Revisiting the Hamiltonian p-median problem: a new formulation on directed graphs and a branch-and-cut algorithm

Tolga Bektaş[1], **Luís Gouveia**[2]*, **Daniel Santos**[2]

[1]University of Liverpool Management School
Chatham Street, Liverpool, L69 7ZH, United Kingdom
t.bektas@liverpool.ac.uk

[2]Centro de Matemática, Aplicações Fundamentais e Investigação Operacional (CMAF-CIO)
Faculdade de Ciências da Universidade de Lisboa
Lisboa, C6 - Piso 4, 1749-016, Portugal
legouveia@fc.ul.pt, d.r.santos@outlook.com

Abstract

This paper studies the asymmetric Hamiltonian p-median problem, which consists of finding p mutually disjoint circuits of minimum total cost in a directed graph, such that each node of the graph is included in one of the circuits. Earlier formulations view the problem as the intersection of two subproblems, one requiring at most p, and the other requiring at least p circuits, in a feasible solution. This paper makes an explicit connection between the first subproblem and subtour elimination constraints of the traveling salesman problem, and between the second subproblem and the so-called path elimination constraints that arise in multi-depot/location-routing problems. A new formulation is described that builds on this connection, that uses the concept of an acting depot, resulting in a new set of constraints for the first subproblem, and a strong set of (path elimination) constraints for the second subproblem. The variables of the new model also allow for effective symmetry-breaking constraints to deal with two types of symmetries inherent in the problem. The paper describes a branch-and-cut algorithm that uses the new constraints, for which separation procedures are proposed. Theoretical and computational comparisons between the new formulation and an adaptation of an existing formulation originally proposed for the symmetric Hamiltonian p-median problem are presented. Computational results indicate that the algorithm is able to solve asymmetric instances with up to 171 nodes and symmetric instances with up to 100 nodes.

Keywords: Combinatorial optimization; Hamiltonian p-median; multi-cut inequalities; multi-depot routing; branch-and-cut algorithm.

*Corresponding author.

1 Introduction

The asymmetric *Hamiltonian p-median problem* (HpMP) consists of finding p circuits in a directed graph such that each node of the graph is in one and only one circuit and the total cost of the selected arcs is minimized. If $p = 1$ then the HpMP is the classical asymmetric traveling salesman problem (TSP).

The literature related to the HpMP is not as extensive as that of the TSP. To the best of our knowledge, the HpMP was first introduced by Branco & Coelho (1990) who present two formulations, one of set partitioning and the other defined on a directed graph in a similar vein to Fisher & Jaikumar (1981) for the vehicle routing problem, and describe several heuristics, assuming symmetric cost instances. Other studies include that of Glaab & Pott (2000) and Zohrehbandian (2007), either of which do not include a computational study, and more recently by Hupp & Liers (2013) and Gollowitzer et al. (2014) concerning polyhedral studies on the HpMP. Most of the algorithmic work on this problem is fairly recent (see, e.g., Gollowitzer et al. 2014, Erdoğan et al. 2016, Marzouk et al. 2016) and introduces an additional requirement that the HpMP should not admit solutions that include two-node circuits. Clearly, this requirement may lead to sub-optimal solutions to those of the original HpMP as initially defined by Branco & Coelho (1990), but it is attractive for modeling purposes, since the problem can then be modeled by using formulations based on undirected graphs with only a binary variable associated to each edge. Modeling two-node cycles on an undirected graph, although not as straightforward, is still possible by using $\{0,1,2\}$ variables or, equivalently, an additional set of binary variables that specifically consider this case, as is done in routing problems with multiple depots or in single-depot problems with multiple vehicles (see, e.g., Laporte et al. 1983, 1986, Araque G. et al. 1994, Belenguer et al. 2011, Benavent & Martínez-Sykora 2013). Our work will study the HpMP as was defined by Branco & Coelho (1990), that is, by allowing two-node circuits to exist, however, we will compare our proposed formulation with an adaptation of the formulation proposed by Erdoğan et al. (2016), which the authors proved to be effective for the undirected case in which two-node cycles are not allowed.

Our approach follows the paradigm given in Gollowitzer et al. (2014) in that we present models for the HpMP by partitioning the constraint set of the problem into two, namely one that guarantees that there are at most p, and the other to ensure that there are at least p circuits in a feasible solution, which we will refer to as $(\leq p)$ and $(\geq p)$ constraint sets, respectively. At this point, it is interesting to revisit the special case of the TSP, where one wishes to obtain solutions with $p = 1$ circuit, for which reason one would need to eliminate solutions with more than $p = 1$ circuit. This leads to the observation that the $(\leq p)$ constraints are generalizations of (and similar to) the subtour elimination constraints known from the TSP, and that the $(\geq p)$ constraints are the "odd" ones that are not easy to characterize. One of the contributions of our work is to show a connection between the $(\geq p)$ constraints and a different set of constraints that arise in multi-depot/location-routing problems, namely the so-called path elimination constraints. To do so, we describe extended formulations for the HpMP that use the concept of acting depots, which, although not an entirely new concept, can be explored in new ways. The formulation proposed in this paper can be viewed as an extended version of the acting depot formulations. In particular, we define additional (disaggregated) arc variables that describe whether an arc originates from, is destined to, or is disconnected to an acting depot, and uses an adaptation of the multi-cut (path elimination) constraints introduced by Bektaş et al. (2017). The new variables also allow for an effective way of dealing with two types of symmetries inherent in the HpMP, one induced by the use of the concept of acting depot, and the other resulting from two possible orientations of a given circuit.

The rest of the paper is structured as follows. Section 2 describes the concept of acting depot of a circuit and shows an application on a valid formulation for the HpMP in this context. The new formulation for the asymmetric HpMP is presented in Section 3. Section 4 describes alternative formulations for the problem, including an adaptation an existing formulation originally proposed for the symmetric Hamiltonian p-median problem, and presents theoretical comparison results. Section 5 has a particular focus on the various symmetry issues inherent in formulations using the concept of an acting depot and shows how they can easily be dealt with using the proposed formulation. Section 6 describes the branch-and-cut algorithm that uses the new formulation and its components, namely exact and heuristic separation procedures as well as a primal heuristic. This section presents computational results for asymmetric and symmetric instances, both to assess the effectiveness of the proposed algorithm and to numerically compare the new formulation with the alternative formulations described in this paper. Section 6 also presents comparison results with existing methods on symmetric instances in which two-node circuits are not allowed. Finally, conclusions are stated in Section 7.

2 Generic formulations for the HpMP

We define the HpMP on a directed graph $G = (V, A)$, with a set $V = \{1, \ldots, n\}$ of nodes, a set $A = \{(i, j) : i, j \in V,\ i \neq j\}$ of arcs, and a cost function c associated to the set of arcs. We say that a set of circuits covers the node set V, or is a cover of V, if each node is included in one and only one of the circuits. The objective of the HpMP is to find a minimum cost set of p disjoint circuits that covers V. For simplification, we will assume that G is a complete graph. Our results are applicable to incomplete graphs by simply not considering the pairs (i, j) such that $(i, j) \notin A$ in all of the mathematical expressions below. The formulations use the following notation: for any general one-index variable u we write $u(S) = \sum_{i \in S} u_i$; for any general two-index variable v we write $v(S) = \sum_{i,j \in S,\, i \neq j} v_{ij}$ and $v(S_1, S_2) = \sum_{i \in S_1, j \in S_2} v_{ij}$, where $S_1 \cap S_2 = \emptyset$. For singleton node subsets, say $I = \{i\}$, we write i instead of $\{i\}$.

This section presents two generic formulations for the HpMP, one defined in the space of the arc variables alone, and the other that uses an additional set of variables that differentiate the so-called acting depots from the client nodes.

2.1 A generic formulation with arc variables

In many network design problems, the most efficient formulations in practice are the ones that use arc variables alone, as they have fewer variables as compared to other formulations. Although such formulations usually include exponential-size sets of constraints, effective branch-and-cut algorithms can be devised if the separation of such inequalities can be done efficiently, as often is the case. The generic formulation presented below follows this line of thought and is an adaptation of the one proposed by Gollowitzer et al. (2014) for the symmetric HpMP to the asymmetric case. The model uses binary variables $x_{ij} = 1$ if arc $(i, j) \in A$ is used in any one of the p circuits and, $x_{ij} = 0$ otherwise.

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

$$\text{subject to:} \sum_{j \in V} x_{ij} = 1 \qquad \forall i \in V \tag{1}$$

$$\sum_{j \in V} x_{ji} = 1 \qquad \forall i \in V \tag{2}$$

$$\{(i, j) \in A : x_{ij} = 1\} \text{ forms at most } p \text{ circuits} \tag{3}$$

$$\{(i, j) \in A : x_{ij} = 1\} \text{ forms at least } p \text{ circuits} \tag{4}$$

$$x_{ij} \in \{0, 1\} \qquad \forall (i, j) \in A. \tag{5}$$

Clearly, any solution of (1)–(2) and (5), usually referred to as the assignment relaxation, corresponds to a set of disjoint circuits that cover V. However, such solutions may be composed of more, or of less, than p circuits. The generic constraints (3) ensure that there are at most p circuits in the solution, whereas the generic constraints (4) ensure that any solution is composed of at least p circuits.

In the formulation described by Gollowitzer et al. (2014), constraints (3) are modeled as generalizations of the cut inequalities known from the TSP, whereas inequalities (4) are in the form of cycle-elimination constraints. A polyhedral study based on these inequalities was initiated in Hupp & Liers (2013) and Gollowitzer et al. (2014). The two sets of inequalities can easily be adapted to the asymmetric HpMP. The main drawback, however, is that the separation of both sets of inequalities is NP-hard. This was proven by Gollowitzer et al. (2014) for the symmetric case. The complexity of the separation is still open for the asymmetric HpMP but there is no reason to suspect that it will be different to the symmetric case. This motivates the study of solution methods for the HpMP based on formulations using additional sets of variables such that the separation problem is polynomial, and which is one of the contributions of this paper.

2.2 A generic formulation extended with acting depot variables

The HpMP can be modeled by using a connection with multi-depot/location-routing problems that considers p nodes of V to be viewed as the acting depots of the p circuits. We now present another generic formulation for the HpMP that uses additional variables for the nodes acting as depots, namely a set of binary variables $y_i = 1$ if node $i \in V$ is an acting depot, and $y_i = 0$ otherwise. For simplicity, we will refer to a node i as a depot if $y_i = 1$, and as a client if $y_i = 0$, in a given solution.

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

$$\text{subject to: } \sum_{j\in V} x_{ij} = 1 \qquad \forall i \in V \quad (1)$$

$$\sum_{j\in V} x_{ji} = 1 \qquad \forall i \in V \quad (2)$$

$$\sum_{i\in V} y_i = p \quad (6)$$

$$\{(i,j) \in A : x_{ij} = 1 \text{ and } i \in V : y_i = 1\} \text{ contains no circuit with zero depots} \quad (7)$$

$$\{(i,j) \in A : x_{ij} = 1 \text{ and } i \in V : y_i = 1\} \text{ contains no circuit with two or more depots} \quad (8)$$

$$x_{ij} \in \{0,1\} \qquad \forall (i,j) \in A \quad (5)$$

$$y_i \in \{0,1\} \qquad \forall i \in V. \quad (9)$$

Constraints (6) ensure that there are exactly p depots in any feasible solution, while constraints (7) and (8) relate the y variables with the x variables and guarantee that in each circuit one and only one node has the corresponding y variable equal to 1. To see the connection with the HpMP, observe that if there exists a solution with less than p circuits then at least one of the circuits has more than one depot, and if there exists a solution with more than p circuits then at least one of the circuits will have zero depots. In this respect, constraints (7) and (8) correspond to (3) and (4), respectively.

In the context of multi-depot/location-routing problems, the generic inequalities (7) are modeled as classical subtour elimination constraints and the generic inequalities (8) as path elimination constraints, which are used to prevent paths between two depots. By using the acting depot concept, we can "translate" these inequalities to the context of the HpMP. In fact, the generic subtour elimination constraints (7) can be used to prevent circuits composed of only client nodes (i.e., nodes such that $y_i = 0$), whereas the generic path elimination constraints (8) prevent the formation of paths between two depots (i.e., nodes such that $y_i = 1$).

To illustrate this connection we present a valid formulation for the HpMP defined on the space of the x and y variables. To model the generic inequalities (7), we use the following set of (subtour elimination) constraints

$$x(S) \leq |S| - 1 + y(S) \qquad \forall S \subset V,\ S \neq \emptyset, \quad (10)$$

that correspond to the directed version of a set of inequalities proposed by Laporte et al. (1983) (see also Gollowitzer et al. 2014), which are based on the observation that if for a given subset S of nodes we have $y(S) = 0$, that is, there is no acting depot in the set S, then a circuit composed only of nodes of S cannot exist.

As for the generic inequalities (8), we can adapt, in a non-straightforward way, the multi-cut inequalities presented by Bektaş et al. (2017) for a multi-depot routing problem to the space of the x and y variables as follows:

$$x(S', i) + x(S', S) + x(i, S) \geq y_i + y(I) - |I| \qquad \forall \text{ partitions } S',\ S,\ I,\ \{i\} \text{ of } V, \quad (11)$$

such that $|I| = p - 1$. The reason why these constraints are not a straightforward adaptation of the inequalities from Bektaş et al. (2017) is that in the multi-depot routing problem there is a clear differentiation between two sets of nodes, namely that of candidate depots or that of customers. In the HpMP, there is no distinction between the two sets, and we need to select a set of candidate nodes to play the role of acting depots. This is the role of the subset I given in the previous expression. Observe that the inequality is of interest only when node i and all nodes in I are acting depots, in which the interpretation of the inequality is then the same as the one in Bektaş et al. (2017). That is, if there is another acting depot, say $j \in I$, in the circuit containing node i (which we assumed is an acting depot otherwise the inequality would be redundant), by selecting S' as the set of nodes which are in the path between i and j and by choosing S as the remaining nodes, that is, $S = V \setminus (S' \cup I \cup \{i\})$, we can easily see that a constraint (11) is violated.

By using the indegree constraints (2) for node i, it is easy to observe that if $S = \emptyset$ then constraints (11) become $y(I) + x(I, i) + y_i \leq 1 + |I|$. Thus, when $S = \emptyset$, these constraints assure that there can be no arcs between nodes chosen as depots, which is consistent with the interpretation of nodes acting as depots.

The observations above indicate that we obtain a valid formulation for the HpMP by using inequalities (11) to model the generic set (8) for all partitions S', S, I, $\{i\}$ of V such that $|I| = p-1$. Although not needed to obtain a valid integer formulation, we believe that the cases where $|I| < p-1$ might not be redundant in the associated linear programming relaxation. In fact, by using again the degree constraints, it can be seen that inequality (11) for $I = \{j\}$ is a stronger version of the following path elimination inequalities

$$y_i + x(P_{ij}) + y_j \leq |P_{ij}| \qquad \forall i, j \in V,\ i \neq j, \tag{12}$$

where P_{ij} is an elementary path linking nodes i and j and $|P_{ij}|$ is the number of nodes in the path.

There are, however, two clear drawbacks to using a model that uses arc variables and acting depot variables, such as the one defined above including the inequalities (10) and (11):

1. The first drawback is related to the separation of the inequalities (11). Even though the exact separation algorithm used by Bektaş et al. (2017) could be adapted to this case, it would no longer be polynomial in time given that the set of depots is no longer fixed and, thus, one would need to fix every possible set $I \cup \{i\}$ such that $|I \cup \{i\}| = p$. The other option would be to find a different separation algorithm that would determine the partition of $V \setminus \{i\}$ into the three sets S', S and I and that would simultaneously minimize the left-hand side and maximize the right-hand side of the inequalities (11). In our opinion, it is not clear that the resulting algorithm would be polynomial in time. Observe that if we resorted to heuristic separation algorithms, then this would defeat the purpose of using this formulation in the (x, y)-space since we might as well be using a formulation in the x-space based on the one proposed by Gollowitzer et al. (2014). One might argue that other adaptations of path elimination constraints could be used instead of inequalities (11). In any case, we believe that any "good" set of path elimination constraints relies on the fact that the depots and the clients are clearly identified.

