in S_1, p, q \in S_2. \end{aligned} \quad (23)$$

We have shown that the projection of the linear programming feasible set of PCGDDL with the constraints (20) into the space of the x_{ij} variables is contained in the linear programming relaxation of SPCB with inequalities (22) and (23). One interesting topic of research is to verify whether the previous inclusion is strict, or not.

3.1.2 k -Cycle Inequalities ($k \geq 3$)

One other intuitive set of inequalities "borrowed" from the linear ordering problem is given by the " k -cycle" inequalities

$$v_{i_1}^{i_2} + \dots + v_{i_{k-1}}^{i_k} + v_{i_k}^{i_1} \leq k - 1 \quad \text{for all sequences of nodes } \{i_1, \dots, i_k\} \subset V \setminus \{1\} \quad (24)$$

and for all integer $k \in \{3, \dots, n - 1\}$.

These inequalities prevent cycles of precedence constraints of length k . We can generalize the reasoning used in the previous subsection, in order to obtain all the precedence cycle breaking inequalities (15) from the GDDL inequalities (11) and the k -cycle inequalities (24). Suppose that S_1, \dots, S_m are m disjoint subsets of nodes of $V \setminus \{1\}$, each one containing at least two nodes. Let i_1, \dots, i_{2m} be $2m$ nodes of $V \setminus \{1\}$ such that $i_{2j-1}, i_{2j} \in S_j$ for $j = 1, \dots, m$. In the following, we assume that $S_{m+1} = S_1$, $i_{2m+1} = i_1$ and $i_{2m+2} = i_2$. Consider now the following GDDL inequalities (11), each one associated to a set S_j ,

$$v_{i_{2j}}^{i_{2j+1}} + x(E(S_{j+1})) \leq v_{i_{2j}}^{i_{2j+2}} + |S_{j+1}| - 1 \quad \text{for } j = 1, \dots, m.$$

If we add these m inequalities, we obtain the following inequality

$$\sum_{j=1}^{j=m} v_{i_{2j}}^{i_{2j+1}} + \sum_{j=1}^{j=m} x(E(S_j)) \leq \sum_{j=1}^{j=m} v_{i_{2j}}^{i_{2j+2}} + \sum_{j=1}^{j=m} |S_j| - m.$$

By adding to the previous inequality, the m -cycle inequality given by the sequence $\{i_2, i_4, \dots, i_{2m-2}, i_{2m}\}$, we obtain the inequality

$$\sum_{j=1}^{j=m} v_{i_{2j}}^{i_{2j+1}} + \sum_{j=1}^{j=m} x(E(S_j)) \leq \sum_{j=1}^{j=m} |S_j| - 1. \quad (25)$$

that is a generalization of the inequality (21) mentioned in the previous subsection.

Suppose, now, that we have the m precedence constraints $i_{2j} \prec i_{2j+1}$ for $j = 1, \dots, m$. It follows that $v_{i_{2j}}^{i_{2j+1}} = 1$ for $j = 1, \dots, m$ and we obtain the cycle breaking inequality (15) associated to the sets S_1, \dots, S_m . Note that our derivation does not "create" cycle breaking inequalities (15) where one of the sets is reduced to a single node set. However, as it has been shown in [4], these inequalities are redundant relatively to other inequalities (15).

Note also that, as we have done in the previous section, we can consider that only some of the pairs of nodes (i_{2j}, i_{2j+1}) for $j = 1, \dots, m$ define precedence constraints. For the remaining pairs, by using inequalities (10), we replace the corresponding v_i^j variables by x_{ij} variables and obtain generalizations of constraints (23). However, we can show that the resulting inequality is dominated by an inequality (15) where we group together two consecutive node sets not having a precedence constraint between nodes in each of these two sets.

Our results will show that the 3-cycle constraints are not as effective to close the linear programming gap of PCGDDL as are the 2-cycle constraints. We did not test k -cycle inequalities with $k \geq 4$ as it is well known that such constraints are implied by 3-cycle inequalities (see [13]). The more general k -cycle inequalities were presented only for the reason to simplify our explanation on how to generate all the cycle breaking inequalities (15).

However, as we show in the next subsection, we can do better.

3.1.3 Lifted k -Cycle Inequalities ($k \geq 3$)

In this subsection we show that the k -cycle inequalities can be strengthened by adding to the corresponding left hand sides, some terms on the x_{ij} variables.

As it has been suggested by Letchford [18], it is possible to lift the 3-cycle inequalities and obtain the following "lifted 3-cycle" inequalities

$$v_i^j + v_j^k + v_k^i + x_{ji} \leq 2, \quad \text{for } i, j, k = 2, \dots, n. \quad (26)$$

These inequalities have also been used by Sarin et al. [24].

Contrary to what happens with the unlifted inequalities, our results will show that these inequalities are quite effective to close the gap of the linear programming relaxation of the PCGDDL formulation. Unfortunately, separating all of these inequalities, although it can be performed in polynomial time by simple enumeration, is rather time consuming.

The question, now, is to know how we can lift the k -cycle inequalities for $k \geq 4$. First, notice that we can always obtain a lifted k -cycle inequality by using the same recursive argument presented in the previous subsection but, using lifted k' -cycle instead of k -cycle inequalities ($k' < k$). However, inequalities obtained in this way are not interesting as they are still redundant. Can we do better? Let us look at the case with $k = 4$. Let i, j, k and l be four distinct nodes of $V \setminus \{1\}$ and consider the lifted 4-cycle inequality

$$v_i^j + v_j^k + v_k^l + v_l^i + x_{ji} + x_{kj} + x_{ki} \leq 3, \quad (27)$$

which is easily seen to be valid by examining a few cases.

The question, now, is to know whether we can still add an extra x term on the left hand side of (27) and maintain the validity of the inequality. The following argument shows that it is not possible. In fact, if we choose any lifted 4-cycle inequality (27) and add it to the one corresponding to the reversed cycle, we obtain a clique inequality (7) for three nodes. (A similar situation happens by combining two appropriate lifted 3-cycle inequalities (26) leading to a clique inequality (7) for two nodes.) The form of the lifted 3- and 4-cycle inequalities suggest the following generalization for $k = 5$ in order to obtain a clique inequality on 5 nodes (following the lifted 3- and 4-cycle inequalities, we have chosen 4 nodes in the 5-cycle inequality and added x variables associated to arcs linking all of them in the reversed order)

$$v_i^j + v_j^k + v_k^l + v_l^m + v_m^i + x_{ji} + x_{kj} + x_{lk} + x_{ki} + x_{lj} + x_{li} \leq 4 \quad i, j, k, l, m \in \{2, \dots, n\}.$$

Again, one can easily see that the inequality is valid if $x_{lj} = x_{li} = 0$ (as it results from the combination of the lifted 4-cycle inequality for nodes i, j, k, m , together with the lifted 3-cycle inequality for nodes k, l, m and the 2-cycle equation for nodes k and m). However, it is not difficult to see that the inequality is not valid with the two extra terms. In fact, one interesting point of research is to find k -lifted cycle inequalities for $k > 4$ that cannot be obtained by combining lifted k' -cycle inequalities with $k' < k$.

