Extended Formulations and Branch-and-Cut Algorithms for the Black-and-White Traveling Salesman Problem

Luis Gouveia[1], Markus Leitner[2] and Mario Ruthmair[2]

[1]Universidade de Lisboa, Faculdade de Ciências, Departamento de Estatística e Investigação Operacional, Lisbon, Portugal.
legouveia@fc.ul.pt

[2]Department of Statistics and Operations Research, Faculty of Business, Economics and Statistics, University of Vienna, Vienna, Austria.
markus.leitner@univie.ac.at, mario.ruthmair@univie.ac.at

March 8, 2017

Abstract

In this paper we study integer linear programming models and develop branch-and-cut algorithms to solve the Black-and-White Traveling Salesman Problem (BWTSP) [5] which is a variant of the well known Traveling Salesman Problem (TSP). Two strategies to model the BWTSP have been used in the literature. The problem is either modeled on the original graph as TSP using a single set of binary edge variables and with additional non-trivial hop and distance constraints between every pair of black nodes (see Ghiani et al. [7]) or as a sequence of constrained paths composed of white nodes connecting pairs of black nodes (see Muter [22]). In this paper, we study and develop an intermediate approach based on the observation that it is sufficient to guarantee the required distance (and hop) limit of the path from a given black node to the next black node without explicitly stating which one it is. Thus, instead of stating the two constraints (after adding appropriately defined variables) for each pair of black nodes, they are stated for each black node only (that represents the source of each path). Based on this idea we develop several variants of position- and distance-dependent reformulations together with corresponding layered graph representations. Branch-and-cut algorithms are developed for all proposed formulations and empirically compared by an extensive computational study. The obtained results allow us to provide insights into individual advantages and disadvantages of the different models.

Keywords: Traveling Salesman Problem, Hop Constraint, Distance Constraint, Integer Linear Programming, Layered Graph, Branch-and-Cut

1 Introduction

The Traveling Salesman Problem (TSP) arguably is the most well studied combinatorial optimization problem. Many real-world applications have been described both for the classical TSP (see, e.g., Applegate et al. [2], Lawler et al. [19]) as well as of many TSP variants (see, e.g., Gutin and Punen [17]). One such variant is the Black-and-White TSP (BWTSP) studied in this article whose name has been coined in Bourgeois et al. [5] where the BWTSP is introduced as a variant of related problems that have been considered and studied even before.

Formally, the BWTSP is defined on an undirected graph $G = (V, E)$ where the set of nodes $V = B \cup W$ is partitioned into two sets of nodes, the set of black nodes B and the set of white nodes W. Each edge $e \in E$ is associated with a cost $c_e \in \mathbb{N}$ and a distance $d_e \in \mathbb{N}$. The objective is to find a TSP tour with minimum total length in which the path between every two consecutive black nodes contains at most Q white nodes and has a distance of at most L. An example instance of the BWTSP together with a solution is given in Figure 1.

Applications of the BWTSP are known from short-haul airline operations and from the design of survivable fiber networks using SONET technology (see Ghiani et al. [7]). In the context of short-haul

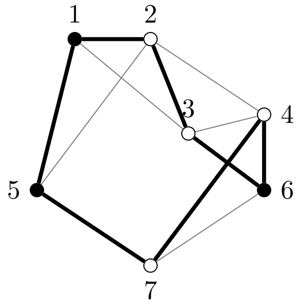

Figure 1: An example instance with $B = \{1, 5, 6\}$, $W = \{2, 3, 4, 7\}$ and a feasible solution (indicated by bold edges) to that instance for $Q = 2$ and $L = 3$. The indicated solution is optimal when, e.g., assuming that $c_e = d_e = 1$ for all edges e contained in the solution, while $c_e = d_e = 2$ for all remaining edges.

airline operations, black nodes model maintenance stations that must be visited by an aircraft after at most $Q + 1$ legs or after a total operational distance of L. In telecommunication applications, black vertices represent ring offices whose distance between each other is limited (both in terms of hops and length) while white vertices model standard hubs.

Several slightly different problems have been considered as the BWTSP in the literature. These include a variant defined on a directed graph (V, A) with asymmetric arc costs and in particular a variant in which the distance limit is imposed on edge (arc) costs which are then interpreted as travel times [5, 7, 22]. The latter problem variant is a special case of the BWTSP variant studied in this article when $c_e = d_e$, $\forall e \in E$.

Previous Work Integer Linear Programming (ILP) models for the BWTSP have been proposed, developed and tested by Ghiani et al. [7] and Muter [22]. Ghiani et al. [7] propose a formulation involving only 0-1 edge variables which is augmented with several non-trivial classes of inequalities to guarantee the extra requirements as well as different sets of non-trivial valid inequalities that strengthen the linear programming relaxation (LP) bound of the resulting model. They develop a branch-and-cut method and present results showing that the proposed method can solve instances with up to 100 nodes. In their work they consider $d_e = c_e$, $\forall e \in E$, and the distance constraints therefore correspond to a knapsack-cost constraint for each path between a black node and the next one in the tour. As often observed for natural variable space models, their branch-and-cut algorithm works well for loosely constrained instances. When considering tight instances, however, too many violated inequalities have to be added leading to long solution times. Muter [22] proposes a new formulation using three-index edge variables that, for each edge $e \in E$, also identify the black nodes that are immediately before and after. Dantzig-Wolfe decomposition is used to obtain a formulation that considers one variable for each hop- and distance-feasible path between each pair of black nodes. Compared to the model by Ghiani et al. [7], the number of constraints needed in the model can be substantially reduced but due to the large number of variables, a column generation approach embedded in a branch-and-bound algorithm is proposed and tested. Results show that the proposed algorithm solves randomly generated instances with up to 80 nodes. Here, instances with loose constraints lead to the generation of a large number of columns and therefore to extensive solution runtimes. The name BWTSP has been coined by Bourgeois et al. [5] who develop several constructive heuristics and tailored 2-opt improvement procedures. Results of computational experiments on instances with at most 200 nodes are given. Bourgeois et al. [5] also observe that several problems considered in earlier articles (see [21, 27, 29]) could be modeled as the BWTSP. One such problem is the asymmetric traveling salesman problem with replenishment arcs for which a simulated annealing algorithm and a Lagrangian relaxation based method have been developed by Mak and Boland [21]. In this problem which has applications in airline planning, nodes are associated with resource values $w_i \geq 0$ and a replenishment arc has to be visited after accumulating at most W resource units by traversing nodes. A variant of the BWTSP without hop-constraints and for which in general $c_{ij} \neq d_{ij}$ holds, is obtained by setting $d_{ij} = w_j$ for all arcs (i, j).

Several of the models proposed in this paper are based on position-dependent reformulations. In the following, we therefore provide a short review of this domain. The earliest reference to a position-dependent formulation has probably been made by interpreting the formulation by Picard and Queyranne [24] for the time-dependent TSP in this context, see, e.g., Abeledo et al. [1], Godinho et al. [8], Gouveia and Voß [12]. More explicit allusions to layered graphs have been made in a subsequent set of articles studying formulations for different network design problems with hop and/or distance constraints, see, e.g., [4, 9, 13, 10]. In the latter works each commodity is associated to one layered graph (or equivalently, a position-dependent flow system). Recently, the designation "layered graph" is typically associated to formulations in which the whole problem considered is modeled in such a graph. Perhaps, the first such reference is the work by Gouveia et al. [14] in which the authors show how to model hop- or distance-constrained spanning trees in a single layered graph. A similar approach has been used in several subsequent articles, see, e.g., [11, 15, 16, 20, 25, 26]. With the exceptions given in Gouveia et al. [16] and Gouveia and Ruthmair [11], these works use layered graphs with two dimensions that are associated to nodes of the graph and the considered resource value (e.g., distance, delay, or time). Gouveia and Ruthmair [11] consider a three-dimensional layered graph for pickup and delivery problems in which each layered graph node simultaneously indicates the original node, its position in the tour and the load of the vehicle. Gouveia et al. [16] study a problem that can be modeled as a Steiner tree with hop constraints that includes a hop-constrained spanning spanning tree. The authors first show that the problem can be modeled using two layered graphs, one representing the spanning tree and a second one representing the Steiner tree. In addition, a less straightforward, three-dimensional, generalization of layered graphs is proposed which models the complete problem at once. Here, the main idea is to include a third dimension that represents the distance between the centers of the two trees. These studies have shown that layered graphs and variants of them can be used in different ways to model problems with resource constraints. As we will demonstrate in this paper, the number of relevant modeling possibilities increases notably when considering problems with two resource constraints.

Scientific Contribution and Outline As noted above, two strategies to model this problem have been used in the literature. The problem is either modeled on the original graph as traveling salesman problem (TSP) using a single set of binary edge variables and with additional non-trivial hop and distance constraints between every pair of black nodes (see Ghiani et al. [7]) or as a sequence of constrained paths composed of white nodes connecting pairs of black nodes (Muter [22]). In this paper, we use an intermediate approach based on the observation that it is sufficient to guarantee the required distance (and hop) of the path from a given black node to the next black node without explicitly stating which one it is. Thus, instead of stating the two constraints (after adding appropriate extra variables) for each pair of black nodes, in the new model they are stated for each black node only (that represents the source tof each path). A generic formulation based on this idea is introduced in the remainder of this section. Section 2 discusses ILP formulations based on arc variables that are disaggregated by black nodes. They also form the basis for the position-dependent reformulations that are introduced in Section 3 together with corresponding layered graph representations. In Section 4 we discuss alternative, distance-dependent, reformulations as well as formulations arising from position- and distance-dependent reformulations which are also explained from a layered graph perspective. The existence of these options arises from the fact that the BWTSP has two independent resource constraints. In this work we discuss several variants of layered graph models that are positioned between two extreme representations. In one extreme case, we model explicitly the distance and hop constraints in the original graph and thus, there is no layered graph representation. In the other one we consider a multidimensional layered graph in which a node in the layered graph represents an original node in one particular position and in one particular distance in the path starting from the last black node. Examples of "intermediate" layered graph representations that are discussed in the following include the consideration of (i) a layered graph representing one resource (hop or distance) and explicit constraints (on this layered graph) for the other resource (we propose two such representations since there are two resources) and (ii) two layered graphs, each modeling implicitly one of the resources and which are linked through additional constraints.