2. The second drawback arises from a symmetry problem that is a consequence of this modeling approach that selects nodes to be the acting depots of the circuits, allowing too many alternative representations of a given solution. For example, a circuit $(i_1, i_2, ..., i_m, i_1)$, with $i_j \in V,\ \forall j \in \{1, \ldots, m\}$, can be represented in m different ways, depending on the selection of m possible acting depots. It is not clear how this symmetry problem can be addressed with formulations based on these two sets of variables.

In the next section, we present a new formulation for the HpMP that overcomes the two drawbacks identified above, and yields path elimination constraints stronger than (11) that we will show to be polynomially separable.

3 The PQR formulation

In the formulation discussed in Section 2.2 we used the binary y variables to indicate whether a node is an acting depot. In this section, we propose a model which "attaches" this information to the arc variables. More precisely, we propose three new sets of binary variables which distinguish between the cases of whether or not an arc (i, j) is used in one of the circuits and where either (i) i is a depot; (ii) j is a depot; and (iii) neither i nor j are depots. More precisely, for each arc (i, j), we create a binary variable p_{ij} which indicates whether or not arc (i, j) is used in one of the circuits where i is a depot; a binary variable q_{ij} which indicates whether or not arc (i, j) is used in one of the circuits where j is a depot; and a binary variable r_{ij} indicating whether or not arc (i, j) is used in one of the circuits where neither i nor j are depots. None of the three cases described above consider the situation where the two nodes i and j can be depots at the same time. Therefore, the definition of the three new sets of variables prevents solutions where two depots are directly linked. In other words, we ensure that if an (infeasible) path between two depots exists then at least one client node is included in that path.

In the next two sections we show how to model the two main sets of constraints, that is, the $(\geq p)$ constraints and the $(\leq p)$ constraints, by using the new sets of variables.

3.1 Modeling the $(\geq p)$ constraints

Recall that the $(\geq p)$ constraints ensure that the solution has at least p circuits which can be modeled by ensuring that there are no circuits with two or more acting depots. These constraints in the PQR formulation are defined as follows:

$$q(S', i) + r(S', S) + p(i, S) \geq y_i \qquad \forall i \in V,\ \forall \text{ partitions } (S', S) \text{ of } V \setminus \{i\}. \tag{13}$$

The proof of the validity of these multi-cut constraints is similar to the proof of constraints (11) described in the previous section.

Proposition 1. *The inequalities* (13) *are valid and eliminate circuits with two acting depots.*

Proof. Constraints (13) are clearly valid if $y_i = 0$. When $y_i = 1$, let (S', S) form a partition of $V \setminus \{i\}$ and suppose that $q(S', i) = r(S', S) = p(i, S) = 0$. Since i is a depot then it must be that $q(S, i) = 1 = p(i, S')$. Note that in the circuit of depot i there can be no more depots, hence all remaining arcs must be r arcs. But then it is not possible to complete the circuit for depot i since $r(S', S) = 0$.

To see why these constraints cut-off solutions in which there are two depots in the same circuit suppose that i is an acting depot ($y_i = 1$) and that there exists a node $j \in V \setminus \{i\}$ such that $y_j = 1$ and that i and j are in the same circuit. In this situation, there must exist at least one client node in the path from i to j since both nodes i and j are depots (this follows from the observation made regarding the definition of the p, q and r variables). If we consider that the set of nodes which are in the path from i to j are in S' and that the remaining nodes, except i and j, are in S, we obtain $p(i, S) = 0$, since the arc leaving i goes to S', and $q(S', i) = 0$, since the arc entering i comes from S. In addition, because j is a depot, then there is no r-arc incident to j, and regardless of whether $j \in S$ or $j \in S'$ we have $r(S', S) = 0$. □

An important remark regarding the new constraints (13) is that they are defined for all partitions (S', S) of $V \setminus \{i\}$, in contrast to the previous constraints (11) that were defined for all partitions (S', S, I) of $V \setminus \{i\}$. The difference in the number of subsets in the partition is relevant for the complexity of the corresponding separation problem and, in fact, as we shall show later on, constraints (13) can be separated in polynomial time by resorting to max-flow/min-cut computations. In a subsequent section we will show that constraints (13) imply a set of constraints similar to (11) and rewritten with the p, q and r variables. In other words, we will show that the additional set I in constraints (11) does not need to be explicitly considered in order to write inequalities in the space of the new variables since it is implicitly contained in their definition.

3.2 Modeling the $(\leq p)$ constraints

The $(\leq p)$ constraints ensure that any solution has at most p circuits which, as previously stated, can be modeled by preventing solutions containing circuits with no acting depots. From the definition of the p, q and r variables, a circuit will be composed of client nodes only, or equivalently have no acting depots, if and only if all arcs in the circuit are r-arcs. Hence, in order to prevent such a situation, we can use use any set of subtour elimination constraints that is known from the literature on the TSP (possibly with minor modifications motivated by the context) rewritten with the client-only variables r. For our model we use the well-known exponentially-sized set due to Dantzig et al. (1954), namely

$$r(S) \leq |S| - 1 \qquad \forall S \subset V. \tag{14}$$

Other sets of inequalities in place of (14) could be used instead. For instance, Bektaş (2012) uses the Miller-Tucker-Zemlin constraints (see Miller et al. 1960) for a related problem which have the advantage of being polynomial in number, but lead to substantially weaker LP relaxation bounds when compared to other formulations that use exponentially-sized sets of constraints such as (14).

The information provided by the variables of the PQR formulation permit us to strengthen constraints (14), as the following result shows.

Proposition 2. *The following inequalities are valid for the PQR formulation:*

$$p(i, S) + y(S \setminus \{i\}) + r(S) \leq |S| - 1 \qquad \forall S \subset V,\ \forall i \in S \tag{15}$$

$$q(S, i) + y(S \setminus \{i\}) + r(S) \leq |S| - 1 \qquad \forall S \subset V,\ \forall i \in S. \tag{16}$$

Proof. We start by observing that the following weaker set of constraints,

$$y(S \setminus \{i\}) + r(S) \leq |S| - 1 \qquad \forall S \subset V,\ \forall i \in S, \tag{17}$$

are valid based on the fact that $y(S \setminus \{i\}) \leq |S| - 1$ and on the observation that if $y_j = 1$ then $r(j, V) = r(V, j) = 0$ for any $j \in V$. We now lift inequalities (17) by adding the term $\alpha p(i, S)$ to the left hand side, where α is the lifting coefficient. The only case of interest is $p(i, S) = 1$. If $y(S \setminus \{i\}) = k \geq 0$, then $r(S) \leq |S| - k - 2$, implying that the maximum value of the lifting coefficient is $\alpha = 1$ and thus yielding (15). A similar lifting procedure using the term $\alpha q(S, i)$ yields (16). □

From here on out, we will use the inequalities (15) and (16) as the ($\leq p$) constraints in the PQR formulation, replacing the weaker set (14). Observe that the inequalities (15) and (16) can be equivalently written in cut-like form, respectively, as follows:

$$p(V,S) + r(S',S) \geq 1 - p(i,S') \qquad \forall \text{ partitions } (S',S) \text{ of } V,\ \forall i \in S \tag{18}$$
$$p(V,S) + r(S',S) \geq 1 - q(S',i) \qquad \forall \text{ partitions } (S',S) \text{ of } V,\ \forall i \in S, \tag{19}$$

which can be easily seen by appropriately using constraints which will be presented in the next section. The cut-like form of these inequalities suggests that they can be separated by resorting to max-flow/min-cut computations on a suitable graph. Such separation algorithms will be described in Section 6.2.1.

Inequalities (15) and (16) can be strengthened when $|S| > p$, as shown in the proposition below.

Proposition 3. *The following inequalities are valid for the PQR formulation:*

$$y(S) + r(S) \leq |S| - 1 \qquad \forall S \subset V : |S| > p. \tag{20}$$

Proof. The proof of this result follows from the fact that $y(S) \leq p \leq |S| - 1$ and, thus, we can apply the same reasoning used to prove the validity of inequalities (17). □

Inequalities (20), which can be written in cut-like form as

$$p(V,S) + r(S',S) \geq 1 \qquad \forall \text{ partitions } (S',S) \text{ of } V : |S| > p, \tag{21}$$

will be added as valid inequalities in the branch-and-cut algorithm. They only provide slight improvements in the algorithm, however, we can easily incorporate their separation in the separation algorithm of inequalities (15) and (16) by resorting to negligible-time computations.

3.3 The complete formulation

For completeness and clarity, we provide below the complete PQR formulation. Observe, first, that the "old" x and y variables and the "new" p, q and r variables are related by the following equalities (which are easily seen to be valid):

$$x_{ij} = p_{ij} + r_{ij} + q_{ij} \qquad \forall (i,j) \in A \tag{22}$$
$$y_i = \sum_{j \in V \setminus \{i\}} p_{ij} = \sum_{j \in V \setminus \{i\}} q_{ji} \qquad \forall i \in V. \tag{23}$$

The complete PQR formulation can be described by using only the p, q and r variables due to the above relations, however, for simplification, we will use the y variables as well. It is as follows:

$$\text{Minimize} \sum_{(i,j) \in A} c_{ij}\,(p_{ij} + q_{ij} + r_{ij})$$

$$\text{subject to:} \sum_{j \in V \setminus \{i\}} (p_{ij} + q_{ij} + r_{ij}) = 1 \qquad \forall i \in V \tag{24}$$
$$\sum_{j \in V \setminus \{i\}} (p_{ji} + q_{ji} + r_{ji}) = 1 \qquad \forall i \in V \tag{25}$$
$$y_i = \sum_{j \in V \setminus \{i\}} p_{ij} = \sum_{j \in V \setminus \{i\}} q_{ji} \qquad \forall i \in V \tag{23}$$
$$\sum_{i \in V} y_i = p \tag{6}$$
$$q(S',i) + r(S',S) + p(i,S) \geq y_i \qquad \forall i \in V,\ \forall \text{ partitions } (S',S) \text{ of } V \setminus \{i\} \tag{13}$$
$$p(V,S) + r(S',S) \geq 1 - p(i,S') \qquad \forall \text{ partitions } (S',S) \text{ of } V,\ \forall i \in S \tag{18}$$
$$p(V,S) + r(S',S) \geq 1 - q(S',i) \qquad \forall \text{ partitions } (S',S) \text{ of } V,\ \forall i \in S \tag{19}$$
$$p_{ij} \in \{0,1\} \qquad \forall (i,j) \in A \tag{26}$$
$$q_{ij} \in \{0,1\} \qquad \forall (i,j) \in A \tag{27}$$
$$r_{ij} \in \{0,1\} \qquad \forall (i,j) \in A \tag{28}$$
$$y_i \in \{0,1\} \qquad \forall i \in V. \tag{9}$$

In addition, and in order to simplify the proofs in the next section, we will also consider the two following sets of equalities,

$$\sum_{j\in V\setminus\{i\}} (q_{ij} + r_{ij}) = 1 - y_i \qquad \forall i \in V \tag{29}$$

$$\sum_{j\in V\setminus\{i\}} (p_{ji} + r_{ji}) = 1 - y_i \qquad \forall i \in V, \tag{30}$$

the validity of which can easily be proven by appropriately combining (23) with the degree constraints (24) and (25), respectively.

3.4 Inequalities in the (x, y)-space implied by the PQR formulation

Due to the relations (22) and (23) we can view the PQR formulation as an extended version of a formulation defined in the (x, y)-space. This raises the question of which $(\leq p)$ and $(\geq p)$ inequalities defined on the (x, y)-space can be obtained by projecting the feasible set of the linear programming relaxation of the PQR formulation onto the (x, y)-space. Obtaining the complete description of the projected polyhedron does not appear to be an easy task. It is, however, possible to establish two results concerning the way in which inequalities (10) and (11) are related to the projection.

Proposition 4. *The LP relaxation of the PQR formulation implies inequalities* (10).

Proof. Consider a set $S \subset V$. By adding $y(S) + p(i, S')$, where $i \in S$, to each side of (15) and by using the relations (23) between the y and the p variables for node i, we obtain

$$y(S) + y(S) + r(S) \leq |S| - 1 + y(S) + p(i, S').$$

Now, by using the relations (23) between the y and p variables for the first $y(S)$ term on the left-hand side, and the relations (23) between the y and q variables for the second $y(S)$ term on the left-hand side, we obtain

$$p(S) + p(S, S') + q(S) + q(S', S) + r(S) \leq |S| - 1 + y(S) + p(i, S').$$

From the relations (22) between the x and the p, q and r variables, we know that $x(S) = p(S) + r(S) + q(S)$. Hence, we can write the previous equation as follows:

$$x(S) \leq |S| - 1 + y(S) + p(i, S') - p(S, S') - q(S', S).$$

Finally, since the expression $p(i, S') - p(S, S') - q(S', S)$ is non-positive, we obtain an inequality that implies (10) for the same set S. A similar proof exist by starting with inequalities (16). □

Regarding the inequalities (11), first consider the following inequalities which can be seen as (11) rewritten with the p, q and r variables:

$$q(S', i) + r(S', S) + p(i, S) \geq y_i + y(I) - |I| \qquad \forall \text{ partitions } S',\, S,\, I,\, \{i\} \text{ of } V. \tag{31}$$

The following proposition shows that these inequalities are redundant in the PQR formulation.

Proposition 5. *The LP relaxation of the PQR formulation implies inequalities* (31).

Proof. Consider an inequality (13) for a node $i \in V$, a partition (S', S) of $V \setminus \{i\}$ and such that $S' = B' \cup C'$ and $S = B \cup C$, which can be written as:

$$q(B', i) + q(C', i) + r(B', B) + r(C', C) + r(B', C) + r(C', B) + p(i, B) + p(i, C) \geq y_i.$$

By defining $I = C \cup C'$, and by using $|C'| - y(C') \geq q(C', i) + r(C', C) + r(C', B)$ and $|C| - y(C) \geq r(B', C) + p(i, C)$, where both inequalities follow from (29) and (30), respectively, we obtain inequality (31) for the partition (B', B, I) of $V \setminus \{i\}$. □

In the proof of Proposition 5, the subsets C and C' correspond to sets of potential acting depots included in S and S', respectively. This shows that the variables of the PQR formulation implicitly hold information about the set I of potential acting depots. The following result therefore follows from Proposition 5.

Proposition 6. *The LP relaxation of the PQR formulation implies inequalities* (11).

Proof. It suffices to add adequate non-negativity constraints for the variables p, q and r to the left-hand side of (31) and to use equality (22) that relates the x with the p, q and r variables. □

4 Alternative formulations for the asymmetric HpMP

This section presents three alternative formulations for the asymmetric HpMP, namely (i) a compact formulation that is based on a 3-layered graph, (ii) an adaptation of an existing formulation of the problem proposed by Erdoğan et al. (2016) for the symmetric HpMP, and (iii) a formulation that provides a stronger LP bound than both (i) and (ii). We also present theoretical results comparing the three formulations and the PQR formulation, in the same way as in the work by Gollowitzer et al. (2014), that is, by differentiating between the cases ($\leq p$) and the ($\geq p$). This distinction follows from a finding suggested by preliminary computational experiments. In particular, let p' be the number of circuits in the assignment relaxation of the problem. Then, apart from a few exceptions, the models described in this paper yield the same optimal solution and LP bound without the ($\geq p$) constraints for when $p < p'$, and without the ($\leq p$) constraints when $p > p'$. Additionally, we find that either none or very few violated constraints of the type ($\geq p$) are identified when $p < p'$, and none or very few violated constraints of the type ($\leq p$) are identified if $p > p'$, in a branch-and-cut algorithm.

4.1 A compact formulation of the ($\geq p$) constraints and connections with the multi-cut inequalities (13)

The multi-cut inequalities (13) can be viewed as standard cut constraints in a corresponding 3-layered graph (see Bektaş et al. 2017) which allows for a better explanation of their underlying separation procedure as well as a derivation of a compact representation of the same set of constraints.

Consider a 3-layered graph $L = (V_L, A_L)$ where V_L is composed of three copies of each node in V. V_L is partitioned into three subsets, which are the three layers of L, each subset with a copy of each original node. The first layer and the third layer represent the copies of the nodes of graph G that are viewed as the acting depots. The two layers correspond to viewing them, respectively, as starting points and as ending points of one of the p circuits. The second layer represents the copies of nodes of the graph G that are viewed as the client nodes. The arc set A_L is also partitioned into three subsets. The first subset corresponds to the arcs going from the first layer to the second layer, with an arc existing for every existing p variable. The second subset corresponds to the arcs between the nodes in the second layer, which are represented by the r variables in the original graph. Finally, the second layer has arcs linking it to the third layer, with an arc existing if and only if a corresponding q variable exists.