We conclude this subsection by observing that the existence of lifted k -cycle inequalities shows that the precedence cycle breaking inequalities (15) with more than 2 sets cannot be facet defining. Indeed, we can use the process described in the previous subsection (with the lifted k -cycle inequalities instead of the k -cycle inequalities) to obtain lifted precedence cycle breaking inequalities. In fact, it will be interesting to study the polyhedral properties of the lifted precedence cycle breaking inequalities for 3 sets (by using (26)) and for 4 sets (by using (27)).

3.2 Cut-like inequalities

In this and the next subsections we will present several exponential sized sets of new cut-like and multicut-like inequalities relating the two sets of variables.

3.2.1 Simple Cut Inequalities

The first set of inequalities is directly based on the "meaning" of the equality $v_i^j = 1$. The equality means that i is before j , that is, i must be in the path from 1 to j in the tour. Then, we must have a path between node 1 and node i in the graph $G \setminus j$, a path between node i and node j in the graph $G \setminus 1$ and a path between node j and node 1 in the graph $G \setminus i$. Using the well-known fact that "there exists an (s, t) -path in a directed graph if and only if there is at least one arc in each (s, t) -cut", we obtain the following valid inequalities

$$v_i^j \leq x(\delta(V \setminus (S \cup \{j\}), S)) \quad i, j = 2, \dots, n \text{ and for all } S \subseteq \{1, \dots, n\}, 1, j \notin S, i \in S, \quad (28)$$

$$v_i^j \leq x(\delta(V \setminus (S \cup \{1\}), S)) \quad i, j = 2, \dots, n \text{ and for all } S \subseteq \{1, \dots, n\}, 1, i \notin S, j \in S, \quad (29)$$

$$v_i^j \leq x(\delta(V \setminus (S \cup \{i\}), S)) \quad i, j = 2, \dots, n \text{ and for all } S \subseteq \{1, \dots, n\}, i, j \notin S, 1 \in S. \quad (30)$$

All these inequalities can be separated in polynomial time by computing a minimum cut for each combination of two nodes i, j among $V \setminus \{1\}$ and by checking that the value of this minimum cut is greater than or equal to v_i^j . If this is not the case, we have found a violated inequality. Our results will indicate that the inclusion of such inequalities in our model leads to substantial improvements on the lower bounding values. One other advantage of including these inequalities is the substantial reduction in the overall CPU times. The reason for this is that with the cut inequalities, many fewer violated GDDL inequalities (11) seems to be generated. This may come from the fact that by combining adequately one inequality (28) and one inequality (29), we obtain an inequality that is slightly weaker than a GDDL inequality. It follows that we may find fewer violated GDDL inequalities and this can reduce the number of calls of the routine separating GDDL inequalities (whose complexity is greater than the complexity of the routine separating inequalities (28)-(30)).

Note that when $S = \{i\}$ the first inequality can be written as

$$v_i^j \leq \sum_{k=1, k \neq j}^{k=n} x_{ki}.$$

By using the assignment constraint (2) for node i we obtain the constraint (9) for the nodes i and j . Thus, the first set of cut-like constraints alone generalizes constraints (9).

Note that constraints (9) (and (10), by using (9) and the 2-cycle equations (20)), together with other inequalities, have been used before to obtain "projected" inequalities. Examples of this appeared in the proof of Theorem 1 to obtain inequalities (17) and (18) and later on, to create inequalities (23). In each of these three cases we can obtain more general inequalities by using (28)-(30) instead of (9) (or, by using (28)-(30) together with the 2-cycle inequality (20) instead of (10)).

Constraints (28)-(30) can be generalized in several ways. This is discussed in the following subsections

3.2.2 Path-Cut Inequalities

To motivate the path-cut constraints, consider the following generic constraint

$$v_i^j + v_j^k \leq 1 + x(A)$$

where A is a set of arcs. We want to find sets A leading to valid inequalities for the PCATS problem. Note, first, that the inequality is of interest only when $v_i^j = v_j^k = 1$ (otherwise it is trivially valid). But in this case, we have that i is before j and j is before k in the tour and we can say that the solution must contain an arc in any $(1, i)$ -cut in the graph $G \setminus \{j, k\}$ (that is, two nodes can be excluded from the adequate cut). This leads to the following valid inequalities

$$v_i^j + v_j^k - 1 \leq x(\delta(V \setminus (S \cup \{j, k\}), S)) \quad i, j, k = 2, \dots, n \text{ and for all } S \subseteq \{1, \dots, n\}, 1, j, k \notin S, i \in S. \quad (31)$$

By using the same circular reasoning that was used to derive inequalities (29) and (30), we can obtain the following additional 3 sets of inequalities

$$v_i^j + v_j^k - 1 \leq x(\delta(V \setminus (S \cup \{k, 1\}), S)) \quad i, j, k = 2, \dots, n \text{ and for all } S \subseteq \{1, \dots, n\}, 1, i, k \notin S, j \in S, \quad (32)$$

$$v_i^j + v_j^k - 1 \leq x(\delta(V \setminus (S \cup \{1, i\}), S)) \quad i, j, k = 2, \dots, n \text{ and for all } S \subseteq \{1, \dots, n\}, 1, i, j \notin S, k \in S, \quad (33)$$

$$v_i^j + v_j^k - 1 \leq x(\delta(V \setminus (S \cup \{i, j\}), S)) \quad i, j, k = 2, \dots, n \text{ and for all } S \subseteq \{1, \dots, n\}, i, j, k \notin S, 1 \in S. \quad (34)$$

We would like to point out that the previous inequalities do not capture the precedence relation information between the two nodes outside the cuts. We shall say more about this in Subsection 3.3 where we will present a more general set of multi-cut like inequalities.

As for inequalities (28)-(30), these inequalities can be separated in polynomial time by enumerating all combinations of three nodes and performing for each one a minimum-cut algorithm in an adequate graph. Clearly, enumerating all combinations of three nodes leads to a substantial increase in CPU times. On the other hand, our results will show that these inequalities further improve the lower bounds.