The options considered are generic in the sense that similar ideas can be applied for other problems with two (or multiple) independent resource constraints. Thus, the relevance of this study might go significantly beyond the study of formulations for the BWTSP. Section 7 details the results of our extensive computational study and provides insights into individual advantages and disadvantages of the different models. Finally, our conclusions are drawn in Section 8.

The Generic model The abstract generic formulation (1)–(6) will serve as a common basis for the models introduced in the following sections. It uses edge decision variables $x_e \in \{0,1\}$, $\forall e \in E$, indicating whether an edge is included in the tour.

$$\min \quad \sum_{e \in E} c_e x_e \tag{1}$$

$$\text{s.t.} \quad x(\delta(i)) = 2 \qquad \forall i \in V \tag{2}$$

$$x(E(S)) \geq 2 \qquad \forall S \subseteq V \setminus \{1\}, S \neq \emptyset \tag{3}$$

$$\{e \in E : x_e = 1\} \quad \text{satisfies the hop constraint for the path after each } k \in B \tag{4}$$

$$\{e \in E : x_e = 1\} \quad \text{satisfies the distance constraint for the path after each } k \in B \tag{5}$$

$$\mathbf{x} \in \{0,1\}^{|E|} \tag{6}$$

Several ways for modeling Hamiltonian cycles are known. In this work, we use degree constraints (2) and the exponentially sized set of cut constraints (3). Notation $x(\delta(i))$ and $x(E(S))$ used in these two sets of constraints is introduced in the next paragraph.

As noted before, in the work by Ghiani et al. [7], the additional constraints (4) and (5) are modeled with non-trivial inequalities in the space of $\mathbf{x}$ variables. On the contrary, Muter [22] used additional three-index variables indicating whether an edge is used on a path between two black nodes. Using these variables, constraints (4) and (5) are easy to model. As pointed out before, Muter [22] also considers an alternative so-called path formulation with one variable for each hop- and distance-feasible path that implicitly satisfies constraints (4) and (5).

Notation The formulations introduced in the following will make use of arc set $A = \{(i,j) \mid \{i,j\} \in E\}$ obtained from bi-directing the edge set E. For a subset of nodes $S \subset V$, we will also use notation $\delta(S) = \{\{i,j\} \in E \mid i \in S, j \notin S\}$, $\delta^+(S) = \{(i,j) \in A \mid i \in S, j \notin S\}$, and $\delta^-(S) = \{(i,j) \in A, i \notin S, j \in S\}$. Analogous notation will be used for graphs different from G (and arc sets different from A) without explicitly mentioning the considered graph whenever it is clear from the context. For a set of variables $\mathbf{z}$ defined on set W and a subset $W' \subseteq W$, we will use notation $z(W') = \sum_{u \in W'} z_u$.

2 Path Segment Model

In this section, we introduce a formulation that is based on the view described in the introduction. That is, we identify the path from a given black node $k \in B$ to the next black node without explicitly stating which one it is. In the following, such a path will be denoted as the "path segment associated to node k". Let variables $x^k_{ij} \in \{0,1\}$, $\forall k \in B$, $\forall (i,j) \in A$, indicate whether the arc (i,j) is contained in the path segment associated to black node k. These variables allow us to model the path segment associated to each black node by constraints (7)–(10) as well as to formulate the corresponding hop and distance constraints, (4) and (5), respectively.

$$x^k(\delta^+(k)) = 1 \qquad \forall k \in B \tag{7}$$

$$\sum_{j \in B \setminus \{k\}} x^j(\delta^-(k)) = 1 \qquad \forall k \in B \tag{8}$$

$$x^k(\delta^-(i)) = x^k(\delta^+(i)) \qquad \forall k \in B, \forall i \in W \tag{9}$$

$$\sum_{k \in B} x^k_{ij} + x^k_{ji} = x_e \qquad \forall e = \{i,j\} \in E \tag{10}$$

Equations (7) and (8) are degree constraints ensuring that exactly one ingoing and one outgoing arc is selected for each black node. Thus, each black node is the initial node of a segment associated to itself and is also the end node of a segment associated to another black node. Constraints (9) are typical flow balance constraints for the white nodes. Constraints (10) are the linking constraints between the two sets of variables. A valid formulation for the BWTSP is obtained by additionally including the hop constraints (11) and the distance constraints (12).

$$\sum_{e=\{i,j\}\in E} (x_{ij}^k + x_{ji}^k) \leq Q + 1 \qquad \forall k \in B \tag{11}$$

$$\sum_{e=\{i,j\}\in E} d_e(x_{ij}^k + x_{ji}^k) \leq L \qquad \forall k \in B \tag{12}$$

We denote by PS the resulting formulation which is defined by replacing the abstract constraints (4) and (5) by (7)–(12), and the definitional constraints for variables x_{ij}^k.

Using the new variables and assuming that $|B| \geq 3$, we can also consider valid inequalities (13) that relate the path segments associated to different black nodes.

$$x^k(\delta^-(j)) + x^j(\delta^-(k)) \leq 1 \qquad \forall k \neq j \in B \tag{13}$$

These two-cycle elimination constraints (13) consider the sequence of black nodes in a solution. They ensure that if black node j is the black node immediately after black node k (that is j is part of the path segment associated to k) then black k cannot be immediately after black node j. Our computational results show that these inequalities improve the LP bounds of PS for some of the instances tested and we will denote the variant of PS including inequalities (13) as PS$^+$.

3 Position-Dependent Reformulations

As mentioned in Section 1, so-called time-dependent formulations provide strong LP bounds. In general and also in the formulations introduced in this section such reformulations are characterized by adequately attaching and exploiting information about the position (time) of an arc in the tour or in the path segment associated to a black node. At the price of a (significantly) larger number of variables, they ensure that the hop constraints are implicitly satisfied. In particular, if the hop limit is not too large, algorithmic approaches based on such formulations can produce results comparable with state-of-the-art results, or even better, cf. Section 1. To be more specific, in the following we will speak of "position-dependent reformulations" instead of using the more general notion of "time dependency".

In this section we introduce two such approaches. The first is based on adding information about the position of an arc with respect to its position after the previous black node. In this first formulation, the information about the black node initializing each particular path segment is, however, not provided in an explicit way. Hence, the distance constraints need to be included in a way similar to the one by Ghiani et al. [7]. The second approach can be viewed as a position-dependent reformulation of the path segment model introduced in Section 2. Information about the previous black node and the position relative to it is added to each arc. As will be detailed later, this additional information allows to easily model the distance constraints. These two approaches are discussed in Sections 3.1 and 3.2, respectively. Table 1, given in Section 5, contains an overview over all formulations proposed in this section.

Also, time- or position- (or other resource-) dependent formulations can be interpreted in appropriately defined layered graphs. Such interpretations often help to better understand the structure of solutions in the corresponding extended variable space and consequently to derive valid inequalities in this extended space. Thus, we will also use the concept of layered graphs when introducing and explaining the two position-indexed formulations. By replicating original nodes an appropriate number of times, both layered graphs that will be formally defined in the following subsections allow to implicitly ensure the hop limit between successive black nodes. Details from each approach differ, however, significantly. The first position-dependent formulation (see Section 3.1) is based on a single non-acyclic layered graph while the layered graph associated to the second model (see Section 3.2) can be seen as a combination of several acyclic layered subgraphs, one for each black node.

3.1 A pure position-dependent formulation

The formulation introduced in this section is based on classical position-dependent variables $X_{ij}^p \in \{0,1\}$, $\forall (i,j) \in A$, $\forall p \in \{1,2,\ldots,Q+1\}$. For each arc $(i,j) \in A$ variable X_{ij}^p is equal to one if and only if that arc is at position p in the path starting from the previous black node. As p indicates the position in a path segment and not in the overall tour, several arcs may be at the same position p.

$$X^1(\delta^+(k)) = 1 \qquad \forall k \in B \tag{14}$$

$$\sum_{p=1}^{Q+1} X^p(\delta^-(k)) = 1 \qquad \forall k \in B \tag{15}$$

$$X^p(\delta^-(i)) = X^{p+1}(\delta^+(i)) \qquad \forall i \in W, \forall p \in \{1,2,\ldots,Q\} \tag{16}$$

$$\sum_{p=1}^{Q+1} \left(X^p_{ij} + X^p_{ji}\right) = x_e \qquad \forall e = \{i,j\} \in E \tag{17}$$

First observe that the hop constraints (4) are implicitly guaranteed by the considered range of p. Thus, they become redundant and can be removed. Constraints (14)–(16) arc flow balance constraints in the space of the position-indexed variables. They state (i) that one arc leaves each black node in position one, (ii) that one arc at (at a feasible position) enters each black node, and (iii) that an arc may only leave a white node i at position $p+1$ if and only if another arc enters the same white node at position p. Constraints (17) finally link the position-indexed arc variables to the previously defined undirected edge variables.

Recall that the position-indexed variables included in this model do not specify the black node initializing each path-segment. Thus, there is no trivial way to model the distance constraints. Similar to Ghiani et al. [7] we therefore include path elimination constraints (18) to eliminate all paths that violate the distance constraints. Thereby, $\mathcal{P}_{\text{inf}} \subset 2^E$ denotes the set of all paths between two black nodes in G that do not contain any other black nodes and that violate the distance constraint, i.e. $\sum_{e\in P} d_e > L$, $\forall P \in \mathcal{P}_{\text{inf}}$.

$$\sum_{e\in P} x_e \leq |P| - 1 \qquad \forall P \in \mathcal{P}_{\text{inf}} \tag{18}$$

Similar sets of path elimination constraints as well as different separation techniques have been considered in the literature, see, e.g., Ascheuer et al. [3].

We denote by PD the resulting position-dependent formulation which is formally obtained from the generic formulation by including constraints (14)–(18) and definitional constraints for the position dependent variables X^p_{ij} instead of the generic hop- and distance constraints (4) and (5), respectively.