To model the ($\geq p$) constraints of the HpMP in the 3-layered graph we need to guarantee the existence of p paths in L such that each path starts in a node in the first layer and ends in a node in the third layer, with the additional constraint that the start and end nodes of a path are copies of the same corresponding node in the original graph. Consider the following compact system of constraints based on the binary variables $z_{ij}^k = 1$ if an arc (i, j) is in the path from the copy of node k in the first layer to its copy in the third layer, and $z_{ij}^k = 0$ otherwise. The same variables also indicate whether or not arc (i, j) is used in the circuit of the acting depot k in the original graph. The ($\geq p$) constraints in the 3-layered may be modeled by the constraints below:

$$\sum_{j \in V, j \neq k} z_{kj}^k = y_k \qquad \forall k \in V \tag{32}$$

$$\sum_{j \in V, j \neq k} z_{jk}^k = y_k \qquad \forall k \in V \tag{33}$$

$$\sum_{j \in V, j \neq i} z_{ji}^k = \sum_{j \in V, j \neq i} z_{ij}^k \qquad \forall k \in V, \ \forall i \in V : i \neq k \tag{34}$$

$$z_{kj}^k \leq p_{kj} \qquad \forall k \in V, \ \forall (k, j) \in A \tag{35}$$

$$z_{jk}^k \leq q_{jk} \qquad \forall k \in V, \ \forall (j, k) \in A \tag{36}$$

$$z_{ij}^k \leq r_{ij} \qquad \forall k \in V, \ \forall (i, j) \in A \tag{37}$$

$$z_{ij}^k \in \{0, 1\} \qquad \forall k \in V, \ \forall (i, j) \in A. \tag{38}$$

Let 3I be the system defined by (32)–(37) and the following non-negativity constrains:

$$z_{ij}^k \geq 0 \qquad \forall k \in V,\ \forall (i,j) \in A \tag{39}$$

$$p_{ij} \geq 0 \qquad \forall (i,j) \in A \tag{40}$$

$$q_{ij} \geq 0 \qquad \forall (i,j) \in A \tag{41}$$

$$r_{ij} \geq 0 \qquad \forall (i,j) \in A \tag{42}$$

$$y_i \geq 0 \qquad \forall i \in V. \tag{43}$$

By using a result similar to that shown by Bektaş et al. (2017), one can prove that the projection of the polyhedron defined by 3I onto the space of the p, q, r and y variables is given by the multi-cut inequalities (13) and the non-negativity constraints (40)–(43), as the following proposition shows. The proof is similar to the one by Bektaş et al. (2017), however it is important to explicitly show it here since it will be fundamental to understand another result further on.

Proposition 7. *The projection of the polyhedron defined by 3I onto the space of the p, q, r and y variables is given by the multi-cut inequalities* (13) *and the non-negativity constraints* (40)–(43).

Proof. The result follows from the max-flow/min-cut theorem and the interpretation of the 3I system in the 3-layered graph L. The max-flow/min-cut theorem states that for each node k, y_k units of flow are sent from its copy in the first layer, say k^1, to its copy in the third layer, say k^3, without passing through its copy in the second layer (due to the flow-conservation constraints (34) which are not defined for the copy of node k in the second layer), with arc capacities given by the values of the variables p, q and r, if and only if every cut separating k^1 from k^3 has capacity with value at least y_k. This last requirement corresponds precisely to the constraints $q(S', k) + r(S', S) + p(k, S) \geq y_k$ for all partitions (S', S) of $V \setminus \{k\}$, which are the multi-cut inequalities (13). □

Consider the formulation obtained from the complete model of Section 3.3 by replacing the multi-cut inequalities (13) with the inequalities (32)–(38), which we denote by PQRz formulation. The result of Proposition 7 indicates that the two formulations provide the same LP bound.

The 3I system can be strengthened by replacing (37) with the stronger set of constraints,

$$\sum_{k \in V \setminus \{i,j\}} z_{ij}^k \leq r_{ij} \qquad \forall (i,j) \in A, \tag{44}$$

and define a system, denoted by $3I^+$, that includes fewer but stronger sets of constraints as compared to 3I. We denote by $PQRz^+$ the formulation PQRz where constraints (37) are replaced by (44). As we will see, our computational experiment shows that the $PQRz^+$ formulation is able to provide some improvements on the LP bounds when compared to the PQRz and the PQR formulations in a few cases. However, the $PQRz^+$ formulation includes many more variables and constraints than the PQR formulation and, thus, in practice, it has a worse performance. One topic worth of studying is to check whether the stronger system $3I^+$ implies a generalized version of the multi-cut inequalities (13), in the same way as was done by Bektaş et al. (2017) for the multi-depot routing problem, and use the resulting formulation in a branch-and-cut approach as it is suggested in this paper by using the multi-cut inequalities (13) alone. However, it is not clear how to generalize the multi-cut inequalities in the setting of the PQR formulation and, in addition, the reported improvements obtained by the $PQRz^+$ formulation when compared to the PQR formulation are not substantial. Thus, we will not explore this any further.

4.2 The x-v formulation and a comparison with the PQR formulation

In this section we present an adaptation of a model known from the literature for the symmetric HpMP. Besides the x_{ij} and y_i variables, the model also uses node-depot assignment binary variables v_i^j for nodes $i, j \in V$ such that $v_i^j = 1$ if node j is in the circuit of the acting depot i, and $v_i^j = 0$ otherwise. For a given subset of nodes $S \subset V$, we write v_S^j to denote $\sum_{i \in S} v_i^j$.

The following formulation, denoted by x-v, is an adaptation of the model by Erdoğan et al. (2016), which in turn is a strengthened version of that of Gollowitzer et al. (2014):

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

subject to: (1), (2), (5), (6) and (9)

$$y_i = v_i^i \qquad \forall i \in V \tag{45}$$

$$\sum_{j\in V} v_j^i = 1 \qquad \forall i \in V \tag{46}$$

$$v_i^j \leq v_i^i \qquad \forall i, j \in V : i \neq j \tag{47}$$

$$x(S', S) \geq v_{S'}^i \qquad \forall \text{ partitions } (S', S) \text{ of } V,\ \forall i \in S \tag{48}$$

$$v_S^i + x_{ij} \leq v_S^j + 1 \qquad \forall i, j \in V : i \neq j,\ \forall S \subset V \setminus \{j\} \tag{49}$$

$$v_i^j \in \{0, 1\} \qquad \forall i, j \in V. \tag{50}$$

In contrast with the model presented by Erdoğan et al. (2016), the model presented above uses arc variables x_{ij} to allow asymmetric costs to be modeled, whereas the original model uses edge variables. Additionally, normal constraint adaptations that are motivated from changing from an undirected model into a directed one result in constraints (48) and (49) that are simply directed counterparts of constraints in the earlier model. Another difference between the two models is, however, more relevant and results from the fact that two-node circuits are allowed in the version of the problem studied in this paper while they are not in the work by Erdoğan et al. (2016), as we mentioned in the introduction of this paper. In order to allow two-node circuits to exist, our adapted model uses single terms $+x_{ij}$ in constraints (49) in contrast to using terms $+x_{ij} + x_{ji}$ which would provide valid and stronger inequalities in the case where two-node circuits are not allowed.

As with the previous models, the x-v model can also be viewed as containing two specific sets of constraints, one preventing less than p circuits and the other preventing more than p circuits. Constraints (49) are the $(\geq p)$ constraints and they are a nice generalization proposed by Erdoğan et al. (2016) of simpler inequalities described in Gollowitzer et al. (2014). The cut constraints (48) model the $(\leq p)$ constraints and, as pointed out before, they are the directed version of inequalities presented by Erdoğan et al. (2016) and earlier by Gollowitzer et al. (2014).

Observe that we can relate the v and the z variables in the following way:

$$v_k^i = \sum_{j\in V} z_{ij}^k \qquad \forall k \in V,\ \forall i \in V. \tag{51}$$

The validity of these equalities is easy to establish and one can also see that the addition of these definitional constraints does not alter the LP value of the 3I$^+$ system. Observe that due to the flow conservation constraints (34), we can also derive the following set of equivalent "reversed" linking equalities:

$$v_k^i = \sum_{j\in V} z_{ji}^k \qquad \forall k \in V,\ \forall i \in V. \tag{52}$$

Clearly, in the presence of the flow conservation constraints (34), we only need to add either (51) or (52). In the following, and to simplify the proofs of the propositions appearing in the remainder of Section 4, we assume that the two equivalent linking equalities are added to the 3I$^+$ system. We show next that the LP relaxation of this "augmented" 3I$^+$ system implies constraints (49).

Proposition 8. *The 3I$^+$ system with the addition of either* (51) *or* (52) *implies the* $(\geq p)$ *constraints* (49) *of the x-v model.*

Proof. Consider two nodes $i, j \in V : i \neq j$ and a subset $S \subset V \setminus \{j\}$. We will prove the result assuming that $i \in S$. For the case in which $i \notin S$, the proof is similar.

If we add the flow conservation constraints (34) for the nodes in the subset S and, for node j, use the equality (51) and weaken the right-hand side of the resulting equation, we obtain:

$$v_S^j = \sum_{d\in S}\sum_{k\in V} z_{jk}^d = \sum_{d\in S}\sum_{k\in V} z_{kj}^d \geq \sum_{d\in S} z_{ij}^d.$$

Now, if we add $\sum_{d\in S}\sum_{k\in V:k\neq j} z_{ik}^d$ to both sides of the inequality above and use (51) for node i we obtain:

$$v_S^j + \sum_{d\in S}\sum_{k\in V:k\neq j} z_{ik}^d \geq v_S^i.$$

Finally, observe that

$$\sum_{d\in S}\sum_{k\in V:k\neq j} z_{ik}^d = \sum_{k\in V:k\neq j}\sum_{d\in S\setminus\{i\},d\neq k} z_{ik}^d + \sum_{d\in S\setminus\{i\}} z_{id}^d + \sum_{k\in V:k\neq j} z_{ik}^i,$$

and that

$$\sum_{k\in V:k\neq j}\sum_{d\in S\setminus\{i\},d\neq k} z_{ik}^d + \sum_{d\in S\setminus\{i\}} z_{id}^d + \sum_{k\in V:k\neq j} z_{ik}^i \leq r(i, V\setminus\{j\}) + q(i, S\setminus\{i\}) + p(i, V\setminus\{j\}) \leq x(i, V\setminus\{j\}).$$

In the expression above, the first inequality follows from the linking constraints (35), (36) and (44) and the last one from the equalities (22). Then, by using the degree constraints (1) we obtain

$$v_S^j + 1 \geq v_S^i + x_{ij},$$

which is the inequality (49) for node j and the subset S. □

The discussion above has two main implications in terms of LP relaxations:

- PQRz$^+$ dominates PQR (Proposition 7),
- If the ($\leq p$) constraints are removed from both formulations, then
  - PQRz$^+$ dominates x-v (Proposition 8)
  - It can be shown that there is no dominance between PQR and x-v, the details of which we omit here for reasons of brevity.

From a *practical* point of view, however, we will show in Section 6 that the reduction in the LP relaxation value going from the PQRz$^+$ model to the PQR model is much smaller when compared to weakening PQRz$^+$ to the x-v model.

4.3 Strengthening the PQRz$^+$ formulation (and comparing the ($\leq p$) constraints)

We now consider the situation when the ($\geq p$) constraints are removed from both PQR and x-v, and in this case, no dominance relationship can be established between the two formulations in terms of the LP relaxations, the details of which we also omit here for the sake of brevity. One way to arrive at a dominance relationship is to incrementally strengthen both formulations. Instead, we will describe in this section a strengthening of PQRz$^+$ that also dominates x-v. This is achieved by the addition of a new set of cut-like ($\leq p$) constraints to PQRz$^+$ as shown in the formulation below, which we denote by PQRz$^+$-v:

$$\begin{aligned}
&\text{Minimize} \sum_{(i,j)\in A} c_{ij}\,(p_{ij} + q_{ij} + r_{ij}) \\
&\text{subject to: } (6),\ (9),\ (23)\text{–}(28),\ (32)\text{–}(36), \\
&\qquad (38),\ (44),\ (45)\text{–}(47) \text{ and } (50)\text{–}(51) \\
&\qquad z^k(S', S) \geq v_k^i \qquad \forall \text{ partitions } (S', S) \text{ of } V : k \in S',\ i \in S. \qquad (53)
\end{aligned}$$

Since PQRz$^+$-v includes the 3I$^+$ system, then it implies the ($\geq p$) constraints of both PQR and x-v. In the remainder of this section, we will show that it also implies the ($\leq p$) constraints of both models.

The validity of the new constraints (53) is easy to establish. Their "reversed" counterpart

$$z^k(S, S') \geq v_k^i \qquad \forall k \in V,\ \forall \text{ partitions } (S', S) \text{ of } V : k \in S',\ i \in S, \qquad (54)$$

is redundant due to the flow-conservation constraints (34). Note also that when $|S| = 1$, the inequality (53) is implied by constraints (51).

Constraints (53) can be separated in polynomial-time by resorting to max-flow/min-cut computations, however, the PQRz$^+$-v model is hard to use in practice. The interest of this model is mainly theoretical since it allows us to compare the ($\leq p$) constraints of the PQR and the x-v models in an indirect way. We start by showing that the PQRz$^+$-v model implies the following new large set of cut-like constraints, that strictly contains (48):

$$x(S', S) \geq v_{L_1}^{i_1} + \ldots + v_{L_k}^{i_k} \qquad \forall \text{ partitions } (S', S) \text{ of } V, \ \forall \text{ partitions } (L_1, \ldots, L_k) \text{ of } S', \ i_1, \ldots, i_k \in S. \tag{55}$$

Proposition 9. *The linear programming relaxation of the PQRz$^+$-v model implies constraints* (55).

Proof. Let $k \geq 1$ and consider $i_1, \ldots, i_k$ distinct nodes of a set $S \subset V$ and $L_1, \ldots, L_k$ a partition of $S' = V \setminus S$. For each $m = 1, \ldots, k$ consider the following constraints (53)

$$z^k(S', S) \geq v_k^{i_m} \qquad \forall k \in L_m,$$

which can be equivalently written as

$$\sum_{j \in S'} z^k(j, S) \geq v_k^{i_m} \qquad \forall k \in L_m.$$

By adding all of these constraints for $m = 1, \ldots, k$, we obtain:

$$\sum_{m=1}^{k} \sum_{d \in L_m} \sum_{j \in S'} z^d(j, S) \geq \sum_{m=1}^{k} \sum_{d \in L_m} v_d^{i_m} = v_{L_1}^{i_1} + \ldots + v_{L_k}^{i_k}.$$

The right-hand side of the above constraint is equal to the right-hand side of constraints (55). As for the left-hand side, observe that it can be written as:

$$\sum_{m=1}^{k} \sum_{d \in L_m} \sum_{j \in S'} z^d(j, S) = \sum_{d \in S'} \sum_{j \in S'} z^d(j, S) = \sum_{d \in S'} z^d(d, S) + \sum_{j \in S'} \sum_{d \in S', d \neq j} z^d(j, S). \tag{56}$$

Finally, by using the relations (35) for the first term of the above third expression, the relations (44) for the second term of the above third expression and, lastly, the relations (22) between the x and the p, q and r variables we obtain:

$$\sum_{d \in S'} z^d(d, S) + \sum_{j \in S'} \sum_{d \in S', d \neq j} z^d(j, S) \leq p(S', S) + r(S', S) \leq x(S', S),$$

which completes the proof. □

The inequalities (55) include as a special case the ($\leq p$) constraints of the x-v model when $k = 1$. One interesting question, from a practical point of view, is to see how the efficiency of the inequalities (55) changes when the value of k increases. In fact, given that the left-hand side of the inequalities (55) is a cut-set, an exact separation algorithm for these inequalities consists in performing max-flow/min-cut computations in an adequate graph in which the nodes $i_1, \ldots, i_k$ have to be fixed as target nodes. Clearly, as k increases, there are more possibilities for fixing target nodes and, thus, many more max-flow/min-cut computations have to be performed to ensure exactness in the separation. Additionally, much like constraints (48) are directed counterparts of inequalities presented by Erdoğan et al. (2016) and Gollowitzer et al. (2014), it is possible to adapt inequalities (55) for $k \geq 2$ to the symmetric HpMP case.