The reasoning used to obtain (31)-(34) can be generalized for an arbitrary number of nodes. Let i_1, \dots, i_q be a sequence of $q \leq n - 2$ nodes of $\{2, \dots, n\}$. Suppose that $i_0 = 1$. Let $r \in \{0, \dots, q\}$ and let S be any node set of $(V \setminus \{i_0, \dots, i_q\}) \cup \{i_r\}$ with $i_r \in S$. Let $S^C = (V \setminus \{i_0, \dots, i_q\}) \cup \{i_{r+1}\}$ where $i_{r+1} = i_0$ if $r = q$. We then have the following valid inequality

$$v_{i_1}^{i_2} + \dots + v_{i_{q-1}}^{i_q} - q + 2 \leq x(\delta(S, S^C)) \quad i_1, \dots, i_q \in \{2, \dots, n\} \text{ and} \quad (35)$$

for all $S \subset (V \setminus \{i_0, \dots, i_q\}) \cup \{i_r\}$ with $i_r \in S$.

Note that if $q = 2$ (resp. $q = 3$) these inequalities correspond to inequalities (28)-(30) (resp. (31)-(34)). Note also that for any given combination of nodes, it is possible to separate inequalities (35) in polynomial time by again computing a minimum-cut in an appropriate graph. Unfortunately, these inequalities do not seem to be separable in polynomial time when q is not fixed. Even if q is fixed, the naive separation algorithm which consist in enumerating each set of q nodes is practical only for small values of q .

3.2.3 3-Cycle-Cut Inequalities

The good results that we obtained by using the lifted 3-cycle inequalities (26) suggest that we try to generalize the lifted 3-cycle inequalities by using concept of cuts, described in the previous two subsections.

Following the motivation given for the path-cut inequalities (35), we want to obtain a general inequality of the form

$$v_i^j + v_j^k + v_k^i \leq 1 + x(A)$$

where A is a set of arcs. At first sight, the three terms on the left hand side suggest that we may be able to obtain an inequality that is stronger than the path-cut inequalities (35) for $q = 3$. However, this is not the case. First, note that the inequality is of interest only when the left hand side is equal to 2 (the inequality is trivially true when the left hand side is equal to 1, and by the 3-cycle inequalities, it cannot be equal to 3). The problem here, is that we do not know which v variable is going to be equal to zero and we can not use that information when choosing an adequate set A . However, if we ignore the fact that the tour starts and ends on the node 1, then the nodes i, j and k must be in this relative order whenever two v variables are equal to 1. It follows that we must have a path between i and j in the graph $G \setminus k$. We then have the following valid inequalities

$$v_i^j + v_j^k + v_k^i - 1 \leq x(\delta(V \setminus (S \cup \{k\}), S)) \quad i, j, k = 2, \dots, n \text{ and for all } S \subseteq V \setminus \{k\}, j \in S, i \notin S. \quad (36)$$

Using the circular argument stating that there must exist a path between j and k in the graph $G \setminus i$ and a path between k and i in the graph $G \setminus j$, leads to the same set of inequalities. Note that when $S = \{j\}$ in the inequality (36) and by using adequately the assignment equality (2), we obtain the lifted 3-cycle inequalities (26). These inequalities can also be separated in polynomial time by enumeration. The complexity of separating is the same as the one of separating the path-cut inequality (35) when $q = 3$. However, in practice, the separation of the 3-cycle-cut inequality is more time consuming. This is due to the fact that when we separate path-cut and 3-cycle-cut inequalities, the computation of a minimum cut is performed only if the left hand side is positive. As the left hand side of the 3-cycle-cut inequalities contains more terms, it has more chances to be positive and then a larger number of minimum-cut calls are performed when separating the 3-cycle-cut inequalities. Anyway, a few experiments performed by us indicate that separating the more general inequalities leads only to small improvements when compared with what we could achieve by using the lifted 3-cycle inequalities alone.

3.3 Multicut inequalities

In this subsection, we describe inequalities generalizing constraints of the previous subsection but involving more than two sets of nodes. Let i and j be two distinct nodes of $\{2, \dots, n\}$ and let (V_0, V_1, V_2) define a partition of V such that $1 \in V_0, i \in V_1$ and $j \in V_2$. We have the following "multi-cut" inequality

$$v_i^j + 1 \leq x(\delta(V_0, V_1 \cup V_2)) + x(\delta(V_1, V_2)). \quad (37)$$

Before showing its validity, we note that they generalize the three cut-like inequalities (28)-(30). In fact, when V_0 (resp. V_1, V_2) is reduced to a single node set, then inequality (37) is nothing else but inequality (29) (resp. (30), (28)). To see that the inequality (37) is valid, note first that if $v_i^j = 0$, then inequality (37) is a weaker version of the cut inequalities (8) and thus, it is valid. If $v_i^j = 1$, we must have two paths: one going from node 1 to node i and the second from node i to node j . Each of these paths must use at least one arc of $\delta(V_0, V_1 \cup V_2) \cup \delta(V_1, V_2)$. As the paths must be disjoint, the inequality is valid.

Unfortunately, it appears that such inequalities cannot be separated in polynomial time and for the moment we use the three simpler special cases (28)-(30) in our experiments.

The form of the previous inequalities suggest a stronger version of (35) for $q = 3$. Let i, j, k be three distinct nodes of $\{2, \dots, n\}$. Consider a partition V_0, V_1, V_2, V_3 of V with $1 \in V_0, i \in V_1, j \in V_2$ and $k \in V_3$, and the following inequality

$$v_i^j + v_j^k + 1 \leq x(\delta(V_0, V_1 \cup V_2 \cup V_3)) + x(\delta(V_1, V_2 \cup V_3)) + x(\delta(V_2, V_3)). \quad (38)$$

First, we can notice that this inequality is trivially valid when both v_i^j and v_j^k are equal to 0 as it is a weaker version of the cut inequalities (8). If only one of the variables v_i^j and v_j^k is equal to one, say v_i^j , the inequality is a weaker form of the inequality (37) (the one for the partition $V_0, V_1 \cup V_2, V_3$ and the two nodes j and k). Thus, this inequality is of interest only when $v_i^j = v_j^k = 1$. But in this case, a solution must contain a path from node 1 to node i , a path from node i to node j and a path from node j to node k . As these three paths have to be disjoint and each one uses at least one arc of the multicut, the inequality is valid.

Constraint (38) generalizes constraints (31)-(34). In fact, if we take, for instance, $V_2 = \{j\}$ and $V_3 = \{k\}$, and by combining the inequality with the inequalities (2) for nodes j and k , we obtain the inequality

$$v_i^j + v_j^k + x_{kj} - 1 \leq x(V_0, V_1)$$

that dominates the path-cut inequality (31). Indeed, the “lifted” term x_{kj} captures the “missed” information that k may be before j .

By reducing adequate sets to single node sets, we can also show that inequality (38) generalizes all inequalities (32)-(34).