One polynomial set of inequalities that typically strengthen the LP bounds of time-dependent formulations are the so-called two-cycle inequalities, see, e.g., Abeledo et al. [1], Godinho et al. [8]. For the case of formulation PD, they are defined by constraints (19) and (20). These inequalities simply state that if an arc (i,j) is in position $p+1$ in a given segment, then the previous arc in position p, converging into node i, cannot be diverging from node j. We will denote by PD$^+$ the variant of formulation PD augmented with the 2-cycle inequalities (19) and (20).

$$X^p_{ij} \leq \sum_{h\neq i} X^{p+1}_{jh} \qquad \forall p \in \{1,2,\ldots,Q\}, \forall e = \{i,j\} \in E, j \in W \tag{19}$$

$$X^p_{ij} \leq \sum_{h\neq i} X^1_{jh} \qquad \forall p \in \{1,2,\ldots,Q+1\}, \forall e = \{i,j\} \in E, j \in B \tag{20}$$

As noted in the introduction to this section, position-dependent formulations can be viewed as formulations in a special layered graph. Next, we define the layered digraph $G_{\text{Q}} = (V_{\text{Q}}, A_{\text{Q}})$ associated to formulation PD. The node set $V_{\text{Q}} = \{i_0 \mid i \in B\} \cup \{i_p \mid i \in W, 1 \leq p \leq Q\}$ consists of black nodes at layer zero (since the hop count of each path segment initialized by some black node starts with zero) and copies of white nodes at all feasible layers p, $1 \leq p \leq Q$. A node i_p included in a solution hence indicates that the unique path to node i from its preceding black node has length p. Thus, in the space of layered digraph G_{Q}, any feasible solution to the BWTSP must visit all black nodes (at layer zero) and exactly one copy of each white node (at some layer p, $1 \leq p \leq Q$). The arc set A_{Q} consists of (i) arcs (i_p, j_{p+1})

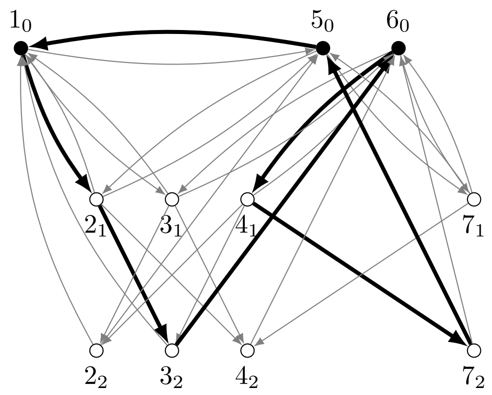

Figure 2: Layered digraph $G_Q = (V_Q, A_Q)$ corresponding to the instance given in Figure 1. Bold arcs represent the solution shown in Figure 1.

between two white nodes i and j, (ii) arcs (i_0, j_0) between two black nodes i and j, (iii) arcs (i_0, j_1) from a black node i to a white node j, and (iv) arcs (i_p, j_0) from a white node i to a black node j. Formally, we have $A_Q = \{(i_p, j_{p+1}) \mid i_p, j_{p+1} \in V_Q, (i,j) \in A\} \cup \{(i_0, j_0) \mid i,j \in B, (i,j) \in A\} \cup \{(i_p, j_0) \mid i \in W, j \in B, i_p \in V_Q, (i,j) \in A\}$. We note that the interpretation of formulation PD on graph G_Q is based on associating each of the above defined variables X_{ij}^p either with arc (i_{p-1}, j_p) (if $j \in W$) or arc (i_{p-1}, j_0) (if $j \in B$). Figure 2 shows the solution depicted in Figure 1 in the context of digraph G_Q.

There are several valid inequalities that significantly strengthen the LP bounds of the time-dependent model and that are more easily described in terms of layered digraph G_Q. Following previous studies, see e.g., [11, 14, 15, 16], we can consider more general layered graph specific cut inequalities that do not correspond to the $\mathbf{x}$-cuts rewritten with the X_{ij}^p variables via equations (17). Different classes of such cut inequalities can be obtained from the observation that every node must be reachable from the source node 1_0 which is without loss of generality assumed to be the start of the tour. Here, we consider the particular set of cut inequalities (21) ensuring that at least one arc must cross each cut separating the set of black nodes from all copies of a given white node.

$$X(\delta^-(S)) \geq 1 \qquad \forall S \subseteq V_Q \setminus \{j_0 : j \in B\}, \exists i \in W \mid \{i_p, p \in \{1,2,\ldots,Q\}\} \subseteq S \tag{21}$$

The variant of PD$^+$ augmented with cut constraints (21) will be denoted by PD^{++}. Several other similar families of layered graph cuts could be defined. In fact, one larger family of valid inequalities is obtained by allowing copies of black nodes k_0, $k \neq 1$, to be on the target side of the cut. Using only (21) has, however, turned out to provide the best trade-off between additional strength and necessary separation time in preliminary computational experiments. We therefore refrain from giving further details of such additional families of inequalities. We also note that similar observations also hold for the formulations that will be introduced in the following two subsections for which we will not explicitly mention these additional families of valid inequalities.

3.2 A position-dependent reformulation of the path segment model

We can incorporate ideas of the previous position-dependent model in the context of the path segment model introduced in Section 2. Consider the variables $Y_{ij}^{k,p} \in \{0,1\}$, $\forall k \in B$, $\forall p \in \{1,\ldots,Q+1\}$, $\forall (i,j) \in A$, that disaggregate both the variables used in the path segment model as well as the position-dependent variables introduced in the previous subsection. Variable $Y_{ij}^{k,p}$ will indicate whether arc (i,j) is at position p in the path starting at black node k. To obtain the new formulation we first consider constraints (22)–(25) which generalize constraints (7)–(10) but also constraints (14)–(17) from the previous position-dependent formulation.

$$Y^{k,1}(\delta^+(k)) = 1 \qquad \forall k \in B \tag{22}$$

$$\sum_{j\in B\setminus\{k\}}\sum_{p=1}^{Q+1} Y^{j,p}(\delta^-(k)) = 1 \qquad \forall k \in B \qquad (23)$$

$$Y^{k,p}(\delta^-(i)) = Y^{k,p+1}(\delta^+(i)) \qquad \forall k \in B,\ \forall i \in W, \forall p \in \{1,\dots,Q\} \qquad (24)$$

$$\sum_{k\in B}\sum_{p=1}^{Q+1}\left(Y_{ij}^{k,p} + Y_{ji}^{k,p}\right) = x_e \qquad \forall e = \{i,j\} \in E \qquad (25)$$

As in the model of the previous subsection, the hop constraints (4) are implicitly satisfied due to the definition of the extended variables. Flow conservation constraints (22)–(24) state (i) that one arc leaves each black node at position one, (ii) that exactly one arc associated to the segment of a different black node has to converge into a black node, and (iii) if an arc contained in the segment associated to black node k converges into a white node into position p, then another arc in a path from the same black node diverges from the same node in position $p+1$. Constraints (25) link the new variables to the edge design variables.

As the considered variables explicitly specify the arcs associated to each path segment, the distance constraints can also be easily written as follows:

$$\sum_{p=1}^{Q+1}\sum_{e=\{i,j\}\in E} d_e(Y_{ij}^{k,p} + Y_{ji}^{k,p}) \le L \qquad \forall k \in B \qquad (26)$$

We will denote by PDPS the position-dependent reformulation of model PS introduced in this section. It is formally defined by replacing the generic hop- and distance constraints (4) and (5) by (22)–(26). Similar to the model proposed in the previous subsection, we can strengthen PDPS by considering the 2-cycle inequalities (27) and (28) for white and black nodes, respectively, yielding model PDPS$^+$.

$$Y_{ij}^{k,p} \le \sum_{h\neq i} Y_{jh}^{k,p+1} \qquad \forall k \in B, \forall p \in \{1,2,\dots,Q\}, \forall e = \{i,j\} \in E, j \in W \qquad (27)$$

$$Y_{ij}^{k,p} \le \sum_{h\neq i} Y_{jh}^{j,1} \qquad \forall k \in B, \forall p \in \{1,2,\dots,Q+1\}, \forall e = \{i,j\} \in E, j \in B\setminus\{k\} \qquad (28)$$

As discussed for the formulation of the previous subsection (and for similar reasons), the formulation in this section can also be viewed in a layered graph. The layered graph described in this section results from combining layered subgraphs, one for each black node with the idea that the whole path associated to a black node is contained in one layered graph. Consider the layered graph $G_{\text{QB}} = (V_{\text{QB}}, A_{\text{QB}})$ which implicitly encodes information about the black node initiating a path segment and the position of all white nodes contained on that path. Its node set is defined as $V_{\text{QB}} = \{i_0^i \mid i \in B\} \cup \{i_p^k \mid i \in W, p \in \{1,2,\dots,Q\}, k \in B\}$ where copy i_p^k of node $i \in V$ indicates that node i included in a solution at position p on the path starting at black node k.

Again, each solution visits precisely one copy of each original node. The arc set is defined as $A_{\text{QB}} = \bigcup_{k\in B} A_{\text{QB}}(k)$ where for each $k \in B$, $A_{\text{QB}}(k) = \{(i_p^k, j_{p+1}^k) \mid \{i_p^k, j_p^k\} \subset V_{\text{QB}}, (i,j) \in A\} \cup \{(i_p^k, j_0^j) \mid (i,j) \in A, j \in B\setminus\{k\}\}$ describes the arcs that might be contained in the path from k to its succeeding black node. Each of the above defined variables $Y_{ij}^{k,p}$ is either associated with arc (i_{p-1}^k, j_p^k) (if $j \in W$) or with arc (i_{p-1}^k, j_0^j) (if $j \in B$). Figure 3 shows the graph G_{QB} corresponding to the instance given in Figure 1 together with an embedding of the solution provided in the same figure.