Another interesting investigation is to see how the new formulation and the ($\leq p$) constraints of the PQR model are related. In fact, we can prove that the PQRz$^+$-v model implies constraints (18) and (19), which are the ($\leq p$) constraints of the PQR model.

Proposition 10. *The linear programming relaxation of the PQRz$^+$-v model implies constraints* (18) *and* (19).

Proof. Consider a partition (S', S) of V, a node $i \in S$ and, for each $d \in S \setminus \{i\}$, an inequality (53) as follows:

$$z^d(S' \cup \{d\}, S \setminus \{d\}) \geq v^i_d.$$

By adding these constraints for all $d \in S \setminus \{i\}$ we obtain:

$$\sum_{d \in S \setminus \{i\}} z^d(S' \cup \{d\}, S \setminus \{d\}) \geq v^i_{S \setminus \{i\}}.$$

Now consider the third expression under (56) in the proof of Proposition 9 in the case in which $k = 1$. If we add that expression to the above inequality we obtain:

$$\sum_{d \in S'} z^d(d, S) + \sum_{j \in S'} \sum_{d \in S', d \neq j} z^d(j, S) + \sum_{d \in S \setminus \{i\}} z^d(S' \cup \{d\}, S \setminus \{d\}) \geq v^i_{S'} + v^i_{S \setminus \{i\}} = 1 - y_i.$$

By manipulating the expression on the left-hand side of the above inequality, the details of which we omit for simplification, we can arrive at the following expression:

$$\sum_{d \in V \setminus \{i\}} z^d(d, S) + \sum_{j \in S'} \sum_{d \in V \setminus \{i\}, d \neq j} z^d(j, S \setminus \{d\}).$$

If we add $z^i(i, S)$ to both sides, we obtain:

$$\sum_{d \in V} z^d(d, S) + \sum_{j \in S'} \sum_{d \in V \setminus \{i\}, d \neq j} z^d(j, S \setminus \{d\}) \geq 1 - z^i(i, S'). \tag{57}$$

Thus, we can derive,

$$1 - p(i, S') \leq 1 - z^i(i, S') \leq \sum_{d \in V} z^d(d, S) + \sum_{j \in S'} \sum_{d \in V \setminus \{i\}, d \neq j} z^d(j, S \setminus \{d\}) \leq p(V, S) + r(S', S),$$

which are exactly the ($\leq p$) constraints (18).

Regarding constraints (19), we can start the proof from the “reversed” constraints (54) and use a similar reasoning. □

We conclude this section by noting that constraints (21) that are valid only for $|S| > p$ are not implied by the LP relaxation of the PQRz$^+$-v model. This can be easily seen by identifying solutions of the LP relaxation of the model that violate (21), the details of which we do not present here for reasons of brevity.

5 Symmetry-breaking constraints

This section discusses how to address two types of symmetries existing in the solutions obtained by the formulations described in Sections 3 and 4.

5.1 Symmetry of type I - Reducing the number of candidate acting depots in a circuit

This type of symmetry is induced by the m different ways a circuit $(i_1, i_2, ..., i_m, i_1)$ can be represented by selecting each node of the circuit as an acting depot. To address it, we use constraints that are motivated by the well-known idea (see, e.g., Campêlo et al. 2004) that the acting depot in any circuit should be the node with the lowest index to reduce the m possible representations into one, e.g., choosing node 2 as an acting depot for the circuit $(3, 2, 7, 6, 3)$. This idea can be easily implemented in the PQRz$^+$-v, PQRz$^+$ and the PQRz models presented in Section 4 as follows,

$$z^k_{kj} = 0 \qquad \forall k \in V,\ \forall (k, j) \in A : k > j \tag{58}$$

$$z^k_{jk} = 0 \qquad \forall k \in V,\ \forall (j, k) \in A : k < j \tag{59}$$

$$z^k_{ij} = 0 \qquad \forall k \in V,\ \forall (i, j) \in A : i < k \text{ or } j < k, \tag{60}$$

which collectively impose that the depot of each circuit should be the node with the lowest index by disallowing the use of arcs that do not satisfy this condition, and which can be partially adapted to the PQR formulation by considering the following set of symmetry-breaking constraints:

$$p_{ij} = 0 \qquad \forall (i,j) \in A : i > j \tag{61}$$

$$q_{ij} = 0 \qquad \forall (i,j) \in A : i < j. \tag{62}$$

Constraints (61)–(62) prevent some alternative representations of the same circuit, but do not guarantee that the depot of a circuit will be the node with the lowest index. For example, whereas they eliminate the solution $(3, 2, 7, 6, 3)$ since $p_{32} = 0$, they would not eliminate solutions such as the circuit $(3, 7, 2, 6, 3)$.

Constraints to deal with these type of symmetries have already been proposed by Gollowitzer et al. (2014) (and later used by Erdoğan et al. 2016) in the context of undirected versions of the x-v model. In fact, the x-v model can also benefit from using the following symmetry-breaking constraints

$$v_i^j = 0 \qquad \forall i, j \in V : \; j < i, \tag{63}$$

which we can see to be equivalent to the symmetry-breaking constraints (58)–(60) due to the the equalities (51) or (52).

It is not straightforward how to adapt a similar idea to the context of the PQR model. However, we can indirectly provide such a set of constraints by using (58)–(60) together with the projection result of Proposition 7. In particular, since there exists a path of value y_i from the copy of node i in the first layer to the copy of the same node in the third layer if the capacity of any cut separating the node copies of node i is at least y_i, restricting the arcs allowed to be in the path by removing arcs in the second layer according to constraints (58)–(60) also restricts the arcs allowed in the cut sets. This leads to the following restricted multi-cut inequalities that can be used for symmetry-breaking purposes,

$$q(S', i) + r(S', S) + p(i, S) \geq y_i \qquad \forall i \in V, \; \forall \text{ partitions } (S', S) \text{ of } V \setminus \{1, \dots, i\}, \tag{64}$$

which use all the information provided by the inequalities (58)–(60) and therefore ensure that the acting depot in any circuit will be the node with the lowest index. As the computational results will show, the computational times substantially decrease when the restricted multi-cut constraints (64) are used instead of (13). Although equalities (61)–(62) are not needed in the presence of the restricted multi-cut inequalities, we will still use them as they help to eliminate nearly half the number of p and q variables from the PQR formulation.

In many situations, symmetry-breaking constraints do not improve the value of the LP relaxation. However, for the case of the PQR model with the restricted multi-cut constraints (64) (and the PQRz$^+$-v, PQRz$^+$ and PQRz models with the symmetry-breaking constraints (58)-(60)) the computational results show that there are non-negligible improvements on the LP bounds for asymmetric instances.

5.2 Symmetry of type II - Eliminating reversed circuits

In the HpMP modeled on directed graphs with a symmetric cost function, one other symmetry arises for a circuit and its reverse, both of the same cost, and which therefore represent equivalent solutions even if they are structurally different. Observe that for circuits with only two nodes this problem is non-existent, thus, we focus on circuits which have at least three nodes. These equivalent solutions can pose a non-negligible problem if a large number, say m, of the p circuits in any solution are circuits with at least three nodes, since by combining both possible orientations for these $m \leq p$ circuits we have 2^m different representations of the same solution. We discuss below how such symmetries can be broken in the PQR formulation for circuits which have at least three nodes.

Observe that in a given circuit with at least three nodes, one of the two nodes adjacent to the acting depot has a smaller index than the other. In order to break these symmetries, we enforce that the first node visited after the acting depot is given a lower index than the node visited just before the acting depot. In the case of the circuits $c_1 = (2, 5, 7, 4, 2)$ and $c_2 = (2, 4, 7, 5, 2)$, for example, the circuit c_1 would be considered "infeasible", leaving circuit c_2 to be one of the p possible circuits. This idea has already been discussed for solving symmetric instances of the TSP using formulations based on directed graphs, and it can be introduced to the PQR formulation by the use of the following constraints:

$$\sum_{k \geq j} q_{ki} \geq p_{ij}, \qquad \forall i, j \in V, \; i \neq j. \tag{65}$$

Inequalities (65) state that if i is an acting depot and node j is visited immediately after i, then the node k visited just before i should be such that $k \geq j$. Notice that the case $k = j$ is required in order to avoid cutting-off solutions that are composed of two-node circuits. Furthermore, we can do even better and strengthen the inequalities (65) to:

$$\sum_{k \geq j} q_{ki} \geq \sum_{k \geq j} p_{ik}, \qquad \forall i, j \in V,\ i \neq j. \tag{66}$$

To see that this lifted version is valid, consider a pair of nodes i and j, and observe that: i) the right-hand side of these constraints is still at most 1; ii) if $p_{ij} = 1$, then the validity of the lifted constraints follows from the validity of the original constraints (65); iii) if $p_{ij} = 0$ and $p_{ik'} = 1$ for a node $k' > j$ then the validity of the constraints follow from the validity of the original constraint (65) for the case when $j = k'$; and iv) for $j = n$ the lifted constraints and the original constraints are the same.

Observe that this idea heavily relies on the fact that the acting depot information is in the arc variables, namely the p and the q variables of the PQR formulation. Hence, we believe that it is not as easy to implement this idea in formulations such as the x-v model. Nevertheless, the solutions identified are usually comprised of several circuits with two nodes, that is, m is usually much lower than p, and, thus, this type of symmetry is not as problematic as the one of the previous section.

We conclude this section by noting that constraints analogue to (66), that is, constraints in which the q variables are on the right-hand side and the p variables are on the left-hand side, can also be defined, however they can be shown to be redundant in the presence of constraints (23).

6 Computational results

In this section we present the numerical results from our computational experiment. Section 6.1 provides information on the instances used in the computations. In Section 6.2 we present details of the method proposed in this paper based on the PQR formulation. In Section 6.2.1 we present the separation algorithms for the set of constraints (13), the new restricted set (64), and the set of constraints (18) and (19). In Section 6.2.2 we present a basic primal heuristic that feeds feasible solutions to the branch-and-cut method by using information given by the LP relaxation in the branch-and-bound nodes. Section 6.2.3 describes the full branch-and-cut procedure used to obtain the numerical results. Finally, Section 6.3 shows the numerical results obtained with this branch-and-cut algorithm.

The last two subsections of this section consider additional computational results. Section 6.4 presents results to compare the models presented in this paper and in Section 6.5 we present results for our method based on the PQR formulation adapted to the case in which two-node circuits are not allowed and compare them to ones provided by Erdoğan et al. (2016) and Marzouk et al. (2016).

6.1 Instances

For the computational experiments reported in this paper we use two sets, A and B, of instances. The first set A is a subset of the very well-known TSPLIB symmetric TSP benchmark instances, namely dantzig42, swiss42, att48, gr48, hk48, eil51, berlin52, brazil58, st70, eil76, pr76, gr96, rat99, kroA100, kroB100, kroC100, kroD100, kroE100 and rd100. The number of nodes in these instances varies from 42 to 100. The set B of instances includes TSPLIB asymmetric TSP benchmark instances, namely ftv33 to ftv170, p43, ry48p, ft53, ft70 and kro124p, with a number of nodes which varies from 34 to 171. The complete description for the sets A and B of instances is available at `https://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/`. In the symmetric instances of set A where node coordinates are provided instead of explicit cost values, we determine the cost of an arc (i, j), and consequently of arc (j, i), by rounding up the Euclidean distance between i and j.

6.2 The branch-and-cut algorithm

This section describes the branch-and-cut algorithm devised to solve the problem and its ingredients, namely exact and heuristic separation procedures, and a primal heuristic to generate feasible solutions.

6.2.1 Separation procedures

These procedures are relevant to inequalities (13) and their symmetry-breaking restricted version (64), which model the ($\geq p$) constraints of the PQR formulation, and the inequalities (18) and (19), which model the ($\leq p$) constraints of the PQR formulation.

- *Separating inequalities* (13) *and* (64)

  The algorithm to separate the inequalities (13) is based on the 3-layered graph L view of the HpMP described in Section 4.1. Let (p^*, q^*, r^*, y^*) denote the current solution (either fractional or integer). For each arc of L, we set its capacity to the corresponding p^*, q^* or r^* value. Then, for every node $i \in V$, we remove node i from the second layer and determine the maximum flow v from the copy of node i in the first layer to the copy of the same node i in the third layer. If $v < y_i^*$ then we have found a minimum-cut with a value lower than y_i^* which partitions the nodes in the second layer into two subsets S' and S such that $S' \cup S = V \setminus \{i\}$. This cut unequivocally identifies a violated inequality (13), which is associated to the sets S' and S and node i. In order to separate the restricted symmetry-breaking version of the multi-cut inequalities (64), we use the same algorithm but in which the max-flow procedure is performed in a restricted 3-layered graph where not only node i but also every node j such that $j < i$ is removed from the second layer. These procedures are similar to the one presented by Bektaş et al. (2017) for the multi-cut constraints for the multi-depot routing problem. Observe that the procedures just described are exact, that is, if no violated inequality is found then the current solution (p^*, q^*, r^*, y^*) satisfies all the inequalities (13) or (64), respectively.

- *Separating inequalities* (18) *and* (19)

  To separate the inequalities (18) and (19), and additionally the slightly improved set (21), we devised a single procedure that attempts to find violated inequalities of all the three sets at once, for reasons that will be explained at the end of this subsection. Let (p^*, q^*, r^*, y^*) denote the current solution (either fractional or integer). We start by constructing a 2-layered graph which is, essentially, the 3-layered graph L without the third layer, to which we add an extra node s. This node s is connected to each node in the first layer by an arc with capacity equal to 1. The arcs linking the first with the second layer have as maximum capacity the corresponding value p^*, and the arcs in the second layer have as capacity the corresponding value r^*.

  For each node $i \in V$, we determine the maximum flow v from s to the copy of node i in the second layer. If, on the one hand, the maximum flow v is greater than or equal to 1, then there is no violated inequality (18), (19) or (21). On the other hand, if v is less than 1, then a minimum cut that partitions the nodes in the second layer into two subsets S' and S, such that $i \in S$, is obtained. If $|S| > p$, then we have found a violated inequality (21). If $|S| \leq p$, then we evaluate the expressions $v < 1 - p^*(i, S')$ and $v < 1 - q^*(S', i)$, to check whether an inequality (18) or (19) is violated by the current solution. If both $v < 1 - p^*(i, S')$ and $v < 1 - q^*(S', i)$, then we only consider the inequality (18) or (19) for which the violation is the greatest.

  The separation procedure described above is not exact for any of the three sets of inequalities (18), (19) or (21). Regarding the inequalities (18) or (19), note that the right-hand side depends on the set S'. More precisely, we do not take into account the values $p^*(i, S')$ and $q^*(S', i)$ when determining the minimum-cut. As for the inequalities (21), the additional condition that $|S| > p$ is also not taken into account during the minimum-cut computation. In other words, the proposed max-flow/min-cut procedure is, in fact, a relaxation (without the respective additional conditions) of the three min-cut procedures that we would need to solve in order to separate in an exact way the three sets of inequalities. Nevertheless, the proposed separation procedure is exact with respect to the following weaker set of inequalities

  $$p(V, S) + r(S', S) \geq 1 - y_i \qquad \forall \text{ partitions } (S', S) \text{ of } V,\ \forall i \in S, \tag{67}$$

  which are the cut-form of constraints (17).

  These inequalities can also model the ($\leq p$) constraints of the PQR model, however, they are dominated by (18) and (19) as we mentioned. Nevertheless, observe that when the algorithm stops, if $|S| > p$ we know that $v \geq 1 \geq 1 - y_i^*$ for any $i \in S$, and if $|S| \leq p$ we know that $v \geq \max\{1 - p^*(i, S'), 1 - q^*(S', i)\} \geq 1 - y_i^*$ for any $i \in S$. Thus, it is ensured that all the inequalities (67) are not violated, and, therefore, no client-only circuits will exist in the solution after the algorithm stops, since, as we have just pointed out, inequalities (67) are sufficient to model the ($\leq p$) constraints.