As it happens with the simpler multi-cut inequalities (37), it appears that the inequalities (38) cannot be separated in polynomial time and for the moment, we use the weaker form (31)-(34) in our experiments.

We conclude this section by noting that it is not obvious how to generalize inequalities (37) and (38) for partitions of V with more than four sets as the straightforward generalization is not valid. As noted before, a similar situation happens with a straightforward generalization of the lifted 3-cycle and 4-cycle inequalities.

4 Systematizing ways of obtaining inequalities

Before presenting computational results, we will present a brief section where we systematize the different ways we have used to obtain inequalities for this class of formulations. We will also show that these ways suggest how to derive new inequalities for these formulations.

4.1 Lifting ATSP inequalities

Consider an inequality of the form $x(A) \leq b$, where A is a set of arcs and b is a constant, which is valid for the ATS problem. The idea is to look for valid inequalities for the ATS and PCATS problems by changing them into $v_p^i + x(A) \leq b + v_p^j$ for adequate pairs (p, i) and (p, j) . In fact, the GDDL inequalities were obtained in this way. We believe that this approach can be used to obtain valid inequalities for the PCATS problem from other valid inequalities for the ATS problem. As one other example, we consider the so-called D_k^+ and D_k^- inequalities (see [15]) which are known to be facet defining for the ATS problem. Let i, j and k be three distinct nodes of $V \setminus \{1\}$ and consider the following D_3^+ inequality

$$x_{ij} + x_{jk} + x_{ki} + 2x_{ji} \leq 2.$$

If we take now a node p from $V \setminus \{1, i, j, k\}$, we can lift the precedent inequality into the following inequality

$$v_p^i + x_{ij} + x_{jk} + x_{ki} + 2x_{ji} \leq 2 + v_p^j.$$

In fact this inequality means that, when the D_3^+ inequality is tight, if node p is before node i then it must also be before node j in the tour. A more general D_k^+ and D_k^- can be modified in a similar way provided that when the D_k^+ or D_k^- inequality is tight, then the two nodes i and j are necessarily linked by a path. We note that such inequalities were relevant for the best lower bounding schemes for the ATS problem (see [10]) and, modified “precedence constrained” versions of the D_k^+ and D_k^- inequalities were also used in the method described in [3].

4.2 Lifting LO inequalities

Similarly to what has been suggested in i), we can obtain inequalities in the (x, v) space by suitably adding x variables to valid inequalities for the LO polytope. Note that we did this before when lifting the 3-cycle and 4-cycle inequalities. We show next that more general Möbius ladder inequalities may also be lifted in a similar fashion. It is known that the so-called “simple” Möbius ladder inequalities are, under certain conditions, facet defining [14] for the LO polytope and that these inequalities belong to a more general class of valid inequalities that can be separated in polynomial time [6].

Consider the “smallest” Möbius ladder inequality [14]

$$v_{u_1}^{u_2} + v_{u_2}^{u_3} + v_{u_3}^{u_4} + v_{u_4}^{u_1} + v_{u_4}^{u_5} + v_{u_5}^{u_6} + v_{u_6}^{u_3} + v_{u_6}^{u_1} + v_{u_1}^{u_5} + v_{u_1}^{u_2} \leq 7. \quad (39)$$

where u_1, \dots, u_6 are six nodes different from node 1.

It is well known (see, for instance, [6]) that this inequality can be obtained by adding several 3-cycle inequalities and upper bounding inequalities on the v variables and by subsequently using integer rounding. Here we show that by using a similar procedure but using instead the lifted 3-cycle inequalities, we can obtain

a lifted (into the x space) Möbius ladder inequality. To simplify our proof we use the following redundant lifted 4-cycle inequalities

$$\begin{aligned} v_{u_1}^{u_2} + v_{u_2}^{u_3} + v_{u_3}^{u_4} + v_{u_4}^{u_1} + x_{u_2 u_1} + x_{u_4 u_3} &\leq 3, \\ v_{u_3}^{u_4} + v_{u_4}^{u_5} + v_{u_5}^{u_6} + v_{u_6}^{u_3} + x_{u_4 u_3} + x_{u_6 u_5} &\leq 3, \\ v_{u_5}^{u_6} + v_{u_6}^{u_1} + v_{u_1}^{u_2} + v_{u_2}^{u_5} + x_{u_6 u_5} + x_{u_2 u_1} &\leq 3. \end{aligned}$$

It is easy to see that such inequalities can be obtained from the lifted 3-cycle inequalities (26) and 2-cycle equations (20). Now, by adding all of them together with upper bounding inequalities $v_i^j \leq 1$ for the pairs of nodes $(i, j) = (u_2, u_3), (u_4, u_1), (u_4, u_5), (u_6, u_3), (u_6, u_1), (u_2, u_5)$; dividing the result by two; and rounding down the right-hand side of the obtained inequality, we obtain the following lifted Möbius ladder inequality

$$v_{u_1}^{u_2} + v_{u_2}^{u_3} + v_{u_3}^{u_4} + v_{u_4}^{u_1} + v_{u_4}^{u_5} + v_{u_5}^{u_6} + v_{u_6}^{u_3} + v_{u_6}^{u_1} + v_{u_2}^{u_5} + x_{u_2 u_1} + x_{u_4 u_3} + x_{u_6 u_5} \leq 7. \quad (40)$$

In fact, we can also lift the original Möbius ladder inequality in a different way by using the lifted 4-cycle inequalities

$$\begin{aligned} v_{u_1}^{u_2} + v_{u_2}^{u_3} + v_{u_3}^{u_4} + v_{u_4}^{u_1} + x_{u_3 u_2} + x_{u_1 u_4} &\leq 3, \\ v_{u_3}^{u_4} + v_{u_4}^{u_5} + v_{u_5}^{u_6} + v_{u_6}^{u_3} + x_{u_5 u_4} + x_{u_3 u_6} &\leq 3, \\ v_{u_5}^{u_6} + v_{u_6}^{u_1} + v_{u_1}^{u_2} + v_{u_2}^{u_5} + x_{u_1 u_6} + x_{u_5 u_2} &\leq 3. \end{aligned}$$

Adding these inequalities together with the inequalities $v_i^j + x_{ji} \leq 1$ for the pairs of nodes $(i, j) = (u_2, u_3), (u_4, u_1), (u_4, u_5), (u_6, u_3), (u_6, u_1), (u_2, u_5)$; dividing by two and rounding down the right-hand side, will give the lifted Möbius ladder inequality

$$v_{u_1}^{u_2} + v_{u_2}^{u_3} + v_{u_3}^{u_4} + v_{u_4}^{u_1} + v_{u_4}^{u_5} + v_{u_5}^{u_6} + v_{u_6}^{u_3} + v_{u_6}^{u_1} + v_{u_2}^{u_5} + x_{u_3 u_2} + x_{u_1 u_4} + x_{u_5 u_4} + x_{u_3 u_6} + x_{u_1 u_6} + x_{u_5 u_2} \leq 7. \quad (41)$$

The main question is to check whether more general Möbius ladder inequalities could also be lifted into (x, v) space in a similar way. In fact, it is not difficult to see that the two procedures mentioned above can be used to lift any Möbius ladder inequality that is obtained by combining an odd number k of 4-cycles (The case discussed above has $k = 3$). We also think that the second procedure described above can be used to lift “any” Möbius ladder inequality by adding the x_{ij} variable associated to every arc (j, i) that appears in only one cycle of the original Möbius ladder inequality.