As in the case of the model of the previous subsection, we can also consider specific layered graph cut inequalities. These cut inequalities also result from the observation that every node must be reachable from the source node 1_0^0 again (we assume that node 1 is the start of the tour). In particular, we consider the set of inequalities (29) ensuring that at least one arc must cross each cut separating the set of black nodes from all copies of a given white node.

$$Y(\delta^-(S)) \ge 1 \qquad \forall S \subseteq V_{\text{QB}} \setminus \{j_0^j : j \in B\}, \exists i \in W \mid \{i_p^k, k \in B, p \in \{1,2,\dots,Q\}\} \subseteq S \qquad (29)$$

We denote by PDPS^{++}, the model PDPS$^+$ augmented with these inequalities.

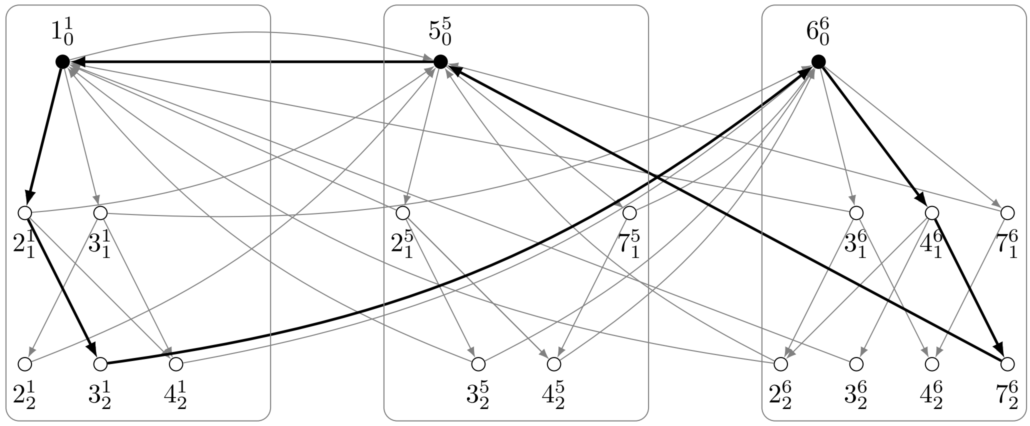

Figure 3: Layered digraph $G_{\text{QB}} = (V_{\text{QB}}, A_{\text{QB}})$ corresponding to the instance given in Figure 1. Bold arcs represent the solution shown in Figure 1. Nodes 4_1^1, 7_1^1, 7_1^2, 3_1^5, 4_1^5, 2_2^5, 7_2^5, and 2_1^6 that are not reachable are not shown.

4 Alternative Time-Dependent Reformulations

In this section, we discuss alternative time-dependent reformulations that result from the fact that the BWTSP has two independent resource constraints. All formulations introduced in Section 3 are based on using additional variables to make explicit the position of arcs within the different path segments. The second resource constraint (i.e., the distance constraint) has been guaranteed by additional inequalities. Based on the extensive discussion of such models in Section 3 it is not too hard to observe that similar formulations can be obtained through the consideration of time-dependent variables (and layered graphs) that focus on the distance of arcs within each segment and therefore implicitly ensure that this distance does not exceed the given threshold L. The hop constraint is then guaranteed by additional constraints in a similar way that the distance constraints were guaranteed in the models of the previous section.

Two such distance-dependent formulations DD and DDPS are easily obtained by appropriate modifications of the position-dependent formulations introduced in Section 3. Formulation DD is a pure distance-dependent formulation that is based on additional variables $\hat{X}_{ij}^d \in \{0,1\}$, $d \in \{1,2,\ldots,L\}$, $\forall (i,j) \in A$, that indicate whether a path to j from the previous black node using arc (i,j) has a total distance of exactly d. Formulation DD implicitly satisfies the distance-constraints and ensures the hop-constraints by using path elimination constraints that are based on defining the set of hop-infeasible paths $\mathcal{P}_{\text{hinf}} \subset 2^E$ between two black nodes in G that do not contain any further black nodes and that violate the hop constraint, i.e., $|P| > Q+1$, $\forall P \in \mathcal{P}_{\text{hinf}}$.

Similarly, DDPS is a distance-dependent reformulation of the path segment model which uses variables $\hat{Y}_{ij}^{k,d} \in \{0,1\}$, $\forall k \in B$, $\forall d \in \{1,2,\ldots,L\}$, $\forall (i,j) \in A$, implicitly encoding the distance of node j on the path from black node k provided that arc (i,j) is used. Variants DD$^+$ and DDPS$^+$ are obtained by including appropriately modified versions of the different two-cycle inequalities introduced previously. We also consider models DD^{++} and DDPS^{++} based on transferring the layered graph interpretation of the two models to the case of distance-constraints and including the appropriately modified layered graph cuts to white nodes. Since the formulations are rather similar to the ones introduced in Section 3 we refrain from explicitly giving further details of them. An overview is, however, provided in Table 1.

Besides exchanging the roles of the two resource constraints, one may consider and derive formulations that implicitly ensure the two constraints, hop and distance together, through appropriate reformulations. In our computational study in Section 7 we will consider two such variants. The reason for including them in our study is to evaluate the improvement (in terms of strength of the formulations) that can be achieved by using such more complicated reformulations. We believe that the BWTSP with these two resource constraints is a quite interesting problem to make a study of such reformulations since the two resources give several possibilities for defining layered graphs with different properties. Making them more attractive to use in practice (maybe through the use of decomposition methods) in a more general context is an important topic for further research.

Formulation PDDD is based on simultaneously considering, at the same time, the pure position-dependent reformulation PD from Section 3.1 and its corresponding distance-dependent reformulation DD. The two sets of extended variables $\mathbf{X}$ and $\hat{\mathbf{X}}$ are linked together via the edge design variables $\mathbf{x}$ due to the linking constraints (17) and the analogous linking constraints of formulation DD. It might be interesting to study possibly existing stronger linking constraints that exploit additional relations between variables $\mathbf{X}$ and $\hat{\mathbf{X}}$ in future research. Clearly, the path elimination constraints with respect to both the distance and the hop resource become redundant and are therefore not included in formulation PDDD and its variants PDDD$^+$ and PDDD^{++} with two-cycle inequalities and layered graph cuts, respectively.

In our study we go a step further and even consider a formulation 3PD that combines information about the position of the arc and its distance to the beginning of the segment. That is, it considers variables $X_{ij}^{p,d}$, $\forall p \in \{1, 2, \ldots, Q+1\}$, $\forall d \in \{1, 2, \ldots, L\}$, $\forall (i,j) \in A$, that make explicit both the position and the distance of arcs within their path segments. Such a formulation can be represented by using a three-dimensional layered graph (see, e.g., Gouveia and Ruthmair [11], Gouveia et al. [16] for recent articles considering such graphs). Again 3PD$^+$ and 3PD^{++} are obtained by including adequate two-cycle inequalities and layered graph cuts.

Finally, we observe that further interesting model variants that might be considered in future work could be obtained by ”enlarging” the models described in the two previous paragraphs by adding information about the black node originating each segment. More precisely, we refer to models based on similar ideas as formulations PDDD and 3PD but which combine the ideas of PDPS and DDPS instead of PD and DD, respectively.

As these formulations are not likely to be of much use in practice before obtaining a better understanding on how to solve them, we do not include results obtained from these variants in the current study.

5 Theoretical Comparison

In this section, we present a few results that relate the LP relaxations of the formulations introduced in this article, see Table 1 for an overview.

Table 1: Overview on the formulations introduced in this article. For each formulation $\mathcal{M}$ variants $\mathcal{M}^+$ and $\mathcal{M}^{++}$ are considered. The latter variant does not exist for the path segment model PS. Variant $\mathcal{M}^+$ includes specific two-cycle inequalities and variant $\mathcal{M}^{++}$ additionally includes layered graph cuts. For each formulation, its name, description, information about the used extended variables and the particular hop- or distance constraints used are given. Thereby, a dash (-) indicates that the hop- or distance constraints are implicitly satisfied by a formulation.

Name	Description	Variables		Distance Constraints	Hop Constraints
PS	path segm.	x_{ij}^k	$k \in B$	$\sum_{e=\{i,j\}\in E} d_e(x_{ij}^k + x_{ji}^k) \leq L, \forall k \in B$	$\sum_{e=\{i,j\}\in E} (x_{ij}^k + x_{ji}^k) \leq Q+1, \forall k \in B$
PD	pos.-dep.	X_{ij}^p	$p \in \{1,\ldots,Q+1\}$	$\sum_{e\in P} x_e \leq \lvert P\rvert - 1, \forall P \in \mathcal{P}_{\text{inf}}$	-
PDPS	pos.-dep. path segm.	$Y_{ij}^{k,p}$	$k \in B$, $p \in \{1,\ldots,Q+1\}$	$\sum_{p=1}^{Q+1} \sum_{e=\{i,j\}\in E} d_e(Y_{ij}^{k,p} + Y_{ji}^{k,p}) \leq L$, $\forall k \in B$	-
DD	dist.-dep.	$\hat{X}_{ij}^d$	$d \in \{1,\ldots,L\}$	-	$\sum_{e\in P} x_e \leq \lvert P\rvert - 1, \forall P \in \mathcal{P}_{\text{hinf}}$
DDPS	dist.-dep. path segm.	$\hat{Y}_{ij}^{k,d}$	$k \in B$, $d \in \{1,\ldots,L\}$	-	$\sum_{d=1}^{L} \sum_{\{i,j\}\in E} \tilde{Y}_{ij}^{k,d} + \tilde{Y}_{ji}^{k,d} \leq Q+1$, $\forall k \in B$
PDDD	pos.-dep.+ dist.-dep.	X_{ij}^p, $\hat{X}_{ij}^d$	$p \in \{1,\ldots,Q+1\}$, $d \in \{1,\ldots,L\}$	-	-
3PD	pos.- and dist.-dep.	X_{ij}^{pd}	$p \in \{1,\ldots,Q+1\}$, $d \in \{1,\ldots,L\}$	-	-

We emphasize that one cannot establish dominance relations between many of the models since they incorporate different and unrelated sets of constraints to model the same set of requirements. Consider, for example the inequalities for modeling the hop- and distance requirements. The sequence of formulations in this article shows a substitution of exponentially sized path elimination constraints by simpler cardinality constraints that are based on using additional variables with additional information. These two modeling approaches are incomparable thus indicating that most pairs of formulations are incomparable. This is also confirmed by our computational results, see Section 7.2. One example is obvious and is given by the two most basic formulations PS and PD that include different sets of constraints to model the distance requirements.