- *Heuristic separation procedure*

  The separation algorithms described above run in polynomial-time. Prior to applying them, however, we first attempt to find violated inequalities (13) or (64) and (18) or (19) (or (21)) by using a standard heuristic procedure that was developed for separating subtour elimination constraints in general problems (see, e.g., Fischetti et al. 1997), and which is adapted to our case. The use of the separation heuristic at the beginning of the whole separation procedure allows for the reduction in the computational times.

  The heuristic procedure consists in determining the connected components induced by the fractional values p^*, q^* and r^* by considering the arcs (i, j) such that $p^*_{ij} > 0$ or $q^*_{ij} > 0$ or $r^*_{ij} > 0$. Three cases can be obtained: i) a component is comprised of client nodes only, and hence, we can define a violated inequality (18) or (19) (or (21)); ii) a component contains two or more acting depots and, in this case, we can define a violated inequality (13) or (64) by finding a path that links two of the acting depots in the component; iii) a component contains only one acting depot and no violated inequality can be defined.

6.2.2 Primal heuristic

During the branch-and-cut algorithm we apply an heuristic procedure that generates a feasible solution by using the information given by the LP relaxation of a branch-and-bound tree node. The aim is to improve the current best known solution.

Given a cost function c', we start by finding the cheapest p arcs to start the p circuits. Then, by using a nearest neighbor type of criteria we insert the remaining nodes in the best possible circuit until we form p disjoint circuits that cover V. To further optimize these p circuits we apply local search operators that first swap nodes inside the circuits and then move nodes from their circuit to a different circuit. If the heuristic is applied by using the original cost function c, the solutions obtained will usually be of poor quality. Instead, we use the fractional values p^*, q^* and r^* at a given node of the tree to modify the cost of an arc (i, j) to $c_{ij} \times (1 - p^*_{ij} - r^*_{ij} - q^*_{ij})$. The reasoning for this modified cost function is that arcs for which the LP value is close to 1 will have a lower cost and hence have a higher chance of being chosen in the constructive part of the heuristic, that is, the feasible solution obtained before applying local search will be as "close" as possible to the current LP solution. The heuristic is applied at every five nodes during the first 250 nodes explored in the branch-and-bound tree, following which the frequency is increased to every 10 nodes.

6.2.3 Description of the overall algorithm

A standard branch-and-cut algorithm is applied to the PQR formulation. Before starting the algorithm, we determine the number of circuits in the assignment relaxation, which we denote by p', since we can use this value to guide the algorithm (note that the assignment problem is polynomially solvable, hence, it is reasonable to perform this first step). In each node of the branch-and-bound tree, we use the separation procedures described in Section 6.2.1. However, and motivated by an earlier discussion, the order in which we use them depends on the value of p'. If $p < p'$, we start with the separation procedure for the $(\leq p)$ constraints, whereas if $p > p'$ we start with the separation procedure for the $(\geq p)$ constraints. As explained previously, our initial computational tests lead to the conclusion that if $p < p'$ then it is likely (but not certain) that the optimal solution (and LP bound) can be obtained solely by separating the $(\leq p)$ constraints, and if $p > p'$ then we have the reverse conclusion that it is likely (but, again, not certain) that the optimal solution (and LP bound) can be obtained solely by separating the $(\geq p)$ constraints. Note that for the $(\geq p)$ constraints we use the restricted multi-cut constraints (64), seeing as they perform much better than their standard counterpart, except if stated otherwise.

A maximum of 20 inequalities are added in total before re-optimizing. However, if more than five violated inequalities of the first set of constraints to be separated are found, then the separation of the second set of constraints to be separated is skipped and the re-optimization phase begins immediately. In addition, we use the primal heuristic described in Section 6.2.2 and the symmetry-breaking constraints (66) when applicable.

The branch-and-cut algorithm was ran on an Intel Core i7-4790 3.6GHz processor with 8GB of RAM, within which CPLEX 12.6.1 Concert Technology for C++ was used. A time limit of 10800 seconds (three hours) has been imposed on the solution time. When obtaining the LP relaxation value for a given instance, we deactivate all pre-processing, heuristics (but not heuristic separations) and CPLEX's general-purpose cuts.

6.3 Numerical results

This section presents the results obtained with the branch-and-cut algorithm. We first analyse the effect of using the two types of symmetry-breaking constraints discussed in Section 5, and then provide results for both asymmetric instances and symmetric instances.

6.3.1 Showcasing the effect of using symmetry-breaking constraints

Following the discussion in Section 5, we provide some insight on the benefits of using symmetry-breaking constraints. For that we compare three variants of the algorithm:

1. The branch-and-cut (B&C) procedure described in Section 6.2.3 in which no symmetry-breaking constraints are used, that is, the standard multi-cut constraints (13) are used as ($\geq p$) constraints and, for symmetric instances, we do not use (66);

2. The B&C procedure described in Section 6.2.3 in which the restricted multi-cut constraints (64) are used as ($\geq p$) constraints but, for symmetric instances, we do not use (66);

3. The B&C procedure described in 6.2.3 in which all symmetry-breaking constraints are used (i.e., the standard approach).

Table 1 shows the comparison results and is divided into four parts. The first column shows the name of the instance, where ftv170 is an asymmetric instance and rat99 is a symmetric one. The next two columns show the number p' of circuits in the assignment relaxation and the value of p. For each variant of the B&C, we then report the LP relaxation value (LP), the time t_L (in seconds) required to obtain the LP value, the optimal value (OPT) and the time t (in seconds) required to obtain the optimal value.

			No symmetry-breaking (SB)				Only type I SB				Both type I and type II SB			
Name	p'	p	LP	t_L	OPT	t	LP	t_L	OPT	t	LP	t_L	OPT	t
ftv170	17	5	2662	1	2683	15	2662	1	2683	14	-	-	-	-
		10	2635	2	2636	8	2635	2	2636	8	-	-	-	-
		15	2631	1	2631	0	2631	1	2631	0	-	-	-	-
		20	2631	6	2631	0	2631	2	2631	0	-	-	-	-
		25	2631	9	2639	972	2635.14	16	2639	28	-	-	-	-
		30	2632.95	15	[2646, 2658]	10800*	2644.38	9	2658	44	-	-	-	-
		35	2643.57	87	[2664, 2705]	10800*	2671.92	56	2704	290	-	-	-	-
		40	2670.93	133	[2713, 2736]	10800*	2714.81	103	2736	508	-	-	-	-
		45	2719.42	169	[2779, 2799]	10800*	2772.13	603	2799	356	-	-	-	-
		50	2781.78	759	[2842, 2884]	10800*	2844.78	1153	2884	735	-	-	-	-
		55	2861.38	962	[2937, 3012]	10800*	2954.38	2083	3008	2470	-	-	-	-
		60	2989.2	1374	[3080, 3212]	10800*	3105.34	4248	3205	3648	-	-	-	-
		65	3198.03	5260	[3246, 3695]	10800*	3325.09	9716	[3428, 3432]	10800*	-	-	-	-
		70	3450.56	10800*	[3518, 3826]	10800*	3567.63	10800**	[3684, 3706]	10800*	-	-	-	-
		75	**	**	**	**	3727.48	10800**	[3767, 4091]	10800*	-	-	-	-
		80	**	**	**	**	3962.44	10800**	[4099, 4403]	10800*	-	-	-	-
		85	**	**	**	**	4644.5	1081	4777	7647	-	-	-	-
rat99	45	5	1228.5	1	1237	17	1228.5	1	1237	9	1228.5	1	1237	8
		10	1196.5	1	1212	41	1196.5	1	1212	58	1196.5	1	1212	68
		15	1178.17	1	1195	37	1178.17	1	1195	25	1178.17	1	1195	39
		20	1164.17	2	1184	321	1164.17	2	1184	308	1164.17	2	1184	172
		25	1154.38	0	1170	279	1154.38	0	1170	113	1154.38	1	1170	177
		30	1148	2	1159	100	1148	2	1159	51	1148	3	1159	41
		35	1144.44	8	1153	150	1144.44	8	1153	108	1144.44	2	1153	50
		40	1142.17	2	1145	76	1142.17	2	1145	14	1142.17	2	1145	15
		45	1142	0	1142	0	1142	0	1142	0	1142	0	1142	0

*Not solved to optimality within the time limit of three hours, ** Ran out of memory before the time limit

Table 1: Comparison results between the different types of symmetry-breaking constraints

The results for the asymmetric instance ftv170 show a substantial improvement by the use of the restricted multi-cut inequalities (64) that allow to optimally solve eight more instances, namely for $p = 30, \ldots, 60$ in increments of five and $p = 85$. A reduction in computational times is observed for other instances that were already solved to optimality, such as the instance with $p = 25$, for which the computational time decreased from about fifteen minutes to only 28 seconds. The results also show that the LP bounds in general increase, which also helps to explain the overall improved results.

As for the symmetric instance rat99, the use of symmetry-breaking constraints does not have a significant effect. This can be easily explained. In particular, there is no increase in LP bound when using the restricted multi-cut constraints (64) to replace their "weaker" counterpart (13). As for the use of the symmetry-breaking constraints (66), most of these solutions have many circuits with only two nodes which only allow one representation and, thus, the effect of the symmetry-breaking constraints is diminished.

To summarize, the results shown in Table 1 give us two indications. First, the restricted multi-cut constraints (64) are very effective for asymmetric instances and somewhat effective for symmetric instances. Note, however, that for the symmetric case the restricted multi-cut inequalities (64) should still be used instead of their "weaker" version (13) since the separation algorithm is very similar but is, in general, faster as the max-flow computations are done in smaller-sized auxiliary graphs. Second, the symmetry-breaking constraints of type II provide a negligible improvement, however, overall the average computational times are slightly reduced. For these reasons we include all symmetry-breaking constraints in the remainder of the experiments.

6.3.2 Results on asymmetric instances

This section presents the results obtained on the asymmetric test instances by using the standard (most effective) B&C procedure presented in Section 6.2.3. Given the rather large number of test instances, we only present results for a subset of the set B of instances in Table 2 in the main body of text, and present the results for the remainder of instances in set B in Table 8 of the appendix. Both tables follow the following format. For each instance, the first three columns indicate the name of the instance, the number p' of circuits in the assignment relaxation and the value of p used, respectively. The next four columns are relevant to the LP relaxation, namely the LP relaxation bound (LP), the computational time needed to solve the LP relaxation (t_L), in seconds, and the number of violated ($\leq p$) and ($\geq p$) constraints found when computing the LP bound, respectively. The last four columns provide information related to the optimal integer solution, where OPT and t are the optimal value of the corresponding instance and the corresponding computational time required, in seconds, and $\#(\leq p)$ and $\#(\geq p)$ are the total number of violated ($\leq p$) and ($\geq p$) constraints found during the branch-and-cut procedure, respectively. For each instance, we present results for values of p which are multiples of 5, starting with $p = 5$ and up to the maximum possible value of p for that instance (e.g., for an instance with 40 nodes, the value of p can go up to 20).

Name	p'	p	LP	t_L	$\#(\leq p)$	$\#(\geq p)$	OPT	t	$\#(\leq p)$	$\#(\geq p)$
ft70	10	5	38055.6	0	89	1	38120	3	851	62
		10	37978	0	7	0	37978	0	0	0
		15	38018.5	0	0	59	38033	1	0	114
		20	38275.2	0	0	185	38390	7	5	976
		25	39027.9	1	0	417	39233	9	0	1604
		30	40258.3	2	0	800	40539	11	0	1460
		35	42297	0	0	119	42908	2	0	166
ftv70	11	5	1804.78	1	137	11	1826	2	474	7
		10	1766	0	8	10	1766	0	0	0
		15	1769.5	0	0	61	1771	0	0	56
		20	1837	0	0	130	1841	1	0	86
		25	1954.41	0	0	241	1978	1	0	243
		30	2140.94	0	0	285	2210	2	0	469
		35	2496.63	0	0	383	2535	1	0	156
kro124p	32	5	35114.9	0	229	3	35435	13	1035	50
		10	34681.5	1	265	2	35010	15	1848	71
		15	34421.8	1	215	1	34799	68	4924	207
		20	34227.8	4	463	4	34433	62	5651	161
		25	34083.1	0	69	7	34267	87	6107	358
		30	33990	0	51	1	34002	1	52	6
		35	34050	0	0	13	34050	0	0	0
		40	34294.1	0	3	24	34310	0	0	9
		45	35082	0	0	28	35331	1	0	39
		50	36663	0	0	23	37541	1	0	53
ftv170	17	5	2662	1	50	0	2683	14	339	7
		10	2635	2	124	1	2636	8	148	0
		15	2631	1	27	7	2631	0	0	0
		20	2631	2	59	107	2631	0	0	0
		25	2635.14	16	2	440	2639	28	1	455
		30	2644.38	9	0	454	2658	44	38	825
		35	2671.92	56	0	1152	2704	290	1240	4304
		40	2714.81	103	0	1698	2736	508	0	3548
		45	2772.13	603	1	3385	2799	356	64	3758
		50	2844.78	1153	1	4950	2884	735	0	4825
		55	2954.38	2083	0	7090	3008	2470	10	8649
		60	3105.34	4248	0	10647	3205	3648	0	9703
		65	3325.09	9716	0	14291	[3428, 3432]	10800*	31	17838
		70	3567.63	10800*	0	16163	[3684, 3706]	10800*	0	14429
		75	3727.48	10800*	0	15918	[3767, 4091]	10800*	0	16046
		80	3962.44	10800*	0	15316	[4099, 4403]	10800*	0	15483
		85	4644.5	1081	0	5890	4777	7647	2	12350

*Not solved to optimality within the time limit of three hours

Table 2: Results for a subset of (asymmetric) instances in set B

The results suggest that the proposed B&C is able to effectively solve asymmetric instances of up to 100 nodes in at most around two minutes. The method was also able to solve most of the p values for instance ftv170, except for $p = 65, 70, 75, 80$. Observe that in these cases the value of p is much larger than p' which leads to the generation of too many violated restricted multi-cut inequalities (64).

6.3.3 Results on symmetric instances

Computational results for a subset of the set A of symmetric instances are shown in Table 3. Results for the remaining instances in set A are presented in Tables 9 and 10 provided in the appendix. The tables follow the same format as described in the previous section.

Name	p'	p	LP	t_L	$\#(\leq p)$	$\#(\geq p)$	OPT	t	$\#(\leq p)$	$\#(\geq p)$
pr76	36	5	91255.6	0	135	0	[96104, 97764]	10800*	79582	738
		10	85634	2	534	0	[90747, 91883]	10800*	120131	944
		15	81211.5	0	58	0	86380	1066	46964	178
		20	79365.2	0	50	0	82311	99	9705	35
		25	78332	0	52	1	82040	1757	39375	137
		30	77731.8	0	52	1	[80612, 81961]	10800*	54850	491
		35	77207.3	1	671	0	77973	1	159	4
rat99	45	5	1228.5	1	297	0	1237	8	1580	3
		10	1196.5	1	198	0	1212	68	4726	46
		15	1178.17	1	218	0	1195	39	5975	35
		20	1164.17	2	250	1	1184	172	25897	145
		25	1154.38	1	272	0	1170	177	18906	198
		30	1148	3	325	1	1159	41	6544	47
		35	1144.44	2	146	0	1153	50	9912	46
		40	1142.17	2	254	0	1145	15	8129	35
		45	1142	0	30	0	1142	0	0	0
kroB100	43	5	20282.3	1	220	0	[21023, 21093]	10800*	55636	4706
		10	19230.5	1	297	0	[19869, 20289]	10800*	120403	4976
		15	18438	2	286	1	[19198, 19364]	10800*	123221	3219
		20	17849.5	1	114	3	18727	8587	96014	1077
		25	17357.5	1	119	2	18132	8559	102279	1484
		30	17057.8	1	138	1	17606	4426	76271	921
		35	16890.3	1	117	5	17210	109	8919	86
		40	16834	112	2527	0	16921	80	8506	62
		45	16830	1	78	0	16838	2	329	2
		50	16830	0	0	0	18684	2	0	0
kroC100	45	5	19016	1	208	0	19998	7676	64310	1693
		10	18250	1	143	0	19077	1769	38756	1126
		15	17713.5	0	92	0	18454	1950	34709	907
		20	17344.8	1	93	1	17958	208	16261	323
		25	17116.3	2	245	0	17668	930	24979	499
		30	16959.5	1	115	2	[17421, 17539]	10800*	93350	1604
		35	16863.5	24	1499	0	17189	5820	54226	610
		40	16808.6	2	240	8	16926	74	8571	104
		45	16801	2	177	1	16801	0	0	0
		50	16801	0	0	0	17738	2	0	0

*Not solved to optimality within the limit of three hours.