4.3 Pure (x, v) inequalities (with the v variables following a pattern)

These inequalities are represented by the simple, path and cycle cut inequalities described in Section 3.2. As suggested by Fischetti [9], a different class is obtained by considering two disjoint sets of nodes A and B and by considering v_p^i variables with p taken from A and i taken from B . Let r be a positive integer and $(i_1, j_1), \dots, (i_r, j_r)$ be r couple of nodes of $\{2, \dots, n\}^2$ such that a node i_q does not appear in $\{j_1, \dots, j_r\}$ and a node j_q does not appear in $\{i_1, \dots, i_r\}$. Note that a same node can appear several times in $\{i_1, \dots, i_r\}$ or in $\{j_1, \dots, j_r\}$. Then, the following inequalities are valid

$$\sum_{q=1}^r v_{i_q}^{j_q} - r + 1 \leq x(\delta(V \setminus (S \cup \{j_1, \dots, j_r\}), S)) \quad \text{for all } S \subset \{2, \dots, n\} \setminus \{j_1, \dots, j_r\}, \{i_1, \dots, i_r\} \subseteq S, \quad (42)$$

$$\sum_{q=1}^r v_{i_q}^{j_q} - r + 1 \leq x(\delta(S, V \setminus (S \cup \{i_1, \dots, i_r\}))) \quad \text{for all } S \subset \{2, \dots, n\} \setminus \{i_1, \dots, i_r\}, \{j_1, \dots, j_r\} \subseteq S. \quad (43)$$

The interest of these inequalities is that by projecting out the v variables, we obtain the predecessor- and successor-inequalities introduced in [4].

4.4 Projecting from a MCF formulation

It is well known that so-called multicommodity flow formulations, usually, provide tight linear programming lower bounds for several classes of network design problems. Following [12], we could adapt the multicommodity flow formulation for the ATSP (see, for instance, [7]) by including the following linking constraints

$$\sum_{j=1}^{j=n} f_{kij} = v_k^i \quad i, k = 2, \dots, n.$$

We have solved the linear programming relaxation of such a formulation. For example, on the instance named ESC25, the gap is equal to 3.14% which is worst than our best result. On the other hand, for the

Name	V	P	A		B		C		D		E		F		Asch
			Gap	CPU	Gap	CPU	Gap	CPU	Gap	CPU	Gap	CPU	Gap	CPU	
ESC07	9	6	0.00	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00
ESC11	13	3	1.30	0	1.28	0	0.84	0	0.84	0	0.73	1	0.65	1	0.00
ESC12	14	7	0.36	1	0.00	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00
ESC25	27	9	6.51	26	5.55	24	2.56	28	2.40	23	1.76	120	1.92	24	2.32
ESC47	49	10	4.10	650	4.10	646	4.08	1011	4.08	719	3.46	3642	3.42	752	3.18
p43.1	44	9	0.04	11512	0.01	1741	0.01	2209	0.01	779	0.01	1909	0.01	7374	0.27
p43.2*	44	20	0.62	53761	0.39	8140	0.60	22258	0.39	7016	0.35	16160	0.29	13567	1.30
p43.3*	44	37	1.43	22620	0.77	5096	1.43	3461	0.77	3786	0.72	8943	0.67	4579	2.34
p43.4*	44	58	16.51	743	16.42	578	16.46	448	16.42	344	16.41	1742	0.16	440	32.67
ry48p.1	49	11	7.98	1500	3.40	1755	6.64	1179	3.03	1586	2.95	3835	3.00	2670	5.29
ry48p.2*	49	23	11.84	1840	7.03	1761	10.52	544	5.85	1643	5.76	3833	5.72	3478	9.29
ry48p.3*	49	42	14.84	1095	13.19	1029	14.02	898	12.80	1061	12.41	3719	10.94	2013	13.30
ry48p.4*	49	58	18.84	899	18.50	765	18.84	709	18.50	611	17.59	2351	12.87	720	13.61
ft53.1	54	12	5.87	3281	3.46	3888	3.66	2373	2.76	2214	2.66	6991	2.29	9699	4.58
ft53.2*	54	25	10.20	3284	7.02	3764	7.56	1876	6.84	1851	6.66	5878	5.89	10780	7.88
ft53.3*	54	48	16.82	1953	14.58	1975	16.49	1180	14.48	1221	13.50	4253	9.94	3293	11.18
ft53.4*	54	63	7.76	1473	7.52	1221	7.54	936	7.31	994	5.58	4181	3.26	836	3.84

Table 1: Results for the PCATS instances

instance ESC47, the gap is around 2.77% and thus better than our best results. These results indicate that it would be worth to study the projection of the linear programming feasible set of such a formulation on the (x, v) space as the projected inequalities might help us to strengthen our cutting plane methods. We note that some preliminar results on this have already been given in [12]. The inequalities given in [12] (and shown to be implied by the multicommodity flow formulation) are special cases of the multicut inequalities (38) (the "projected" inequalities correspond to the multicut inequalities when V_1 and V_3 are reduced to single node set). We have incorporated these special cases in our cutting planes, but apparently, no significant improvement has been noted. The results given by the linear programming relaxation of the multicommodity flow model suggest that other "projected" inequalities should be of interest.

5 Computational Results

In order to give an idea of the strength of the formulations and inequalities given in this paper, in this section we present a set of computational results on instances of the PCATS problem taken from the TSPLIB. In Table 1, the different entries correspond to

- Name: Name of the instance,
- V: Number of nodes of the instance,
- P: Number of precedence constraints,
- A: results given by PCGDDL_L,
- B: results given by PCGDDL_L with 2-cycle equations (20),
- C: results given by PCGDDL_L with simple cut inequalities (28)-(30),
- D: results given by PCGDDL_L with 2-cycle and simple cut inequalities,
- E: results given by PCGDDL_L with 2-cycle, simple cut and path-cut inequalities (35) with $q = 3$,
- F: results given by PCGDDL_L with 2-cycle, simple cut and lifted 3-cycle inequalities (26),
- Gap: gap of the considered formulation,
- CPU: CPU time in seconds needed to solve the corresponding formulation,
- Asch: gap of the root node lower bound obtained by Ascheuer et al. [3].