In some cases it is, however, possible to establish relations based on common knowledge about time-dependent reformulations. Below, we point out those cases that we were able to identify. Thereby, for formulation M, we denote by $P(M)$ the polyhedron induced by its LP relaxation and by $\text{proj}_{\mathbf{x}}(P(M))$ the orthogonal projection of polyhedron $P(M)$ onto the space of undirected edge design variables that are included in each of the considered formulations. A formulation M is considered to be (i) at least as strong as formulation M' if $\text{proj}_{\mathbf{x}}(P(M)) \subseteq \text{proj}_{\mathbf{x}}(P(M'))$ holds for all instances; (ii) stronger than M' if $\text{proj}_{\mathbf{x}}(P(M)) \subseteq \text{proj}_{\mathbf{x}}(P(M'))$ holds for all instances and there exist instances for which the inclusion is strict; (iii) incomparable to M' if there exist instances for which $\text{proj}_{\mathbf{x}}(P(M)) \subseteq \text{proj}_{\mathbf{x}}(P(M'))$ holds and instances for which $\text{proj}_{\mathbf{x}}(P(M')) \subseteq \text{proj}_{\mathbf{x}}(P(M))$ holds.

Theorem 1. *Formulation PDPS is stronger than PS.*

Proof. We show that model PDPS implies model PS by using the linking constraints $\sum_{p=1}^{Q+1} Y_{ij}^{k,p} = x_{ij}^k$, $\forall (i,j) \in A$, $\forall k \in B$, that relate the variables of the two formulations. Using these equations, we observe that constraints (22), (23), (25), and (26) in PDPS are equivalent to constraints (7), (8), (10), and (12) in PS, respectively. Flow conservation constraints (9) in PS are obtained by aggregating constraints (24) in PDPS over all positions p and then using the linking equations.

It remains to show that the hop constraints (11) are also implied by PDPS. We generalize a proof given in Gouveia and Voß [12] to the case of multiple source (i.e., black) nodes. We first show that

$$\sum_{\{i,j\} \in E} Y_{ij}^{k,p} \leq 1 \tag{30}$$

holds for every position $p \in \{1, 2, \ldots, Q+1\}$. The result then follows by summation over all positions and using the above linking equations.

First note that validity of (30) for $p = 1$ follows from (22). For $p = 2$, we observe that

$$\sum_{\{i,j\} \in E} Y_{ij}^{k,2} = \sum_{i \in W} \sum_{\{i,j\} \in E} Y_{ij}^{k,2} = \sum_{i \in W} \sum_{\{k,i\} \in E} Y_{ki}^{k,1} \leq \sum_{\{i,j\} \in E} Y_{ij}^{k,1} \leq 1.$$

Thereby, the second equation follows from summing the flow conservation constraints (24) for all white nodes $i \in W$ and the latter inequality from validity of (30) for $p = 1$. Analogously, we obtain

$$\sum_{\{i,j\} \in E} Y_{ij}^{k,p} = \sum_{i \in W} \sum_{\{i,j\} \in E} Y_{ij}^{k,p} = \sum_{i \in W} \sum_{\{j,i\} \in E} Y_{ji}^{k,p-1} \leq \sum_{\{j,i\} \in E} Y_{ji}^{k,p-1} \leq 1$$

for $p \in \{2, 3, \ldots, Q+1\}$. As before, the second equation is obtained from summation of (24) over all white nodes $i \in W$ and the last inequality by inductively considering validity of (30) for $p - 1$.

Finally, since there are instances, cf. Table 2, for which PDPS produces a stronger LP bound than PS we can conclude that the implication is strict. □

On the contrary, PD is incomparable to PDPS because of the aforementioned different types of constraints for modeling the distance constraints. Considering an artificial formulation PDPS' which is obtained by replacing the distance constraints (26) by the path elimination constraints of model PD rewritten with the variables $\mathbf{Y}$ under the linking constraints (25), we obtain

Theorem 2. *Formulation PDPS' is at least as strong as PD.*

Proof. Similar to the proof of Theorem 1, we use linking equalities $\sum_{k \in B} Y_{ij}^{k,p} = X_{ij}^p$, $\forall (i,j) \in A$, $\forall p \in \{1, 2, \ldots, Q+1\}$, to show that constraints (22), (23), and (25) in PDPS' are equivalent to constraints (14), (15), and (17) in PD, respectively. Flow conservation constraints (16) in PD are obtained by aggregating constraints (24) in PDPS' over all black nodes $k \in B$. The distance limit is ensured with the path elimination constraints (18) in both models. □

We note, however, that computational experiments indicate that formulation PDPS is already significantly stronger than PD in practice and we therefore do not consider formulation PDPS' in the remainder of the text.

Finally, we provide Theorem 3 and note that analogous results to the previous theorems can be stated for the alternative formulations that implicitly ensure the distance constraints and use different types of path eliminations constraints for modeling the hop constraints. The two models in the following result are not explicitly written in the text and thus, the dominance argument given here can only be viewed as a sketch. The arguments, however, have been used (partially) in the proofs of the previous two theorems.

Theorem 3. *Formulation 3PD is stronger than PDDD.*

Proof. We use linking constraints $\sum_{d=1}^{L} X_{ij}^{pd} = X_{ij}^{p}$, $\forall (i,j) \in A$, $\forall p \in \{1, 2, \ldots, Q+1\}$, and $\sum_{p=1}^{Q+1} X_{ij}^{pd} = \hat{X}_{ij}^{d}$, $\forall (i,j) \in A$, $\forall d \in \{1, 2, \ldots, L\}$, to show that out- and in-degree constraints for black nodes and the linking constraints to the edge design variables in formulations PDDD and 3PD are equivalent. The flow conservation constraints for white nodes in PDDD are implied by aggregating the flow conservation constraints in 3PD for each position and distance, respectively. Explicit hop and distance constraints are not needed since they are satisfied implicitly by both models. □

6 Branch-and-Cut Algorithms

Branch-and-cut algorithms based on IBM ILOG CPLEX 12.6.3 have been implemented in C++ for all proposed models. In this section we provide details about instance preprocessing, the cutting plane algorithms, and methods to obtain primal bounds. Default settings for all CPLEX parameters are used except that multi-threading is deactivated and that branching on the $\mathbf{x}$-variables is prioritized. In the following, let $\overline{\mathbf{x}}$ ($\overline{\mathbf{X}}$, etc.) denote the LP relaxation values of variables $\mathbf{x}$ ($\mathbf{X}$, etc.).

6.1 Preprocessing

We apply preprocessing based on computing the minimum number of white nodes $Q_{\min}$ contained in any path between two successive black nodes. To this end, we observe that this minimum is obtained for a path when all other paths contain Q white nodes and if $|W| > Q(|B|-1)$, and is equal to zero otherwise, i.e., we have

$$Q_{\min} = \max\{0, |W| - Q(|B| - 1)\}.$$

This value is used to limit, from below, the left-hand side of the hop constraints in Table 1 (excluding path elimination constraints (18)). For formulation PD it is also used to fix all position-dependent variables that would yield too short paths to zero, i.e., $X_{ij}^{p} = 0$ for all $p \leq Q_{\min}$ and for all $j \in B$. Similar variable fixing is performed for the position dependent variables of formulations PDPS, PDDD, and 3PD.

6.2 Cutting Plane Algorithm

The following sets of valid inequalities are not included a priori in the model but are added dynamically to the model by a cutting plane approach. Besides all introduced families of inequalities of exponential size, we also dynamically separate several large, but polynomially sized families of inequalities as this strategy turned out to be beneficial in preliminary computational experiments. We search for violated inequalities in the order given below and do not continue with the next set but instead resolve the LP relaxation in case at least one violated cut of some set has been identified.

- Violated connectivity cut constraints (3) are found by max-flow computations (using the algorithm of Cherkassky and Goldberg [6]) in a support graph with capacities based on the current LP solution $\overline{\mathbf{x}}$, see, e.g., Padberg and Rinaldi [23].

- Path elimination constraints (18) are only separated for integral solutions since it turned out in preliminary tests that too many violated inequalities are added when applying the separation procedure proposed by Ascheuer et al. [3] to fractional solutions. Since at this point of the cutting plane procedure integral solutions are already TSP tours, finding infeasible paths can be done in linear time.

- Flow balance constraints (9), (16), (24), and corresponding sets in further layered graphs are separated by enumeration.

- 2-cycle elimination constraints (19)–(20), (27)–(28), and corresponding sets in further layered graphs are separated by enumeration.

- Layered graph cuts (21), (29), and corresponding sets in further layered graphs are separated similar to connectivity cuts (3) in graph G. The main difference is that we have to guarantee that all copies of at least one white node are in the target set which is done by setting high capacities on arcs from all copies associated to one particular white node to an artificial flow sink.

To appropriately deal with a possibly large number of added inequalities and slow cutting plane convergence [cf. 28], we apply the following rules:

- Suppose that a valid inequality in layered graph G_{Q} has the form $X(A) \geq b$. We only add a violated cut if $\overline{X}(A) < \Delta \cdot b$ with $\Delta \in (0, 1]$. We use $\Delta = 0.5$ for violated inequalities (21), (29), and corresponding inequalities in the layered graphs induced by the modeling ideas from Section 4, and $\Delta = 1.0$ for all other valid inequalities.

- If the LP relaxation value does not increase in the last five cutting plane iterations within a branch-and-bound node we continue with branching.

- After solving a maximum flow to search for violated cut sets we might obtain multiple minimum cuts. In this case we consider the minimum cuts with the smallest and the largest cutset, respectively, and finally add the one yielding a minimal number of cut arcs.

6.3 Primal Heuristics

Since primal bounds are essential for pruning the branch-and-bound tree and fixing variables based on reduced costs we use heuristics in some nodes of the tree that are similar to the ones proposed by Gouveia and Ruthmair [11]. These heuristics are called after each cut iteration in the root node, in every 5th branch-and-bound node within the first 100 nodes, in every 25th node within the first 1000 nodes, in every 100th node in the first 5000 nodes, and in every 500th node in the rest of the nodes. In the remaining of this subsection we give a brief overview of the used heuristics, see also Algorithm 1).