Table 3: Results for a subset of (symmetric) instances in set A

The main conclusion we derive from the results of this section is that the B&C is worse in performance for symmetric instances as compared to asymmetric instances. One possible explanation for this behavior is the result that the multi-cut inequalities (13) of the PQR model do not improve the bounds given by the assignment relaxation in the case of symmetric costs. This result can be proved by using a similar argument to that given by Bektaş et al. (2017) for a multi-depot routing problem, and is therefore omitted here. In addition, and as mentioned in Section 6.3.1, whereas the restricted multi-cut constraints (64) substantially improve the results for asymmetric instances, the improvement is negligible for symmetric instances.

6.4 Comparisons between the PQR formulation and other formulations

In this section we provide some computational results with two purposes: i) evaluating the quality of the LP bounds given by the PQRz$^+$ and the PQRz$^+$-v formulations compared to the x-v and the PQR formulations; and ii) comparing the performance of the branch-and-cut algorithm based on the PQR model to the performance of a branch-and-cut algorithm based on the x-v model.

Table 4 shows the comparison results in terms of LP bound between the x-v, the PQR, the PQRz$^+$ and the PQRz$^+$-v models. In addition to the name of instance (namely the asymmetric instances ftv55 and ftv70 and the symmetric instance brazil58), the value of p' and the value of p, there is a fourth column indicating the optimal value of the instance. The remaining eight columns are divided into four groups of two, with each group showing the LP relaxation bound (LP) and the computational time needed to solve the LP relaxation (t_L) for each of the four models.

				x-v		PQR		$PQRz^+$		$PQRz^+$-v	
Name	p'	p	OPT	LP	t_L	LP	t_L	LP	t_L	LP	t_L
ftv55	10	5	1482	1462.67	0	1460.33	0	1460.33	3	1462.67	147
		10	1435	1435	0	1435	0	1435	2	1435	36
		15	1445	1438.33	3	1443.5	0	1443.5	4	1443.5	20
		20	1548	1453.67	11	1510	0	1525.5	4	1525.5	9
		25	1790	1515.98	101	1754	0	1769.5	5	1769.5	5
ftv70	11	5	1826	1806.88	1	1804.78	1	1804.78	204	1809.5	995
		10	1766	1766	0	1766	0	1766	28	1766	369
		15	1771	1766.67	4	1769.5	0	1769.5	48	1769.5	77
		20	1841	1783.44	64	1837	0	1839.5	35	1839.5	53
		25	1978	1830.07	364	1954.41	0	1963	40	1963	48
		30	2210	1896.28	1393	2140.94	0	2182	16	2182	15
		35	2535	1988.87	8552	2496.63	0	2499.5	27	2499.5	3
brazil58	27	5	20150	19103.3	0	18569.1	0	18569.1	1	19958	1584
		10	18407	17776.5	1	17369	0	17369	3	18407	1763
		15	17582	17191.5	1	16877	0	16877	1	17569	1014
		20	17017	16748.5	1	16652	0	16652	1	17005.5	982
		25	16583	16582	0	16573	1	16573	3	16583	166

Table 4: LP relaxation comparison between the x-v, the PQR, the $PQRz^+$ and the $PQRz^+$-v formulations

Given the observation stated at the beginning of Section 4, we will divide our evaluation of the results between the $p > p'$ case and the $p < p'$ case. In addition, we will not focus on comparing the PQR and the x-v models since we will provide an extended comparison afterwards.

Regarding the $p > p'$ case, the results of Table 4 show that the $PQRz^+$ formulation is able to provide improved LP bounds when compared to the PQR formulation. Additionally, we can see that the LP bound "loss" from going from the $PQRz^+$ formulation to the PQR is much lower than it is when going from the $PQRz^+$ formulation to the x-v model. Observe also that the $PQRz^+$-v formulation provides the same LP bound as the $PQRz^+$ formulation.

Regarding the $p < p'$ case, we can see that the $PQRz^+$-v formulation provides substantially improved LP bound when compared to the other models. In this case, however, it is the "loss" from going to the x-v model which is lower. Nevertheless, the difference is much greater than the ones reported for the $p < p'$ case, which indicates that there are many improvements still to be made for that case.

We now compare the PQR model and the x-v model in more detail. Observe that the x-v model has two exponentially sized sets of inequalities, namely inequalities (48) and inequalities (49). In the corresponding branch-and-cut, inequalities from these two sets were separated by using straightforward adaptations of the separation algorithms presented in Gollowitzer et al. (2014) and Erdoğan et al. (2016), respectively. The branch-and-cut algorithm for the x-v model implemented for this comparison also includes all the features used in the branch-and-cut algorithm for the PQR model, including the use of heuristic separation algorithms and a primal heuristic. Also, the symmetry-breaking constraints (63) were also incorporated.

Table 5 shows the comparison results in terms of their linear programming relaxation bounds and optimal integer solution times. The instances used for the comparison are the asymmetric instances ftv70 and kro124p and the symmetric instances kroB100 and kroC100. The format of this table is similar to the one of previous tables, with only the number of separated ($\leq p$) and ($\geq p$) constraints separated being omitted. The remaining comparison results of the PQR and the x-v models for the instances in sets A and B are available in Appendix in Tables 11, 12 and 13.

			PQR model				x-v model			
Name	p'	p	LP	t_L	OPT	t	LP	t_L	OPT	t
ftv70	11	5	1804.78	1	1826	2	1806.88	1	1826	3
		10	1766	0	1766	0	1766	0	1766	0
		15	1769.5	0	1771	0	1766.67	4	1771	25
		20	1837	0	1841	1	1783.44	64	[1790, 1990]	10800*
		25	1954.41	0	1978	1	1830.07	364	[1841, 2092]	10800*
		30	2140.94	0	2210	2	1896.28	1393	[1766, 2320]	10800*
		35	2496.63	0	2535	1	1988.87	8552	[2009, 2878]	10800*
kro124p	32	5	35114.9	0	35435	13	35212.9	6	35435	20
		10	34681.5	1	35010	15	34812.5	28	35010	41
		15	34421.8	1	34799	68	34523.6	32	34800	3320
		20	34227.8	4	34433	62	34280.5	37	34433	35
		25	34083.1	0	34267	87	34115.6	17	34267	43
		30	33990	0	34002	1	34002	2	34002	0
		35	34050	0	34050	0	34019.8	2	34050	3257
		40	34294.1	0	34310	0	34171.5	4	34310	8
		45	35082	0	35331	1	34487.2	14	[33991, 36206]	10800*
		50	36663	0	37541	1	34982.3	379	[33978, 46790]	10800*
kroB100	43	5	20282.3	1	[21034, 21093]	10800*	20495.5	1	21082	1995
		10	19230.5	1	[19872, 20289]	10800*	19635.9	42	20127	3929
		15	18438	2	[19204, 19364]	10800*	18944.9	34	19307	1775
		20	17849.5	1	18727	8587	18324.2	70	18727	305
		25	17357.5	1	18132	8559	17809.9	53	18132	93
		30	17057.8	1	17606	4426	17395.5	122	17606	21
		35	16890.3	1	17210	109	17048.2	77	17210	30
		40	16834	112	16921	80	16861.8	11	16921	8
		45	16830	1	16838	2	16830	1	16838	3
		50	16830	0	18684	2	16830	2	[16830, 21852]	10800*
kroC100	45	5	19016	1	19998	7676	19357.2	21	[19421, 21265]	10800*
		10	18250	1	19077	1769	18618.2	122	[18760, 19211]	10800*
		15	17713.5	0	18454	1950	18058	59	[18300, 18651]	10800*
		20	17344.8	1	17958	208	17670	75	[17878, 17958]	10800*
		25	17116.3	2	17668	930	17367.8	43	[17528, 17668]	10800*
		30	16959.5	1	[17434, 17539]	10800*	17118.9	57	[17379, 17476]	10800*
		35	16863.5	24	17189	5820	16940.5	42	17189	123
		40	16808.6	2	16926	74	16816.5	7	16926	23
		45	16801	2	16801	0	16801	1	16801	0
		50	16801	0	17738	2	16801	1	[16801, 21066]	10800*

*Not solved to optimality within the time limit of three hours

Table 5: Comparing the performance of the PQR model and x-v model

We will separate our evaluation on the reported results between the asymmetric instances and the symmetric instances. In the former case, in terms of obtaining the optimal integer solution, the results show that the CPU times of the PQR formulation are in general much lower than the ones obtained by the x-v model. The PQR formulation is able to solve all of the reported instances whereas the x-v model is unable to solve 6 of them. In addition, regarding the instances where both models are able to solve them, the times given by the PQR formulation are much lower, with exception to a few cases. In terms of LP values, the PQR formulation provides better bounds for the case with $p > p'$ and the x-v model for the case with $p < p'$. However, the difference in the LP values between the two models is much greater in the $p > p'$ case. Regarding the complete results for the instance set B, excluding instance ftv170 for which we did not test the x-v model, the PQR formulation is able to solve all of the 65 instances whereas the x-v model is unable to solve 19 of those 65.

Regarding symmetric instances, and in contrast with the asymmetric case, the dominance of the performance of the PQR model over the performance of the x-v model is not as impressive. In terms of the time needed to obtain the optimal integer solution, the PQR formulation is able to solve 16 out of 20 instances of Table 5 whereas the x-v model only 12. However the x-v model was able to solve 3 instances that the PQR formulation was not, namely instances kroB100 for $p = 5, 10, 15$. In addition, in some cases, the times needed by the x-v model to solve the instances are lower than the times needed by the PQR formulation. In general and on average, however, the PQR formulation still performs better even if the x-v model is more competitive than in the asymmetric case. In terms of LP values, we have similar conclusions as the ones for the asymmetric case. In particular, for the case with $p > p'$ the PQR formulation provides better LP bounds, whereas for the case with $p < p'$ the X-v formulation provides better LP bounds. The main difference of these results, and in contrast to the results of the asymmetric case, is that instances with $p < p'$ arise much more often in the symmetric case since the value of p' is in general quite large. This helps to explain why the x-v model is more competitive since, as already noted before for the asymmetric case, the cases with $p > p'$ are the cases where the PQR model provides better LP bounds. Regarding the complete results for the instance set A, the PQR formulation is able to solve 124 out of 134 instances whereas the x-v model only 109. We also observe that among the 134 instances, we chose "favorable" instances for the x-v model to show in Table 5.

To conclude the comparison between these two models, in general the PQR formulation performs much better than the x-v model does. This dominance is more noticeable for the asymmetric case, although, even in the symmetric case, the PQR model is able to solve many more instances than the x-v model does. We believe that the advantages of the PQR model which allow it to perform better for asymmetric instances are the efficiency of the separation of the multi-cut inequalities, the fact that we were able to include symmetry-breaking constraints in their restricted version and the good LP bounds for $p > p'$. For the symmetric instances, the case $p > p'$ is quite rare and, thus, one of three advantages is lost. Nevertheless, the other two appear to still allow the PQR formulation to outperform, in general, the x-v model. An important observation is that the performances of any model greatly depend on whether $p < p'$ or $p > p'$.

6.5 Results on symmetric instances where two-node circuits are not allowed

Most of the recent literature on the symmetric HpMP is based on undirected graphs and does not allow two-node cycles. In this section, we first show how the PQR formulation can be modified to handle this case, and then present the associated computational results.

Possibly the most trivial way to prevent two-node circuits is to add subtour elimination constraints for two nodes, which can be written using the p, q and r variables as follows:

$$p_{ij} + p_{ji} + r_{ij} + r_{ji} + q_{ij} + q_{ji} \leq 1 \qquad \forall i, j \in V,\ i \neq j. \tag{68}$$

The interpretation of the variables in the PQR formulation also allows for other "trivial" ways to model this situation as, for instance, shown by the following set of constraints:

$$\sum_{k \neq j} q_{ki} \geq p_{ij} \qquad \forall i, j \in V,\ i \neq j. \tag{69}$$

Note that constraints (68) state that an arc (i, j) cannot be used in both directions, whereas constraints (69) say that if an arc (i, j) is used in which i is an acting depot, then the arc that enters into i must come from a node k such that $k \neq j$.

It can be easily shown that, with respect to the LP relaxation, one set does not dominate the other, however their behavior is similar in the sense that they both perform quite bad computationally. In fact, some computational experiments while attempting to solve symmetric instances in which two-node circuits are not allowed by adding (68) or (69) (or even both) suggest that the PQR formulation with such constraints does not compete with current state-of-the-art methods . However, by using similar (but adequately modified) symmetry-breaking concepts to the symmetry-breaking constraints (66) presented in Section 5.2, we can derive a new set of constraints that dominates both (68) and (69) and permits the resolution of previously unsolved benchmark symmetric instances where two-node circuits are not allowed.

Observe that since every circuit must have at least three nodes, we can use an adaptation of the symmetry-breaking constraints (66) to impose this condition. In particular, consider the following inequalities:

$$\sum_{k > j} q_{ki} \geq \sum_{k \geq j} p_{ik}, \qquad \forall i, j \in V,\ i \neq j. \tag{70}$$

The difference when compared to the symmetry-breaking constraints (66) is that on the left-hand side we can remove the case $k = j$ since we do not want circuits with only two nodes. The most important observation is that in the regular case in which two-node circuit are allowed we observed that many solutions have a large number of circuits with two nodes and, thus, the use of the symmetry-breaking constraints (66) provided only a slight improvement in the computational times, however, in this case these symmetry-breaking constraints are extremely effective in order to prevent alternative representations of solutions and, additionally, implicitly ensure that every circuit has at least three nodes. We reinforce what we observed at the end of Section 5.2, namely that these constraints heavily depend on the fact that the acting depot is clearly identified in the arc variables, which is a characteristic of the p and the q variables.

The next table is similar to Table 6. It shows the relevance of the symmetry-breaking constraints (70) to improve the LP bounds and reduce the solution times for obtaining the optimal solution of instance rat99 for 5 values of p, for the symmetric case where two-node cycles are not allowed.

			No symmetry-breaking (SB)				Only type I SB				Both type I and type II SB			
Name	p'	p	LP	t_L	OPT	t	LP	t_L	OPT	t	LP	t_L	OPT	t
rat99	8	9	1203.54	18	1209.09	148	1203.54	27	1209.09	76	1203.98	2	1209.09	12
		14	1203.54	206	[1214.84, 1224.1]	10800*	1203.54	839	1224.1	3079	1214.61	5	1224.1	23
		19	1203.99	39	[1224.86, 1245.16]	10800*	1203.99	65	1245.16	7102	1235.13	23	1245.16	43
		24	1209.71	31	[1240.62, 1277.99]	10800*	1209.71	18	[1267.22, 1273.23]	10800*	1263.17	36	1273.23	44
		33	1238.64	47	[1287.55, 1469.58]	10800*	1238.64	55	[1309.4, 1397.26]	10800*	1343.4	32	1373.37	3003

*Not solved to optimality within the time limit of three hours

Table 6: Comparison results between the different types of symmetry-breaking constraints for symmetric instances where 2-node circuits are not allowed

For a comprehensive experimentation we used a subset of the instance set that Erdoğan et al. (2016) use, which is comprised of the TSPLIB symmetric instances of set A with the addition of instance u159 that has 159 nodes. Most of the results are given in Table 7, except for results for instances of smaller size which are shown in Table 14 in appendix. The tables follow the same format as the ones in the previous sections. However, the way that the values for p are chosen for these instances follows Gollowitzer et al. (2014) and Erdoğan et al. (2016).