We use the formula $100 \times (gub - lb_{root}) / gub$ to compute the gap. In this formula, lb_{root} denotes the lower bound given by the linear programming relaxation of the considered formulation and gub denotes either the optimal solution or the best upper bound given by [3]. Instances indicated with an asterisk are instances for which the optimum is not known and the best upper bound value is used to compute the gap.

First, we note that these computational results confirm the different remarks made in the previous section about the practical efficiency of our different formulations and inequalities. In particular, we can see that adding the 2-cycle constraints permit us to reduce significantly the computational time as well as to improve the lower bounds. We can also remark that the addition of the simple cut inequalities either reduces the computational time effort or/and improves the lower bound values. However, the inclusion of the 2-path inequalities seems more dubious as although they improve the lower bound values, the substantial increase

of the computational times seems not to justify their use.

Our best formulation, with a good ratio between quality and CPU time, seems to be PCGDDL where we add the 2- and lifted 3-cycle inequalities as well as the simple cut inequalities. With this formulation, the corresponding results are given by the column F of Table 1, we are able, for most of the instances, to improve the lower bound obtained by Ascheuer et al. [3] at the root node of their Branch-and-Cut algorithm. In fact, for some instances, we even obtain a better lower bound than their global lower bound (that is the best lower bound they obtained after performing their Branch-and-Cut algorithm). In particular, we remark on the improvement done for the instance p43.4 where the gap is reduced from about 32% to 0.16%. This may be explained by the fact that our formulation contains a lifted version of all the precedence cycle breaking inequalities (15) while the Branch-and-Cut algorithm in [3] uses a heuristic to separate the original version of the precedence cycle breaking inequalities (15).

Finally, we have tried, for some instances, to extend the cutting plane algorithm developed into a Branch-and-Cut algorithm. We were able to obtain the optimal solution for the instance p43.4 (with a value of 83005 in 15282 seconds after exploring 289 nodes) that, as far as we know, has not yet been solved to optimality. Unfortunately, we have not been able, yet, to solve the two remaining p43 instances as the Branch-and-Cut algorithm is still too time consuming. We have to improve its implementation by combining exact separation algorithms of some inequalities with perhaps heuristic routines.

6 Conclusion

In this paper, we have studied formulations for the PCATS problem using binary precedence relation variables (besides the binary arc inclusion variables). The emphasis of this paper was on the use of exponential sized sets of cut-like constraints together with the corresponding polynomial separating routines.

On the other hand, our results have shown that the proposed methods produce lower bounds that dominate the best lower bounds known so far. With the new models we have been able to solve to optimality, one instance that has not been solved before.

Our results also indicate that it may be worth looking to new inequalities (see, for instance, Section 4) as in some cases, our lower bounds still need to be improved.

7 Appendix (Proof of Theorem 1)

To prove Theorem 1, we need the following lemma whose proof is given in [23]

Lemma 1 *Let $\bar{x} \in \mathbb{R}^m$ satisfying inequalities (7). Let $m_{ij} = \min\{|S| - 1 - x(E(S)) \mid S \subset \{2, \dots, n\}, i, j \in S\}$ for $i, j = 2, \dots, n$ with $i \neq j$. Then for all sequence of nodes $\{i_1, i_2, \dots, i_t\}$ of $\{2, \dots, n\}$ with $t \geq 3$ we have*

$$m_{i_1 i_t} \leq \sum_{r=1}^{t-1} m_{i_r i_{r+1}}.$$

We will show that the polyhedron P_x defined by (14), (7), (16)-(18) and (5) is the projection on the x space of the polyhedron P given by (9), (10), (11), (12), (13) and

$$x_{ij} \geq 0 \quad \text{for all } i, j \in V \setminus \{1\}. \quad (44)$$

First note that by (9), (10) and (5) we have that $0 \leq v_i^j \leq 1$ for all $i, j \in \{2, \dots, n\}$. It follows that, in P , we can replace equalities (12) and (13) by the inequalities

$$v_i^j \geq 1 \quad \text{for all } (i, j) \in B, \quad (45)$$

$$v_j^i \leq 0 \quad \text{for all } (i, j) \in B, \quad (46)$$

and maintain the same set of solutions.

We have shown before that the projection of P is included in P_x . To show the reverse inclusion, it is sufficient to show that for each solution \bar{x} of P_x , there exists a value assignment \bar{v} to the variables v such that (\bar{x}, \bar{v}) is a solution of P .

For all $i, j, k \in \{2, \dots, n\}$, let $S_{ijk} \subset V \setminus \{1, k\}$ be the set of nodes with $i, j \in S_{ijk}$ and such that

$$|S_{ijk}| - 1 - \bar{x}(E(S_{ijk})) = \min\{|S| - \bar{x}(E(S)), \text{ for all } S \subset V \setminus \{1, k\}, i, j \in S\}.$$

Note that we can assume that $S_{ijk} = S_{jik}$. It follows that a solution (\bar{x}, \bar{v}) satisfying the system

$$-v_j^j \leq -\bar{x}_{ij} \quad \text{for all } i, j \in \{2, \dots, n\}, i \neq j, \quad (47)$$

$$v_j^i \leq 1 - \bar{x}_{ij} \quad \text{for all } i, j \in \{2, \dots, n\}, i \neq j, \quad (48)$$

$$v_k^i - v_k^j \leq |S_{ijk}| - 1 - \bar{x}(E(S_{ijk})) \quad \text{for all } i, j, k \in \{2, \dots, n\}, i \neq j \neq k, \quad (49)$$

$$-v_j^j \leq -1 \quad \text{for all } (i, j) \in B, \quad (50)$$

$$v_j^i \leq 0 \quad \text{for all } (i, j) \in B. \quad (51)$$

will also belong to P. By Farka's lemma, this system has a solution if and only if we have

$$\sum_{i=2}^{i=n} \sum_{j=2, j \neq i}^{j=n} \sum_{k=2, k \neq i, j}^{k=n} (|S_{ijk}| - 1 - \bar{x}(E(S_{ijk}))) f_{ijk} + \sum_{i=2}^{i=n} \sum_{j=2, j \neq i}^{j=n} (1 - \bar{x}_{ij}) f_{ijj} - \sum_{i=2}^{i=n} \sum_{j=2, j \neq i}^{j=n} \bar{x}_{ij} f_{iji} - \sum_{(i,j) \in B} \alpha_{ij} \geq 0 \quad (52)$$