To construct a feasible solution we apply a nearest neighbor heuristic guided by the LP solution of the current branch-and-bound node in the sense that we use modified edge costs $c'_e = c_e(1 - \overline{x}_e)$ for each $e \in E$: The TSP tour is extended step-by-step by choosing the cheapest unvisited successor node without violating the hop constraints but ignoring distance constraints. In complete graphs and by considering the minimal number of white nodes $Q_{\min}$ after each black node we are able to always obtain a hop-feasible tour with this construction heuristic. The distance constraints might, however, be violated.

To repair and further improve a solution, we run a variant of the generalized variable neighborhood search (GVNS) [18]. With a probability of 50% we choose the global best solution as starting point in a GVNS iteration, otherwise we use the best solution found in the current heuristic call. To locally improve the solution an embedded variable neighborhood descent (VND) based on two neighborhood structures is applied: i) One node is shifted to another position in the tour, and ii) two nodes are swapped. In each iteration of the VND we search for moves in both neighborhoods which either reduce the distance violation (if the solution is not yet feasible) or improve solution quality. We choose randomly among the five best moves whereas reducing distance violation is preferred to cost improvement. To diversify the solution in the shaking phase of the GVNS we apply $\lceil 0.05 \cdot |V| \rceil$ random node shifts while ensuring hop-feasibility and without increasing distance violation. If after two GVNS iterations no new global best solution can be found the number of shaking moves is increased by one (up to at most $\lceil 0.3 \cdot |V| \rceil$). We

Input: graph G, LP solution $\overline{\mathbf{x}}$, incumbent solution S_g

Output: solution S for the BWTSP (empty if none can be found)

`// Solution construction`

$S =< 1 >$ `// sequence of nodes starting in node 1`

while $|S| < |V|$ **do**

  append to S the cheapest hop-feasible successor (based on costs $c'_e = c_e(1 - \overline{x}_e), \forall e \in E$)

end

`// Solution repair and improvement`

$I = 0,\ N_s = \lceil 0.05 \cdot |V| \rceil$

while $I < 20$ **do**

  $S' = S_g$ (with $\mathcal{P} = 50\%$), else S

  apply to S' consecutively N_s random feasible node shifts and swaps

  while S' *can be repaired or improved* **do**

    apply to S' randomly one of the five most violation-reducing / cost-improving moves out of all hop-feasible node shifts and swaps

  end

  if $c(S') < c(S)$ **then** $S = S',\ I = 0,\ N_s = \lceil 0.05 \cdot |V| \rceil$

  else $I = I + 1,\ N_s = \min\{N_s + 1, \lceil 0.3 \cdot |V| \rceil\}$

end

return S

Algorithm 1: Primal Heuristic

stop the GVNS if after 20 iterations no new global best solution is identified. Final solutions violating the distance constraints are dropped. The latter did, however, rarely happen in our computational experiments.

7 Computational Results

This section reports and discusses experimental results obtained by our solution approaches for instances of the BWTSP. Throughout this section, we will use the name of the models to refer to the corresponding branch-and-cut algorithms, e.g., PS^+ denotes the branch-and-cut algorithm of the path segment formulation PS augmented with 2-cycle inequalities.

7.1 Configuration and Instances

Each experiment has been performed on a single core of an Intel Xeon E5-2670v2 machine with 2.5 GHz. A memory limit of 5GB and a time limit (CPU time) of 7200 seconds has been applied to each individual test run.

We used three different classes of benchmark instances: The set MUT has been generated by Muter [22] using a strategy similar to Ghiani et al. [7]. Basically, $|V|$ points are randomly placed in a 1000x1000 square with costs and distances corresponding to the Euclidean distances. For each number of nodes $|V| \in \{20, 40, 60, 80\}$ five instances are available which are tested with three different values for parameter $\alpha \in \{0.2, 0.3, 0.4\}$ defining the hop constraint $Q := \alpha|V|$ and parameter $\beta \in \{1.00, 1.33, 1.67\}$ defining the number of black nodes $|B| := \beta b_{\min}$, and distance constraint L (defined later), resulting in 540 instances. Thereby, $b_{\min}$ is the smallest number of black nodes required for feasibility, i.e., the smallest integer satisfying $b_{\min} = \lceil (|V| - b_{\min})/Q \rceil$.

The insights one can obtain from computational experiments based on set MUT are somehow limited because of the following two reasons: (i) Costs and distances are identical for each instance. Thus, one cannot analyze situations in which the two edge resources are uncorrelated. (ii) Black nodes are randomly chosen from the nodes in V. Thus, one cannot compare the instance difficulty for the same values of $|V|$, Q, $|B|$, and L, when the distribution of the black nodes in the square is changed.

We have therefore created two new sets, GLR1 and GLR2, where nodes and edge costs are generated in the same way as in MUT but distances are set independent from the costs to a random integer value in $\{1, 2, \ldots, 10\}$. Further, we differentiate between two distributions of black nodes in the square: In the

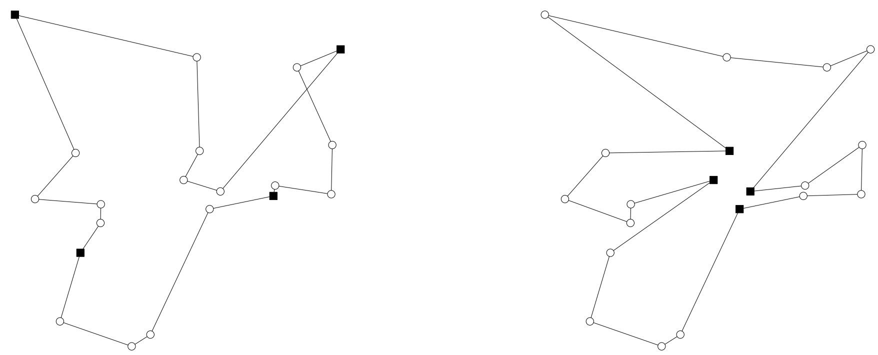

Figure 4: Optimal solutions for the same instance with $|V| = 20, |B| = 4, Q = 4, L = \infty$, with two different distributions of black nodes. The optimal values for the left-hand and right-hand solution are 4311 and 5316, respectively.

set GLR1 the black nodes are spread in the square by first randomly selecting an initial black node and then iteratively choosing the node maximizing the minimal Euclidean distance to the already fixed black nodes. In contrast, in GLR2 we choose the set of black nodes to be the points nearest to the center of the square. See Figure 4 for an exemplary instance with the two distributions described above. For each number of nodes $|V| \in \{20, 40, 60, 80\}$ five instances are generated which are tested with two different values for $Q \in \{4, 8\}$, three values for $\beta \in \{1.00, 1.33, 1.67\}$ and L, resulting in 360 instances for set GLR1 and GLR2, respectively.

Note that choosing reasonable values for the distance limit L is not trivial. We use the same strategy as in Ghiani et al. [7]: $L := \gamma \cdot L_{\max}$, with $\gamma \in (0, 1]$ and $L_{\max}$ being the length of the longest path between two consecutive black nodes in the optimal solution for $L = \infty$. We used model PD$^+$ with a time limit of one day and a memory limit of 16GB to compute these solutions. Since we were not able to obtain an optimal solution for $L = \infty$ for some of the instances we instead use the best solution found within the time and memory limits to compute $L_{\max}$. For set MUT we use $\gamma \in \{0.7, 0.8\}$ according to Muter [22] which leads to infeasibility in several cases (mainly due to the correlation between costs and distances), whereas for sets GLR1/2 and $\gamma \in \{0.6, 0.8\}$ all instances are feasible.

7.2 LP Relaxation Bounds

Table 2 compares the LP relaxation strength of the different models shown in Table 1 in all variants, i.e., plain (“ ”), with 2-cycle inequalities (“$^+$”), and with both 2-cycle and layered graph cut inequalities (“$^{++}$”) where applicable. We only present LP results for $|V| = 20$ and sets GLR1/2 as these results are sufficient to show the relations between the models. For larger instances the LP gaps slightly increase while the relative differences between the models remain similar. Let c_{OPT} be the optimal integer solution value and c_{LP} be the optimal value of the LP relaxation. The LP gap value for one particular model and instance in Table 2 is computed as $(c_{\text{OPT}} - c_{\text{LP}})/c_{\text{OPT}}$ (given in percent). LP relaxation values have been computed using the previously described branch-and-cut algorithms by setting $\Delta = 1$ and disabling early branching. Additionally, we use the separation heuristic in Ascheuer et al. [3] to find violated path elimination inequalities for models PD and DD (and their variants) also for fractional solutions. Note that we do not consider the distance-dependent formulations (i.e., DD, DDPS, PDDD, and 3PD) when the distance constraints are not restrictive, i.e., for $\gamma = 1.0$ which is equivalent to $L = \infty$.

We observe that the LP gaps of most models reduce significantly when adding the 2-cycle inequalities (“$^+$”) to the plain model. On the contrary, the additional gap reduction obtained from adding layered graph cut inequalities (“$^{++}$”) is only marginal in most cases. Thus, the additional separation effort may not pay off. The latter is also confirmed by the branch-and-cut results in Section 7.3 which show that the “$^+$” variants mostly outperform the other options. The results from Table 2 confirm all theoretical relations discussed in Section 5. Several cases—especially those with small γ values—show that models PS and PD and also PD and PDPS are unrelated. The results also confirm that modeling the hop- or distance requirements by cardinality constraints (in some extended variable space) does not dominate the use of path elimination constraints. Since, the latter are separated heuristically, we cannot conclude,

Table 2: Average LP gaps (in %) for instances with $|V| = 20$. Bold values denote the best algorithms for a set. In rows marked with "*" only 4 of 5 instances are used for the average gaps since model (3PD)$^{++}$ was not able to compute the correct LP value for one instance within the time limit.