Name	p'	p	LP	t_L	$\#(\leq p)$	$\#(\geq p)$	OPT	t	$\#(\leq p)$	$\#(\geq p)$
eil76	4	7	541.493	4	224	415	542.954	10	168	338
		10	543.297	5	110	660	545.021	38	681	1928
		15	547.805	12	224	1001	552.149	58	482	2859
		19	553.543	14	257	1172	563.955	133	658	4978
		25	572.847	30	323	1674	[590.395, 612.852]	10800*	1646	39658
pr76	8	7	99028.9	1	168	37	101401	5	851	28
		10	99263	1	91	63	101779	8	280	142
		15	100667	1	155	154	103663	34	1583	306
		19	102559	2	234	237	104482	7	123	210
		25	108023	4	324	166	110074	12	174	385
gr96	14	5	151513	1	312	0	153568	119	4565	553
		20	150339	18	684	494	151403	77	1624	825
rat99	8	9	1203.98	2	199	100	1209.09	12	187	104
		14	1214.61	5	329	231	1224.1	23	408	352
		19	1235.13	23	335	725	1245.16	43	382	868
		24	1263.17	36	406	1179	1273.23	44	354	995
		33	1343.4	32	286	1483	1373.37	3003	2866	14681
kroA100	13	10	19570.2	11	680	3	19900.9	151	6723	385
		14	19380.7	6	635	106	19637.5	95	4110	428
		20	19523.3	20	974	290	19868.6	30	681	440
		25	19815.7	21	920	356	20279.5	51	1275	583
		33	20542.7	106	1726	627	[21498, 23591]	10800*	3130	31147
kroB100	20	10	20444.8	3	507	4	20823.1	145	7759	860
		14	20396.3	7	679	2	20762.9	143	5317	607
		20	20414	18	1151	43	20660	75	2726	185
		25	20581.7	49	1588	448	20786.9	16	87	274
		33	21413.9	39	1306	336	[22204.6, 24968.4]	10800*	2587	33454
kroC100	13	10	19703.3	5	695	2	19923.3	98	5063	526
		14	19725.6	14	725	296	19938.8	67	4141	1006
		20	19853.1	39	1065	511	20135	55	1120	890
		25	20033.8	28	813	692	20428	436	3698	1959
		33	20286	30	899	672	[21536.6, 23759]	10800*	5350	25233
kroD100	14	10	19957	3	462	6	20270.6	48	2775	212
		14	19962.9	12	825	57	20267.2	42	1347	147
		20	20021.2	46	1086	495	20457	197	4308	2070
		25	20156.9	52	971	455	20671.2	160	2803	2090
		33	20669.6	64	1802	487	[21720.3, 22439.7]	10800*	4162	25151
kroE100	12	10	20651	3	451	1	20766.4	25	2164	297
		14	20641	5	467	213	20777.7	37	1529	449
		20	20715	18	859	444	20937.4	65	1677	1251
		25	20891.6	37	920	632	21174.9	92	2372	1114
		33	21485.3	40	1271	450	[22470.1, 22843.6]	10800*	5026	20484
rd100	14	10	7338.77	4	648	3	7524.08	147	5527	662
		14	7336.96	20	1142	28	7500.44	57	2449	105
		20	7354.05	33	1424	216	7537.98	101	4456	1131
		25	7419.3	37	1103	511	7555.83	42	721	699
		33	7670.45	35	1184	443	[7919.09, 8211.2]	10800*	3624	27475
u159	20	5	41079	3	236	1	41695	1194	8017	312
		30	41071.4	309	2667	81	41723	540	8432	306

*Not solved to optimality within the limit of three hours.

Table 7: Results for the symmetric case in which two-node circuits are not allowed

A comparison of our results with those reported by Erdoğan et al. (2016) shows that we have been able to optimally solve instances that Erdoğan et al. (2016) have not, namely the instance pr76 with $p = 15$, instance gr96 with $p = 20$, instance rat99 with $p \in \{14, 19, 24, 33\}$ and instance u159 with $p \in \{5, 30\}$. Conversely, there are other instances which we have been unable to solve to optimality that Erdoğan et al. (2016) have, namely instance att48 with $p = 16$ reported in the Appendix, instance eil76 with $p = 25$ and instance kroE100 with $p = 33$.

As for the branch-and-price algorithm described by Marzouk et al. (2016), we observe that this algorithm is not able to easily solve instances with small values of p, whereas the algorithm we propose here is. In particular, we have been able to optimally solve a number of instances that were not solved by Marzouk et al. (2016) (e.g., pr76 for $p = 15$, gr96 for $p = 5$ and kroA100, kroB100 and kroD100 for $p \in \{10, 14\}$). In contrast, some instances optimally solved by Marzouk et al. (2016) proved difficult to solve with our algorithm (for instance, eil76 for $p = 25$), in addition to other values of p that we have not tested which they have (note that Marzouk et al. (2016) test, for many instances, all values of p).

Although the PQR formulation was created to solve the original HpMP as described by Branco & Coelho (1990), the results provided for the case in which two-node circuits are not allowed suggest that the algorithm we proposed is able to compete with the current state-of-the-art methods by providing optimal solutions for instances that were previously unsolved. In relation to the difficulty of the problem, we concur with Erdoğan et al. (2016) that the instances with larger values of p appear to be the hardest ones to solve. As for instances with small values of p, those seem, in general, easier to solve by our method but appear to be harder to solve by the method described by Marzouk et al. (2016). We conclude by emphasizing that constraints (70) which play the role of both symmetry-breaking and two-node circuit elimination were key in the efficiency of the method based on the PQR formulation.

7 Conclusions

One of the main contributions of this paper was to describe a formulation for the Hamiltonian p-median problem (HpMP), primarily aimed at solving the problem as was originally defined on a directed graph. The new formulation has revealed rich theoretical insight as regards to the relationship to other alternative formulations of the problem. The new formulation also allows for ways to effectively eliminate symmetries caused by the use of acting depot variables. Using the symmetry breaking constraints in combination with separation algorithms and a primal heuristic, the paper described a branch-and-cut algorithm operating on the new formulation that allowed to optimally solve asymmetric instances with up to 171 nodes. Finally, the paper numerically showed that the new formulation could be used to optimally solve instances of the symmetric HpMP of up to 100 nodes, and described how it could be adapted to model the recently studied version of the problem, namely where two-node circuits are not allowed, enabling to optimally solve symmetric instances previously unable to be solved.

Acknowledgments

We thank the two reviewers whose comments have helped to improve the paper. 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. D. Santos was supported by a Programa de Bolsas de Doutoramento da Universidade de Lisboa grant.

References

Araque G., J. R., Kudva, G., Morin, T. L. & Pekny, J. F. (1994), 'A branch-and-cut algorithm for vehicle routing problems', *Annals of Operations Research* **50**(1), 37–59.

Bektaş, T. (2012), 'Formulations and Benders decomposition algorithms for multidepot salesmen problems with load balancing', *European Journal of Operational Research* **216**(1), 83–93.

Bektaş, T., Gouveia, L. & Santos, D. (2017), 'New path elimination constraints for multi-depot routing problems', *Networks* **70**(3), 246–261.

Belenguer, J.-M., Benavent, E., Prins, C., Prodhon, C. & Wolfler Calvo, R. (2011), 'A branch-and-cut method for the capacitated location-routing problem', *Computers & Operations Research* **38**(6), 931–941.

Benavent, E. & Martínez-Sykora, A. (2013), 'Multi-depot multiple TSP: a polyhedral study and computational results', *Annals of Operations Research* **207**(1), 7–25.

Branco, I. M. & Coelho, J. D. (1990), 'The Hamiltonian p-median problem', *European Journal of Operational Research* **47**(1), 86–95.

Campêlo, M., Corrêa, R. & Frota, Y. (2004), 'Cliques, holes and the vertex coloring polytope', *Information Processing Letters* **89**(4), 159–164.

Dantzig, G., Fulkerson, D. & Johnson, S. (1954), 'Solution of a large-scale traveling-salesman problem', *Journal of the Operations Research Society of America* **2**(4), 393–410.

Erdoğan, G., Laporte, G. & Rodríguez-Chía, A. M. (2016), 'Exact and heuristic algorithms for the Hamiltonian p-median problem', *European Journal of Operational Research* **253**(2), 280–289.

Fischetti, M., Salazar-González, J.-J. & Toth, P. (1997), 'A branch-and-cut algorithm for the symmetric generalized traveling salesman problem', *Operations Research* **45**(3), 378–394.

Fisher, M. L. & Jaikumar, R. (1981), 'A generalized assignment heuristic for vehicle routing', *Networks* **11**(2), 109–124.

Glaab, H. & Pott, A. (2000), 'The Hamiltonian p-median problem', *The Electronic Journal of Combinatorics* **7**(1), 42.

Gollowitzer, S., Gouveia, L., Laporte, G., Pereira, D. L. & Wojciechowski, A. (2014), 'A comparison of several models for the Hamiltonian p-median problem', *Networks* **63**(4), 350–363.

Hupp, L. & Liers, F. (2013), 'A polyhedral study of the Hamiltonian p-median problem', *Electronic Notes in Discrete Mathematics* **41**, 213–220.

Laporte, G., Nobert, Y. & Arpin, D. (1986), 'An exact algorithm for solving a capacitated location-routing problem', *Annals of Operations Research* **6**, 293–310.

Laporte, G., Nobert, Y. & Pelletier, P. (1983), 'Hamiltonian location problems', *European Journal of Operational Research* **12**(1), 82–89.

Marzouk, A. M., Moreno-Centeno, E. & Üster, H. (2016), 'A branch-and-price algorithm for solving the Hamiltonian p-median problem', *INFORMS Journal on Computing* **28**(4), 674–686.

Miller, C. E., Tucker, A. W. & Zemlin, R. A. (1960), 'Integer programming formulation of traveling salesman problems', *Journal of the Association for Computing Machinery* **7**(4), 326–329.

Zohrehbandian, M. (2007), 'A new formulation of the Hamiltonian p-median problem', *Applied Mathematical Sciences* **1**(8), 355–361.

A Additional computational results

The Appendix presents additional results obtained by the methods proposed in this paper.

A.1 Asymmetric instances with the PQR formulation

Name	p'	p	LP	t_L	$\#(\leq p)$	$\#(\geq p)$	OPT	t	$\#(\leq p)$	$\#(\geq p)$
ftv33	8	5	1193.11	0	98	4	1201	0	109	4
		10	1186	0	4	14	1187	0	0	10
		15	1247.8	0	0	14	1261	0	0	6
ftv35	8	5	1383.75	0	77	1	1387	0	114	0
		10	1382.5	0	0	6	1383	0	0	6
		15	1471.25	0	0	12	1480	0	0	20
ftv38	8	5	1440.75	0	56	2	1444	0	56	0
		10	1439.5	0	0	10	1440	0	0	10
		15	1520.5	0	0	29	1534	0	0	44
p43	16	5	187.833	0	108	1	199	7	1446	52
		10	152	0	262	4	158	3	1029	57
		15	148	0	125	0	148	0	0	0
		20	148	0	5	4	160	0	0	11
ftv44	9	5	1529.25	0	32	3	1543	1	275	15
		10	1522	0	0	8	1522	0	0	0
		15	1570.5	0	0	35	1573	0	1	36
		20	1691	0	0	40	1691	0	0	0
ftv47	11	5	1661.13	0	95	0	1672	1	128	8
		10	1652	0	20	7	1652	0	0	0
		15	1691.5	0	0	55	1703	1	0	90
		20	1815	0	0	48	1815	0	0	0
ry48p	20	5	13165.3	0	125	1	13497	1	417	23
		10	12735.8	0	85	1	12868	1	165	10
		15	12579.3	0	72	0	12677	1	120	16
		20	12517	0	8	0	12517	0	0	0
ft53	8	5	6004.07	0	57	0	6022	1	66	1
		10	5942	0	2	41	5942	0	0	0
		15	6027.75	0	0	57	6049	1	0	67
		20	6272.3	0	0	116	6305	0	0	72
		25	6973.5	0	0	43	7639	1	0	134
ftv55	10	5	1460.33	0	220	2	1482	1	354	14
		10	1435	0	65	27	1435	0	0	0
		15	1443.5	0	0	90	1445	0	0	87
		20	1510	0	0	92	1548	1	0	106
		25	1754	0	0	232	1790	0	0	242
ftv64	9	5	1729.17	0	134	14	1732	0	98	4
		10	1721	0	131	58	1721	0	0	0
		15	1721	0	34	54	1721	0	0	0
		20	1760.5	0	0	146	1767	0	0	128
		25	1878	0	0	136	1888	0	0	101
		30	2098.5	0	0	206	2140	1	0	129

Table 8: Results for the remaining (asymmetric) instances in set B

A.2 Symmetric instances with the PQR formulation

Name	p'	p	LP	t_L	$\#(\leq p)$	$\#(\geq p)$	OPT	t	$\#(\leq p)$	$\#(\geq p)$
dantzig42	20	5	579.5	0	131	0	604	4	1977	17
		10	545	0	24	0	573	3	1033	11
		15	534	0	79	0	548	1	333	4
		20	532	0	3	1	532	0	0	0
swiss42	20	5	1123.17	0	83	0	1155	1	294	10
		10	1045	0	26	1	1084	2	1103	25
		15	1021.83	0	62	2	1034	1	77	3
		20	1009	0	100	0	1009	0	0	0
att48	22	5	28724.3	0	227	0	29816	4	1442	44
		10	27039.8	0	114	2	27456	2	758	50
		15	26675.5	0	33	1	27009	2	800	42
		20	26600.3	0	213	1	26692	1	170	4
gr48	23	5	4469.25	0	113	0	4544	1	190	0
		10	4248	0	23	0	4318	1	471	26
		15	4164.75	0	65	1	4231	4	1622	58
		20	4139.6	0	102	2	4157	0	132	15
hk48	18	5	10511.8	0	107	0	10834	3	1804	57
		10	10083.5	0	126	0	10345	6	3634	59
		15	9899	0	204	0	9946	0	223	6
		20	9870	0	96	8	9916	1	235	21
eil51	23	5	437.13	0	157	0	441	1	123	2
		10	423.9	0	101	0	428	2	652	7
		15	414.75	0	71	0	418	2	1124	58
		20	408	0	222	1	408	0	0	0
		25	408	0	2	4	409	0	2	0
berlin52	23	5	6816.5	0	206	0	7052	3	1069	33
		10	6491.83	0	106	0	6609	9	3113	114
		15	6388.75	0	244	0	6444	2	887	15
		20	6322.39	0	221	1	6359	2	580	9
		25	6312	0	3	0	6373	0	2	0
brazil58	27	5	18569.1	0	68	1	20150	14	2135	109
		10	17369	0	88	0	18407	61	8346	227
		15	16877	0	107	0	17582	120	14661	330
		20	16652	0	50	0	17017	53	6665	151
		25	16573	1	287	0	16583	0	88	4
st70	31	5	643.25	1	268	0	665	87	9800	474
		10	603.083	0	101	2	631	156	8656	300
		15	574.25	1	239	2	607	447	20353	838
		20	564	0	75	2	589	435	13435	356
		25	561.5	2	291	0	573	68	3736	71
		30	560	1	302	1	561	1	374	1
		35	560	0	0	0	610	1	0	0
eil76	35	5	560.1	0	121	1	563	4	480	2
		10	546.25	1	376	0	550	4	923	23
		15	539.75	0	109	2	545	45	5528	735
		20	535.75	0	84	7	539	3	621	7
		25	533.167	1	205	0	536	5	1005	57
		30	531.778	1	125	3	533	4	732	34
		35	531	0	20	0	531	0	0	0
gr96	45	5	147533	1	240	0	150721	45	2012	70
		10	143407	1	173	0	147495	226	10814	435
		15	140685	2	393	0	144249	137	10476	475
		20	138559	1	225	0	142035	492	17145	815
		25	137118	1	158	1	139977	560	20101	1163
		30	136510	0	62	2	138216	60	6985	137
		35	136096	2	162	0	137453	222	11954	375
		40	135777	1	67	0	136338	23	3225	38
		45	135563	1	75	0	135563	0	0	0

Table 9: Results for the remaining (symmetric) instances in set A

Name	p'	p	LP	t_L	$\#(\leq p)$	$\#(\geq p)$	OPT	t	$\#(\leq p)$	$\#(\geq p)$
kroA100	45	5	19658	1	143	0	20224	61	2127	55
		10	18699.5	1	226	1	19392	249	11519	297
		15	18078	1	140	3	18755	447	22261	598
		20	17726.9	1	146	0	18383	1481	30894	529
		25	17442.7	1	120	1	17924	113	7391	101
		30	17278.3	2	233	0	17666	5801	64283	785
		35	17199.3	1	95	2	17432	287	14109	111
		40	17168.5	3	177	0	17212	8	1016	17
		45	17153	3	299	0	17153	0	0	0
		50	17153	0	0	0	18618	1	0	0
kroD100	44	5	19322	1	221	0	20284	10221	60366	7862
		10	18353.7	1	170	0	[19106, 19372]	10800*	103038	4750
		15	17662.2	1	293	0	[18330, 18713]	10800*	83788	5871
		20	17177.8	1	125	0	17914	4433	54159	3153
		25	16899.9	1	196	1	17401	3989	57340	2106
		30	16740.5	1	156	1	17015	130	10774	231
		35	16663	3	271	1	16752	4	537	5
		40	16617.3	24	828	0	16650	4	419	12
		45	16585	1	56	4	16625	1	16	3
		50	16585	0	0	0	18474	39	0	0
kroE100	45	5	20337.3	1	288	0	20839	88	4251	189
		10	19096	1	126	0	19595	16	1030	24
		15	18260.5	1	226	0	18958	174	11682	280
		20	17763.5	1	100	0	18424	631	27384	573
		25	17376.3	1	147	0	17958	1556	37284	809
		30	17151.8	4	257	0	17489	82	6066	68
		35	16968.2	3	238	5	17080	6	723	3
		40	16803	1	108	0	16952	39	4674	33
		45	16741	1	0	0	16741	0	0	0
		50	16741	0	0	0	17730	1	0	0
rd100	46	5	7368	1	184	0	7668	526	11318	349
		10	7074.5	1	158	0	7436	5829	54921	3168
		15	6904.67	4	611	0	[7211, 7234]	10800*	98725	1938
		20	6777.25	2	323	1	7028	8312	84520	651
		25	6698	2	225	1	6871	2506	57975	1689
		30	6650	4	460	1	6777	822	21619	309
		35	6625.75	10	670	0	6698	936	33427	488
		40	6616.5	3	306	0	6660	114	7620	87
		45	6613	8	471	8	6617	5	1377	3
		50	6613	0	0	0	6910	1	0	0

*Not solved to optimality within the limit of three hours.