for all the solutions of the system

$$\sum_{j=2, j \neq i}^{j=n} f_{ijk} - \sum_{j=2, j \neq i}^{j=n} f_{jik} = 0 \quad \text{for all } k, i \in \{2, \dots, n\}, k \neq i, (k, i), (i, k) \notin B, \quad (53)$$

$$\sum_{j=2, j \neq i}^{j=n} f_{ijk} - \sum_{j=2, j \neq i}^{j=n} f_{jik} - \alpha_{ki} = 0 \quad \text{for all } k, i \in \{2, \dots, n\}, k \neq i, (k, i) \in B, \quad (54)$$

$$\sum_{j=2, j \neq i}^{j=n} f_{ijk} - \sum_{j=2, j \neq i}^{j=n} f_{jik} + \beta_{ik} = 0 \quad \text{for all } k, i \in \{2, \dots, n\}, k \neq i, (i, k) \in B, \quad (55)$$

$$f_{ijk} \geq 0 \quad \text{for all } i, j, k \in \{2, \dots, n\}, i \neq j \neq k, \quad (56)$$

$$\alpha_{ki} \geq 0 \quad \text{for all } (k, i) \in B, \quad (57)$$

$$\beta_{ik} \geq 0 \quad \text{for all } (k, i) \in B, \quad (58)$$

$$(59)$$

where variables $f_{iji}, f_{ijj}, f_{ijk}, \alpha_{ij}$ and β_{ij} are the dual variables associated respectively to inequalities (47)-(51).

Note that we can decompose the system given by (53)-(57) for each value of $k \in \{2, \dots, n\}$. Let (53k)-(57k) represent inequalities (53)-(57) restricted to the same value of k . Then, for a given $k \in \{2, \dots, n\}$, the system (53k)-(57k) defines a network flow system where inequalities (53k)-(55k) are flow balance inequalities. Note that the flow balance constraint for node k can be obtained by summing all the inequalities (53k)-(55k). This will give the inequality

$$\sum_{j=2, j \neq k}^{j=n} f_{jkk} - \sum_{j=2, j \neq k}^{j=n} f_{kjk} + \sum_{(i,k) \in B} \beta_{ik} - \sum_{(k,i) \in B} \alpha_{ki} = 0.$$

Note that this time, contrary to inequalities (53k)-(55k), the inequality consider the difference between the incoming flow and the outgoing flow at node k . Then, a solution of the system (53k)-(55k) describes a flow system where a node i such that $(k, i) \in B$ will be a source node if $\alpha_{ki} > 0$, and a node i such that $(i, k) \in B$ will be a sink node if $\beta_{ik} > 0$. Moreover, node k will be a source node if

$$\sum_{(i,k) \in B} \beta_{ik} > \sum_{(k,i) \in B} \alpha_{ki}$$

and a sink node if

$$\sum_{(i,k) \in B} \beta_{ik} < \sum_{(k,i) \in B} \alpha_{ki}.$$

Such a flow can be decomposed into elementary circuits C of flow with value f_C and/or into elementary paths P between a source and a sink node of flow with value f_P . We will now show that the contribution in (52) of each circuit or path starting with node k is positive, and the contribution of each path starting with a node different from k is at least f_P . As the summation of the flows on paths starting on a node different from k is equal to the summation of the α_{ki} for all i such that $(k, i) \in B$, we will have that, for each $k = 2, \dots, n$, the flow given by (53k)-(55k) will have a non-negative contribution in (52) and the proof will be complete.

Thus, consider first the contribution of a circuit C and suppose that the cycle is of length 2. If k belongs to the circuit (and let i be the second node), then the contribution of this circuit in (52) is equal to $(1 - \bar{x}_{ik} - \bar{x}_{ki})f_C$ which is non-negative by (7) for the set $S = \{i, k\}$. If k does not belong to the circuit, then let i and j be the two nodes of the circuit. Its contribution in (52) is thus $2f_C(|S_{ijk}| - 1 - \bar{x}(E(S_{ijk})))$ which is also non-negative by inequalities (7) for the set S_{ijk} .

Suppose now that the circuit contains at least three nodes. As before, we have the two cases whether or not the circuit contains the node k . Suppose first that the circuit is of the form $C = (k, i_1, i_2, \dots, i_t, k)$ with $t \geq 2$. In this case the contribution of C in (52) is

$$\left(\sum_{r=1}^{r=t-1} (|S_{i_r i_{r+1} k}| - 1 - \bar{x}(E(S_{i_r i_{r+1} k}))) + 1 - x_{i_t k} - x_{k i_1} \right) f_C.$$

By Lemma 1, this is greater than or equal to

$$(|S_{i_1 i_t k}| - \bar{x}(E(S_{i_1 i_t k})) - x_{i_t k} - x_{k i_1}) f_C.$$

As k does not belong to $S_{i_1 i_t k}$ this last expression is greater than or equal to

$$(|S_{i_1 i_t k} \cup \{k\}| - 1 - \bar{x}(E(S_{i_1 i_t k} \cup \{k\}))) f_C$$

which is itself non-negative.

Now, consider the case when the circuit does not contain node k , and let $C = (i_1, i_2, \dots, i_t, i_1)$ with $t \geq 3$. Its contribution in (52) is given by

$$\left(\sum_{r=1}^{r=t} (|S_{i_r i_{r+1} k}| - 1 - \bar{x}(E(S_{i_r i_{r+1} k}))) \right) f_C$$

with $i_{t+1} = i_1$. By inequalities (7), it is non-negative.

We will now consider the contribution of a path P . Suppose first that the path starts at node k , that is, P is of the form (k, i_1, \dots, i_t) , $t \geq 1$. It follows that i_t is a sink node and then (i_t, k) is a precedence constraint. If $t = 1$, the contribution in (52) of P is $-\bar{x}_{k i_1}$. As (i_1, k) is a precedence constraint, by (14), we have $\bar{x}_{k i_1} = 0$ and the contribution is non-negative. Now, if $t \geq 2$, then the contribution is

$$\left(\sum_{r=1}^{r=t-1} (|S_{i_r i_{r+1} k}| - 1 - \bar{x}(E(S_{i_r i_{r+1} k}))) - \bar{x}_{k i_1} \right) f_P.$$

By Lemma 1, this expression is greater than or equal to

$$((|S_{i_1 i_t k}| - 1 - \bar{x}(E(S_{i_1 i_t k}))) - \bar{x}_{k i_1}) f_P.$$

By inequality (18) for the nodes i_1, i_t, k , the precedence constraint (i_t, k) and the set $S_{i_1 i_t k}$, this expression is non-negative and so is the contribution of the path P .