				PS		PD			PDPS			DD			DDPS			PDDD			3PD			
Set	Q	β	γ		+		+	++		+	++		+	++		+	++		+	++		+	++	
GLR1	4	1.00	0.6	16.0	16.0	16.6	15.7	15.4	9.6	8.6	8.6	11.0	9.3	8.7	10.2	8.2	7.8	7.8	5.5	5.0	3.4	1.9	**1.5**	
			0.8	13.3	13.3	7.0	6.0	5.8	6.1	4.6	4.6	9.9	7.6	7.4	8.7	7.5	7.3	5.2	3.2	2.7	2.8	1.3	**0.7**	
			1.0	9.3	9.3	2.1	1.2	1.1	1.7	0.6	**0.5**	-	-	-	-	-	-	-	-	-	-	-	-	
		1.33	0.6	15.7	15.6	13.3	13.2	13.2	14.9	14.7	14.7	6.7	3.9	3.9	4.0	**2.9**	**2.9**	7.3	3.8	3.8	6.7	3.8	3.8	
			0.8	5.9	5.9	3.5	3.3	3.3	4.8	4.7	4.7	2.0	1.2	1.0	1.6	1.3	1.3	2.0	0.8	0.8	1.4	0.8	**0.7**	*
			1.0	1.3	1.3	0.6	0.5	0.5	**0.4**	**0.4**	**0.4**	-	-	-	-	-	-	-	-	-	-	-	-	
		1.67	0.6	10.8	10.8	7.1	7.0	7.0	10.0	9.7	9.7	2.7	1.2	1.2	1.0	**0.8**	**0.8**	3.6	1.2	1.2	2.8	1.2	1.2	
			0.8	5.5	5.5	3.4	3.2	3.2	4.7	4.5	4.5	1.9	1.3	1.3	1.7	1.5	1.5	2.0	**1.1**	**1.1**	1.5	**1.1**	**1.1**	
			1.0	1.3	1.3	0.6	0.5	0.5	0.5	**0.4**	**0.4**	-	-	-	-	-	-	-	-	-	-	-	-	
	8	1.00	0.6	8.2	8.2	8.8	8.2	8.1	7.4	7.0	6.8	4.0	1.9	1.3	2.4	0.9	**0.7**	4.0	1.9	1.3	3.8	1.8	1.3	*
			0.8	5.1	5.1	4.0	3.5	3.4	4.1	3.7	3.6	3.2	2.1	1.8	2.7	1.7	**1.6**	3.2	2.1	1.8	3.0	2.0	1.7	*
			1.0	3.6	3.6	2.4	1.9	**1.8**	2.3	**1.8**	**1.8**	-	-	-	-	-	-	-	-	-	-	-	-	
		1.33	0.6	6.8	6.8	7.4	7.4	7.4	6.7	6.6	6.5	2.5	0.9	0.7	1.4	0.5	**0.3**	2.6	0.9	0.7	2.5	0.9	0.6	
			0.8	4.1	4.1	3.3	3.3	3.3	4.0	4.0	3.9	2.1	1.2	1.0	1.5	0.9	**0.8**	2.3	1.2	1.0	2.2	1.2	1.0	
			1.0	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	-	-	-	-	-	-	-	-	-	-	-	-	
		1.67	0.6	11.9	11.9	10.1	10.1	10.1	11.7	11.6	11.6	4.5	2.6	2.6	2.8	**1.8**	**1.8**	5.1	2.6	2.6	4.8	2.6	2.6	
			0.8	3.9	3.9	2.6	2.6	2.6	3.7	3.7	3.7	0.5	**0.1**	**0.1**	0.3	**0.1**	**0.1**	0.8	**0.1**	**0.1**	0.8	**0.1**	**0.1**	*
			1.0	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	-	-	-	-	-	-	-	-	-	-	-	-	
GLR2	4	1.00	0.6	26.6	26.6	17.0	15.9	15.5	12.2	11.1	11.0	14.6	12.2	11.8	13.4	11.0	10.9	10.0	7.6	7.3	5.9	4.7	**4.2**	
			0.8	24.9	24.9	8.6	7.1	6.5	7.6	6.0	5.6	15.7	13.0	12.7	14.2	12.2	11.6	6.5	4.3	3.8	4.7	3.4	**2.7**	
			1.0	22.8	22.8	5.3	3.8	3.1	4.8	3.4	**2.9**	-	-	-	-	-	-	-	-	-	-	-	-	
		1.33	0.6	15.6	15.6	8.4	7.5	7.3	9.4	8.9	8.7	5.0	2.7	2.3	3.2	1.9	1.7	4.1	1.8	1.5	3.2	1.6	**1.3**	
			0.8	11.7	11.7	5.4	4.7	4.7	6.6	6.2	6.1	6.8	5.6	5.2	6.4	5.5	5.5	4.9	2.5	2.3	4.1	2.1	**2.0**	*
			1.0	8.5	8.5	2.6	1.9	1.7	2.3	1.8	**1.6**	-	-	-	-	-	-	-	-	-	-	-	-	
		1.67	0.6	13.9	13.9	8.9	8.2	8.2	9.9	9.6	9.6	4.7	2.8	2.6	3.4	2.3	2.2	4.3	2.2	2.0	3.3	2.0	**1.8**	
			0.8	9.9	9.9	4.9	4.5	4.5	6.0	5.8	5.8	5.5	4.3	4.1	5.4	4.2	4.2	4.2	2.4	2.2	3.5	2.2	**2.1**	
			1.0	6.1	6.1	2.3	**1.9**	**1.9**	2.1	**1.9**	**1.9**	-	-	-	-	-	-	-	-	-	-	-	-	
	8	1.00	0.6	17.2	17.2	17.2	16.0	15.2	13.0	11.5	10.4	7.8	5.6	4.6	6.5	4.9	4.0	7.0	4.8	3.6	6.5	4.5	**3.3**	*
			0.8	14.0	14.0	9.6	8.1	7.3	9.1	7.6	6.5	7.7	5.8	5.0	7.0	5.5	5.0	6.7	4.4	3.1	5.8	3.8	**2.3**	*
			1.0	12.0	12.0	6.7	5.1	4.4	6.5	5.0	**4.0**	-	-	-	-	-	-	-	-	-	-	-	-	
		1.33	0.6	15.5	15.5	14.1	13.3	12.9	13.3	12.4	11.8	4.7	2.5	2.4	3.7	**2.0**	**2.0**	4.7	2.5	2.4	4.7	2.5	2.3	
			0.8	8.9	8.9	6.9	6.0	5.5	6.6	5.7	5.5	3.7	1.5	0.8	2.8	1.1	**0.6**	3.6	1.4	0.8	3.5	1.3	**0.6**	
			1.0	4.2	4.2	2.3	1.4	**1.1**	1.9	**1.1**	**1.1**	-	-	-	-	-	-	-	-	-	-	-	-	
		1.67	0.6	11.6	11.6	9.2	8.8	8.7	10.7	10.4	10.2	3.2	0.9	0.7	1.7	0.5	**0.3**	3.3	0.9	0.7	2.6	0.9	0.7	*
			0.8	5.1	5.1	4.0	3.4	3.2	4.3	3.6	3.4	2.1	0.7	0.5	1.5	0.6	0.5	2.2	0.7	0.5	2.0	0.7	**0.3**	
			1.0	2.1	2.1	1.4	0.7	0.5	1.3	0.6	**0.4**	-	-	-	-	-	-	-	-	-	-	-	-	

however, that the two modeling approaches yield incomparable models (though this is clearly indicated).

When considering instances with $\gamma = 1.0$, i.e., without distance constraint, the simple knapsack inequalities (11) in model PS to ensure the hop constraint lead to weak LP bounds compared to the position-dependent reformulations PD and PDPS. The additional disaggregation of the black-to-black paths in PDPS, however, does not seem to pay off in terms of LP gap reduction. In general the gaps typically increase with decreasing values of γ, i.e., when the distance constraints are more restrictive. As expected, the distance-dependent models DD and DDPS are beneficial for very tight distance constraints for which they yield significantly smaller LP gaps as the position-dependent variants. This gap reduction can even be increased when combining hop- and distance-dependent formulations in PDDD and 3PD. We also observe, however, that for $\gamma > 0.6$ the gaps obtained by PD are typically not much larger than those of other models involving much more variables.

7.3 Branch-and-Cut Result Overview

In this section, we aim to identify the most promising among the branch-and-cut algorithms (from each group representing the main modeling ideas used) in order to reduce the number of considered variants before comparing and analyzing them in more detail. Table 3 shows numbers of solved instances (i.e., either optimality or infeasibility has been shown) for PS and all algorithms based on position-dependent reformulations (i.e., PD and PDPS). Likewise, Table 4 provides the same information for those variants implicitly ensuring the distance constraints.

We first observe that instance set GLR2 with all black nodes in the center of the square is harder to solve than GLR1 for all developed algorithms. From Table 3 we conclude that especially PS solves much fewer instances of GLR2 compared to GLR1. Nevertheless, when compared to the other models PS uses fewer variables and constraints which makes it perform better for the larger instances with loose hop- and distance constraints, see, e.g., the results for set MUT.

As anticipated in the previous subsections, the additional disaggregation of the "black-to-black paths" in model PDPS typically does not pay off when compared to PD. A similar picture is obtained from comparing DD and DDPS (and their variants), see Table 4. Again, the potential increase in terms of LP bounds by DDPS is mostly outweighed by the additional time needed to solve the significantly larger models. The only exceptions to this are observed for set MUT when γ is rather restrictive (in which many instances are infeasible). We also conclude that variants PDDD and 3PD implicitly satisfying both the hop and distance constraints seem to perform reasonably well, at least with respect to the total numbers of solved instances. We also conclude that in general, those variants including 2-cycle inequalities seem to slightly outperform those additionally considering layered graph cuts.