Table 10: Results for the remaining (symmetric) instances in set A (continued)

A.3 Comparison between the PQR model and the x-v model

			PQR model				x-v model			
Name	p'	p	LP	t_L	OPT	t	LP	t_L	OPT	t
ftv33	8	5	1193.11	0	1201	0	1194	0	1201	0
		10	1186	0	1187	0	1185	0	1187	0
		15	1247.8	0	1261	0	1209.63	1	1261	15
ftv35	8	5	1383.75	0	1387	0	1385	0	1387	0
		10	1382.5	0	1383	0	1381.5	0	1383	1
		15	1471.25	0	1480	0	1415.72	15	1480	59
ftv38	8	5	1440.75	0	1444	0	1442	0	1444	0
		10	1439.5	0	1440	0	1438.5	1	1440	0
		15	1520.5	0	1534	0	1465.05	22	1534	3775
p43	16	5	187.833	0	199	7	199	0	199	0
		10	152	0	158	3	157	0	158	0
		15	148	0	148	0	148	0	148	0
		20	148	0	160	0	148	0	160	138
ftv44	9	5	1529.25	0	1543	1	1529.89	0	1543	1
		10	1522	0	1522	0	1522	0	1522	0
		15	1570.5	0	1573	0	1540.83	23	1573	91
		20	1691	0	1691	0	1597.39	49	1691	3711
ftv47	11	5	1661.13	0	1672	1	1664.75	0	1672	1
		10	1652	0	1652	0	1652	0	1652	0
		15	1691.5	0	1703	1	1656	2	1703	453
		20	1815	0	1815	0	1707.81	20	[1774, 1837]	10800*
ry48p	20	5	13165.3	0	13497	1	13320	0	13497	1
		10	12735.8	0	12868	1	12797.8	0	12868	1
		15	12579.3	0	12677	0	12597.5	0	12677	0
		20	12517	0	12517	0	12517	0	12517	0
ft53	8	5	6004.07	0	6022	1	6012.8	0	6022	0
		10	5942	0	5942	0	5936.33	1	5942	5
		15	6027.75	0	6049	1	5960	0	6049	80
		20	6272.3	0	6305	0	6031	1	[6242, 6305]	10800*
		25	6973.5	0	7639	1	6183.19	7	[6203, 8206]	10800*
ftv55	10	5	1460.33	0	1482	1	1462.67	0	1482	1
		10	1435	0	1435	0	1435	0	1435	0
		15	1443.5	0	1445	0	1438.33	3	1445	9
		20	1510	0	1548	1	1453.67	11	[1478, 1601]	10800*
		25	1754	0	1790	0	1515.98	101	[1436, 1802]	10800*
ftv64	9	5	1729.17	0	1732	0	1731.2	1	1732	0
		10	1721	0	1721	0	1721	1	1721	0
		15	1721	0	1721	0	1721	2	1721	0
		20	1760.5	0	1767	0	1726.83	27	[1735, 1843]	10800*
		25	1878	0	1888	0	1767.62	228	[1778, 1958]	10800*
		30	2098.5	0	2140	1	1836.25	407	[1721, 2229]	10800*
ft70	10	5	38055.6	0	38120	3	38113.7	0	38120	1
		10	37978	0	37978	0	37978	0	37978	0
		15	38018.5	0	38033	1	37984.5	3	38033	1078
		20	38275.2	0	38390	7	38013	9	[38040, 39247]	10800*
		25	39027.9	1	39233	9	38109.8	122	[38139, 40027]	10800*
		30	40258.3	2	40539	11	38356.7	640	[38394, 40823]	10800*
		35	42297	0	42908	2	38718.1	1179	[38791, 44946]	10800*

*Not solved to optimality within the limit of three hours.

Table 11: Comparing the performance of the PQR model and x-v model (asymmetric instances continued)

			PQR model				x-v model			
Name	p'	p	LP	t_L	OPT	t	LP	t_L	OPT	t
dantzig42	20	5	579.5	0	604	4	599	0	604	0
		10	545	0	573	3	561.5	1	573	3
		15	534	0	548	1	542.7	1	548	1
		20	532	0	532	0	532	0	532	0
swiss42	20	5	1123.17	0	1155	1	1141.75	0	1155	1
		10	1045	0	1084	2	1065.5	0	1084	1
		15	1021.83	0	1034	1	1026.5	0	1034	0
		20	1009	0	1009	0	1009	0	1009	0
att48	22	5	28724.3	0	29816	4	29195.5	0	29816	1
		10	27039.8	0	27456	2	27256.5	0	27456	1
		15	26675.5	0	27009	2	26794.3	0	27009	1
		20	26600.3	0	26692	1	26610.5	0	26692	0
gr48	23	5	4469.25	0	4544	1	4526.06	0	4544	0
		10	4248	0	4318	1	4260.5	0	4318	1
		15	4164.75	0	4231	4	4183.17	1	4231	2
		20	4139.6	0	4157	0	4140.5	0	4157	1
hk48	18	5	10511.8	0	10834	3	10769.6	1	10834	0
		10	10083.5	0	10345	6	10219.7	0	10345	1
		15	9899	0	9946	0	9918.8	1	9946	0
		20	9870	0	9916	1	9870	0	9916	2
eil51	23	5	437.13	0	441	1	438.571	0	441	1
		10	423.9	0	428	2	425.5	0	428	3
		15	414.75	0	418	2	414.8	1	418	2
		20	408	0	408	0	408	0	408	0
		25	408	0	409	0	408	0	409	0
berlin52	23	5	6816.5	0	7052	3	6913.43	0	7052	1
		10	6491.83	0	6609	9	6564	2	6609	2
		15	6388.75	0	6444	2	6400.5	0	6444	2
		20	6322.39	0	6359	2	6337	1	6359	1
		25	6312	0	6373	0	6312	0	6373	46
brazil58	27	5	18569.1	0	20150	14	19103.3	0	20150	1
		10	17369	0	18407	61	17776.5	1	18407	2
		15	16877	0	17582	120	17191.5	1	17582	2
		20	16652	0	17017	53	16748.5	1	17017	2
		25	16573	1	16583	0	16582	0	16583	0
st70	31	5	643.25	1	665	87	662.1	2	665	2
		10	603.083	0	631	156	621.4	18	631	26
		15	574.25	1	607	447	594.154	19	607	479
		20	564	0	589	435	576.3	5	589	29
		25	561.5	2	573	68	563.5	3	573	2
		30	560	1	561	1	560.167	1	561	0
		35	560	0	610	1	560	0	[560, 728]	10800*

*Not solved to optimality within the limit of three hours.

Table 12: Comparing the performance of the PQR model and x-v model (symmetric instances continued)

			PQR model				x-v model			
Name	p'	p	LP	t_L	OPT	t	LP	t_L	OPT	t
eil76	35	5	560.1	0	563	4	560.889	0	563	6
		10	546.25	1	550	4	548.382	2	550	5
		15	539.75	0	545	45	543	2	545	4
		20	535.75	0	539	3	538	10	539	6
		25	533.167	1	536	5	534	5	536	4
		30	531.778	1	533	4	532	7	533	7
		35	531	0	531	0	531	1	531	0
pr76	36	5	91255.6	0	[96104, 97764]	10800*	93076.5	1	[96803, 97450]	10800*
		10	85634	2	[90747, 91883]	10800*	88006	5	[90206, 92329]	10800*
		15	81211.5	0	86380	1066	84413.3	14	86380	51
		20	79365.2	0	82311	99	81247	13	82311	5
		25	78332	0	82040	1757	79681.2	18	[81148, 82253]	10800*
		30	77731.8	0	[80612, 81961]	10800*	78302.5	10	[79655, 82651]	10800*
		35	77207.3	1	77973	1	77260.5	2	77973	1
gr96	45	5	147533	1	150721	45	149292	2	150721	75
		10	143407	1	147495	226	145200	17	[146647, 147635]	10800*
		15	140685	2	144249	137	142426	51	[143749, 146342]	10800*
		20	138559	1	142035	492	140240	80	142035	9113
		25	137118	1	139977	560	138444	67	139977	64
		30	136510	0	138216	60	137192	47	138216	40
		35	136096	2	137453	222	136350	10	137453	37
		40	135777	1	136338	23	135861	5	136338	11
		45	135563	1	135563	0	135563	1	135563	0
rat99	45	5	1228.5	1	1237	8	1231.5	1	1237	3
		10	1196.5	1	1212	68	1207.6	14	1212	6
		15	1178.17	1	1195	39	1190.88	222	1195	48
		20	1164.17	2	1184	172	1175	38	1184	2837
		25	1154.38	1	1170	177	1161	8	1170	683
		30	1148	3	1159	41	1152.54	33	1159	48
		35	1144.44	2	1153	50	1147.86	23	[1151, 1159]	10800*
		40	1142.17	2	1145	15	1144	20	1145	10
		45	1142	0	1142	0	1142	2	1142	0
kroA100	45	5	19658	1	20224	61	20046.3	2	20224	148
		10	18699.5	1	19392	249	19236.3	19	19392	69
		15	18078	1	18755	447	18572.6	17	18755	94
		20	17726.9	1	18383	1481	18043.8	19	18383	75
		25	17442.7	1	17924	113	17717.8	25	17924	29
		30	17278.3	2	17666	5801	17451.6	74	17666	30
		35	17199.3	1	17432	287	17253.2	13	17432	27
		40	17168.5	3	17212	8	17179.5	7	17212	5
		45	17153	3	17153	0	17153	1	17153	0
		50	17153	0	18618	1	17153	2	[17153, 22460]	10800*
kroD100	44	5	19322	1	20284	10221	19915.5	5	20284	22
		10	18353.7	1	[19106, 19372]	10800*	18934.2	100	19275	150
		15	17662.2	1	[18330, 18713]	10800*	18234.2	104	[18489, 19128]	10800*
		20	17177.8	1	17914	4433	17670.4	77	17914	89
		25	16899.9	1	17401	3989	17259.2	30	17401	17
		30	16740.5	1	17015	130	16935.5	33	17015	11
		35	16663	3	16752	4	16718.5	8	16752	2
		40	16617.3	24	16650	4	16634	8	16650	5
		45	16585	1	16625	1	16585	2	16625	16
		50	16585	0	18474	39	16585	1	[16585, 20732]	10800*
kroE100	45	5	20337.3	1	20839	88	20427	1	20839	128
		10	19096	1	19595	16	19430	11	19595	21
		15	18260.5	1	18958	174	18644	23	18958	50
		20	17763.5	1	18424	631	18043.5	45	18424	1887
		25	17376.3	1	17958	1556	17629	48	17958	3899
		30	17151.8	4	17489	82	17292.9	11	17489	108
		35	16968.2	3	17080	6	17014.8	13	17080	7
		40	16803	1	16952	39	16827.1	7	16952	17
		45	16741	1	16741	0	16741	1	16741	0
		50	16741	0	17730	1	16741	1	[17062, 18524]	10800*
rd100	46	5	7368	1	7668	526	7508.33	2	[7550, 8143]	10800*
		10	7074.5	1	7436	5829	7222.75	103	[7361, 7716]	10800*
		15	6904.67	4	[7211, 7234]	10800*	7024.45	43	[7146, 7264]	10800*
		20	6777.25	2	7028	8312	6860	23	[6949, 7048]	10800*
		25	6698	2	6871	2506	6762.5	20	6871	64
		30	6650	4	6777	822	6703.75	112	6777	39
		35	6625.75	10	6698	936	6658.5	42	6698	27
		40	6616.5	3	6660	114	6629.75	28	6660	15
		45	6613	8	6617	5	6614	10	6617	1
		50	6613	0	6910	1	6613	1	[6613, 8100]	10800*

*Not solved to optimality within the limit of three hours.

Table 13: Comparing the performance of the PQR model and x-v model (symmetric instances continued)

A.4 Additional results for the symmetric case in which two-node circuits are not allowed

Name	p'	p	LP	t_L	$\#(\leq p)$	$\#(\geq p)$	OPT	t	$\#(\leq p)$	$\#(\geq p)$
dantzig42	8	3	641	0	79	0	648	1	226	5
		10	651.5	0	79	17	654	0	62	18
swiss42	7	4	1214.5	0	197	0	1232	1	211	7
		6	1214.5	0	165	21	1231	1	176	25
		8	1218.81	0	131	43	1231	1	87	65
		10	1225.75	0	246	75	1238	1	95	105
		14	1270	0	22	104	1292	1	10	285
att48	5	4	31703.7	0	231	4	31903.3	1	168	3
		6	31671.4	0	134	55	31836.1	1	56	61
		9	31756.7	0	96	83	32195.5	2	99	170
		12	32083.2	1	195	188	32742.9	4	81	416
		16	33217	1	111	289	[35667.8, 37874.3]	10800*	989	29397
gr48	6	4	4770	0	125	1	4841	3	417	120
		6	4769.75	0	144	30	4805	2	308	243
		9	4807	0	37	224	4926	13	364	1094
		12	4886.21	1	73	395	5011	8	145	1094
		16	5119.74	4	234	711	5445	208	407	6848
hk48	6	4	11197.5	0	174	2	11271	2	400	52
		6	11197	0	129	6	11197	0	0	0
		9	11250.3	0	180	139	11292	2	151	270
		12	11367.7	1	191	238	11450	4	315	383
		16	11742.7	1	92	378	12215	118	558	4030
eil51	3	5	419.58	1	131	155	422.323	4	286	354
		7	421.306	1	94	195	424.356	5	153	540
		10	424.737	1	128	312	432.489	10	282	1114
		12	428.425	2	141	441	436.587	14	440	1518
		17	445.355	3	379	423	473.977	1701	978	9965
berlin52	7	5	7167.14	0	165	20	7182.23	2	282	46
		7	7166.87	0	141	21	7167.2	1	134	14
		10	7182.32	1	189	176	7206.7	2	116	70
		13	7263.34	1	247	225	7298.63	2	109	209
		17	7559.85	0	17	167	7800.77	36	179	2031
brazil58	12	5	21001	0	196	1	21744	12	1314	191
		8	20904	1	421	8	21289	6	896	98
		11	20902.4	3	581	31	21080	5	892	16
		14	21023.2	2	689	97	21221	2	70	90
		19	21631.7	1	266	205	22635	71	556	2925
st70	12	7	631.417	1	227	4	638.221	5	437	18
		10	628.559	6	1077	5	632.54	4	430	23
		14	628.708	13	1109	187	630.902	3	102	141
		17	630.017	10	676	367	636.194	9	283	403
		23	648.67	7	560	266	694.495	3739	2440	9607

*Not solved to optimality within the limit of three hours.

Table 14: Results for the symmetric case in which two-node circuits are not allowed (continued)