Next, we will show that the contribution in (52) of the other paths P is at least f_P .

Consider a path $P = (i_1, \dots, i_t)$ not containing the node k . Then We have the two precedence constraints $k \prec i_1$ and $i_t \prec k$. The contribution of P in (52) is

$$\left(\sum_{r=1}^{r=t-1} (|S_{i_r i_{r+1} k}| - 1 - \bar{x}(E(S_{i_r i_{r+1} k}))) \right) f_P.$$

By Lemma 1, the contribution of P is greater than or equal

$$(|S_{i_1 i_t k}| - 1 - \bar{x}(E(S_{i_1 i_t k}))) f_P.$$

However, by inequality (16) for the nodes i_1, i_t, k , the set $S_{i_1 i_t k}$ and the two precedence constraint $k \prec i_1$ and $i_t \prec k$, we have $\bar{x}(E(S_{i_1 i_t k})) \leq |S_{i_1 i_t k}| - 2$ and thus the contribution of P in (52) is at least f_P .

Now, consider a path P whose last node is k , that is $P = (i_1, \dots, i_t, k)$ and then we have the precedence constraint $k \prec i_1$. If $t = 1$, the contribution of P will be $1 - \bar{x}_{i_1 k}$. As $\bar{x}_{i_1 k} = 0$ by inequality (14), the contribution of P is at least f_P . If $t \geq 2$, the contribution of P will be

$$\left(\sum_{r=1}^{r=t-1} (|S_{i_r i_{r+1} k}| - 1 - \bar{x}(E(S_{i_r i_{r+1} k}))) + 1 - \bar{x}_{i_t k} \right) f_P.$$

By Lemma 1, this is greater than or equal to

$$(|S_{i_1 i_t k}| - 1 - \bar{x}(E(S_{i_1 i_t k})) + 1 - \bar{x}_{i_t k})f_P.$$

By inequality (17) for the nodes i_1, i_t, k , the set $S_{i_1 i_t k}$ and the precedence constraint $k \prec i_1$, we have

$$\bar{x}(E(S_{i_1 i_t k})) + \bar{x}_{i_t k} \leq |S_{i_1 i_t k}| - 1$$

and thus, the contribution of P is at least equal to f_P .

Finally, consider a path of the form $(i_1, \dots, i_{p-1}, k, i_{p+1}, \dots, i_t)$. We then have the two precedence constraint $k \prec i_1$ and $i_t \prec k$. In fact, the contribution of such a path is nothing but the addition of the contribution of the two subpath given by (i_1, \dots, i_{p-1}, k) and (k, i_{p+1}, \dots, i_t) . As we have shown above, the first subpath has a contribution of at least f_P while the second has a non-negative contribution. It follows that the path P has a contribution of at least f_P in (52).

This complete the proof of the theorem.

References

- [1] N. Ascheuer. *Hamiltonian Path Problems in the On-line Optimization of Flexible Manufacturing Systems*. PhD thesis, Tech. Univ. Berlin, 1995.
- [2] N. Ascheuer, L. Escudero, M. Grötschel, and M. Stoer. A cutting plane approach to the sequential ordering problem (with applications to job scheduling in manufacturing). *SIAM Journal on Optimization*, 3(1):25–42, 1993.
- [3] N. Ascheuer, M. Jünger, and G. Reinelt. A branch-and-cut algorithm for the asymmetric traveling salesman problem with precedence constraints. *Computational Optimization and Applications*, 17(1):61–84, 2000.
- [4] E. Balas, M. Fischetti, and W. R. Pulleyblank. The precedence-constrained asymmetric traveling salesman polytope. *Mathematical Programming*, 68:241–265, 1995.
- [5] L. Bianco, A. Mingozzi, S. Ricciardelli, and M. Spadoni. Exact and heuristic procedures for the traveling salesman problem with precedence constraints, based on dynamic programming. *INFOR*, 32(1):19–31, 1994.
- [6] A. Caprara and M. Fischetti. 0 – 1/2 chvatal-gomory cuts. *Mathematical Programming*, 74:221–235, 1996.
- [7] A. Claus. A new formulation for the travelling salesman problem. *SIAM J. Algebraic Discrete Methods*, 5(1):21–25, 1984, March.
- [8] M. Desrochers and G. Laporte. Improvements to the Miller-Tucker-Zemlin subtour elimination constraints. *Operations Research Letters*, 10:27–36, 1991.
- [9] M. Fischetti, 2005. Personal communication.
- [10] M. Fischetti and P. Toth. A polyhedral approach to the asymmetric traveling salesman problem. *Management Science*, 43(11):1520–1536, 1997.
- [11] 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.
- [12] 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.
- [13] M. Grötschel, M. Jünger, and G. Reinelt. A cutting plane algorithm for the linear ordering problem. *Operations Research*, 32:1195–1220, 1984.
- [14] M. Grötschel, M. Jünger, and Reinelt Reinelt. Facets of the linear ordering polytope. *Mathematical Programming*, 33:43–60, 1985.
- [15] M. Grötschel and M. Padberg. Polyhedral theory. In E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, editors, *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, pages 251–305. John Wiley, 1985.
- [16] A. Langevin, F. Soumis, and J. Desrosiers. Classification of travelling salesman problem formulations. *Operation Research Letter*, 9:127–132, 1990.
- [17] E. L. Lawler, J. K. Lenstra, A. Rinnooy-Kan, and D. B. Shmoys. *The Travelling Salesman Problem*. Wiley, New York, 1985.

- [18] A. Letchford, 1998. Personal communication.
- [19] C. Miller, A. Tucker, and R. Zemlin. Integer programming formulations and the travelling salesman problems. *J. ACM*, 7:326–329, 1960.
- [20] A. Mingozzi, L. Bianco, and S. Ricciardelli. Dynamic programming strategies for the traveling salesman problem with time window and precedence constraints. *Operations Research*, 45:365–377, 1997.
- [21] Y-S. Myung. Projections of extended formulations with precedence variables for the asymmetric traveling salesman problem. *International Journal of Management Science*, 7(2), 2001.
- [22] M. Padberg and T.-Y. Sung. An analytical comparison of different formulations of the travelling salesman problem. *Mathematical Programming*, 52, 1991.
- [23] J. M. Pires. *Formulações para o Problema do Caixeiro Viajante Assimétrico e sua aplicação a um Problema de Desenho de Redes com Topologia em Forma de Anel*. PhD thesis, Faculdade de Ciências da Universidade de Lisboa, 2001.
- [24] 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.
- [25] M.T. Fiala Timlin and W.R. Pulleyblank. Precedence constrained routing and helicopter scheduling: Heuristic design. *Interfaces*, 22(3):100–111, May-June 1992.
- [26] 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.