Finally, note that a direct comparison to the results of the branch-and-price approach in [22] is only possible for instance set MUT and when $\gamma = 1.0$: The best method from [22] is able to solve 74 instances within 10800 seconds on a hardware similar to ours while our approach based on model PD$^+$ solves 157 instances within 7200 seconds. Since we have different values of $L_{\max}$ (the ones in [22] have been obtained by a heuristic) the instances for restricted L values are not the same. Generating the $L_{\max}$ values in the same way as we did for the other instances, it seems that we solve less instances than Muter ($\gamma = 0.7$: 120 by Muter, 96 by PS$^+$; $\gamma = 0.8$: 115 by Muter, 86 by PS$^+$). To this end, we stress that that the (likely) different values of $L_{\max}$ may significantly influence the overall performance, thus avoiding significant conclusions from this comparison. We also note, that the distance-dependent models heavily depend on the range of the distance values. If we for example would divide all the distance values for the MUT instances by 100 and round up the resulting values, we obtain much better results for the DD$^+$ model ($\gamma = 0.7$: 134; $\gamma = 0.8$: 119).

Different parameter values for α and γ have been used by Ghiani et al. [7]. As will be detailed in A, a direct comparison to them is not meaningful. Nevertheless, some computational results using their parameter values will also be reported in A.

7.4 Branch-and-Cut Result Details

Based on the previous section we select the most promising models, i.e., PS$^+$, PD$^+$, DD$^+$, PDDD$^+$, and 3PD$^+$, and present more detailed results for them in Tables 5, 6, 7, and 8. Let c_{LB} and c_{UB} be the best global lower and upper bounds, respectively, obtained by the algorithm within time and memory limits. The optimality gaps in the tables are given by $(c_{\text{UB}} - c_{\text{LB}})/c_{\text{UB}}$ in percent. If either the lower or

Table 3: Number of instances (out of 20 in each row) solved by our branch-and-cut algorithms based on different models within 7200 seconds and 5 GB memory. Bold values denote the best algorithms for a set and a model type.

				PS		PD			PDPS		
Set	Q/α	β	γ		+		+	++		+	++
GLR1	4	1.00	0.6	3	**4**	**7**	**7**	**7**	**5**	**5**	**5**
			0.8	**5**	**5**	**16**	**16**	**16**	11	**12**	**12**
			1.0	**5**	**5**	17	**19**	**19**	15	**17**	**17**
		1.33	0.6	5	**6**	**12**	**12**	**12**	**6**	5	5
			0.8	11	**13**	19	**20**	**20**	14	**16**	15
			1.0	14	**15**	**20**	**20**	**20**	17	**19**	**19**
		1.67	0.6	9	**10**	**19**	18	**19**	**8**	**8**	**8**
			0.8	**17**	**17**	**20**	**20**	**20**	**19**	**19**	**19**
			1.0	**18**	**18**	**20**	**20**	**20**	**20**	**20**	**20**
	8	1.00	0.6	**6**	5	**6**	**6**	**6**	**6**	**6**	**6**
			0.8	**8**	**8**	10	**11**	10	**9**	**9**	**9**
			1.0	9	**10**	14	**16**	**16**	11	**14**	**14**
		1.33	0.6	**10**	9	**10**	**10**	9	6	**8**	7
			0.8	**16**	15	18	**20**	**20**	11	**13**	12
			1.0	**19**	**19**	19	**20**	**20**	17	**19**	**19**
		1.67	0.6	9	**10**	**11**	**11**	**11**	6	**7**	6
			0.8	**19**	18	**19**	**19**	**19**	**13**	**13**	**13**
			1.0	**19**	**19**	**20**	**20**	**20**	18	**19**	18
GLR2	4	1.00	0.6	**1**	0	5	**6**	**6**	**5**	**5**	**5**
			0.8	1	**3**	**14**	**14**	**14**	9	**12**	**12**
			1.0	**1**	**1**	15	**18**	17	11	**14**	**14**
		1.33	0.6	**4**	**4**	**6**	**6**	**6**	**5**	**5**	**5**
			0.8	**5**	**5**	12	**19**	17	8	**9**	**9**
			1.0	**5**	**5**	17	**19**	18	10	**14**	**14**
		1.67	0.6	**4**	**4**	8	**9**	**9**	**6**	**6**	**6**
			0.8	**6**	5	11	**16**	15	8	**9**	**9**
			1.0	**6**	**6**	17	**19**	**19**	10	**13**	**13**
	8	1.00	0.6	**3**	**3**	**4**	**4**	**4**	**5**	**5**	**5**
			0.8	**5**	**5**	10	**11**	**11**	6	**8**	7
			1.0	**5**	**5**	12	**16**	**16**	9	**12**	11
		1.33	0.6	3	**5**	**4**	**4**	**4**	**5**	4	4
			0.8	**5**	**5**	**8**	**8**	**8**	5	**6**	**6**
			1.0	**5**	**5**	10	**13**	**13**	9	**10**	**10**
		1.67	0.6	**6**	**6**	**6**	**6**	**6**	**6**	**6**	**6**
			0.8	**6**	**6**	9	**12**	**12**	6	**8**	**8**
			1.0	**7**	**7**	12	**14**	**14**	9	**11**	**11**
MUT	0.2	1.00	0.7	**9**	**9**	**9**	**9**	**9**	**8**	**8**	**8**
			0.8	**4**	**4**	**5**	**5**	**5**	**5**	**5**	**5**
			1.0	**7**	6	13	**14**	**14**	12	**13**	**13**
		1.33	0.7	9	**10**	8	**9**	8	**8**	**8**	7
			0.8	**8**	**8**	8	**9**	8	**7**	6	**7**
			1.0	**13**	12	16	**19**	18	**12**	**12**	**12**
		1.67	0.7	**12**	**12**	**10**	**10**	**10**	**10**	**10**	**10**
			0.8	14	**15**	**13**	**13**	12	**11**	**11**	10
			1.0	**18**	**18**	**18**	**18**	**18**	**15**	**15**	**15**
	0.3	1.00	0.7	**10**	**10**	**10**	**10**	**10**	**10**	**10**	**10**
			0.8	6	**7**	**6**	**6**	**6**	**6**	**6**	**6**
			1.0	**17**	**17**	**19**	18	16	14	**16**	**16**
		1.33	0.7	**11**	**11**	**9**	**9**	**9**	**9**	**9**	**9**
			0.8	**12**	**12**	9	**10**	9	**7**	**7**	**7**
			1.0	**19**	**19**	**19**	**19**	**19**	**16**	**16**	**16**
		1.67	0.7	**10**	**10**	**10**	**10**	**10**	**7**	**7**	**7**
			0.8	**11**	10	**10**	**10**	**10**	6	6	**7**
			1.0	**18**	**18**	**19**	**19**	**19**	**15**	**15**	13
	0.4	1.00	0.7	**13**	**13**	**10**	**10**	**10**	**11**	**11**	**11**
			0.8	8	**9**	**8**	**8**	**8**	**8**	**8**	**8**
			1.0	**13**	**13**	**15**	**15**	**15**	**14**	**14**	13
		1.33	0.7	**9**	**9**	**9**	**9**	**9**	**9**	**9**	**9**
			0.8	**9**	8	**8**	**8**	7	**7**	**7**	**7**
			1.0	**19**	**19**	**18**	17	17	**17**	**17**	16
		1.67	0.7	11	**12**	9	**10**	9	**8**	**8**	**8**
			0.8	12	**13**	**11**	**11**	9	**7**	**7**	**7**
			1.0	**18**	**18**	**18**	**18**	**18**	16	**17**	16
			Total	600	**608**	774	**822**	805	619	**664**	652

Table 4: Number of instances (out of 20 in each row) solved by our branch-and-cut algorithms based on different models within 7200 seconds and 5 GB memory. Bold values denote the best algorithms for a set and a model type.

				DD			DDPS			PDDD			3PD		
Set	Q/α	β	γ		+	++		+	++		+	++		+	++
GLR1	4	1.00	0.6	**5**	**5**	**5**	**5**	**5**	**5**	8	**11**	**11**	**14**	**14**	**14**
			0.8	5	**6**	**6**	5	**6**	**6**	10	**13**	12	14	**15**	14
		1.33	0.6	14	**17**	16	**9**	**9**	**9**	12	**18**	16	16	**19**	**19**
			0.8	15	**16**	15	7	**9**	8	16	**19**	18	17	**19**	**19**
		1.67	0.6	18	**20**	**20**	**12**	**12**	**12**	17	**20**	**20**	17	**20**	**20**
			0.8	19	**20**	**20**	**13**	12	12	**20**	**20**	**20**	**20**	**20**	**20**
	8	1.00	0.6	8	**10**	9	5	**7**	**7**	6	**10**	**10**	5	**6**	5
			0.8	7	**8**	**8**	5	6	**7**	7	**9**	8	6	**8**	**8**
		1.33	0.6	13	**16**	15	8	**10**	**10**	11	**13**	**13**	8	11	**12**
			0.8	15	**17**	**17**	**9**	**9**	**9**	14	14	**15**	9	**12**	**12**
		1.67	0.6	16	**20**	19	9	**12**	**12**	13	**14**	**14**	10	**12**	**12**
			0.8	18	**20**	19	**12**	**12**	**12**	17	**18**	17	13	**14**	13
GLR2	4	1.00	0.6	**4**	**4**	**4**	**4**	**4**	**4**	5	**9**	**9**	11	**15**	14
			0.8	**3**	**3**	**3**	**3**	**3**	**3**	8	**13**	12	9	**13**	12
		1.33	0.6	7	**9**	**9**	5	**6**	**6**	10	**11**	**11**	11	**12**	**12**
			0.8	**6**	**6**	**6**	**5**	**5**	**5**	9	9	**10**	10	**12**	11
		1.67	0.6	9	**10**	**10**	5	**7**	**7**	10	13	**14**	12	**15**	**15**
			0.8	**8**	**8**	**8**	5	**6**	**6**	9	**13**	**13**	10	13	**14**
	8	1.00	0.6	6	**9**	8	5	**7**	**7**	6	**9**	8	**5**	**5**	**5**
			0.8	5	**6**	**6**	4	**5**	**5**	6	**7**	6	**5**	**5**	**5**
		1.33	0.6	9	**11**	**11**	5	7	**8**	6	**10**	**10**	5	9	**10**
			0.8	**8**	**8**	**8**	**5**	**5**	**5**	6	**9**	**9**	5	**7**	6
		1.67	0.6	10	**12**	10	6	**8**	**8**	9	**10**	**10**	6	**10**	**10**
			0.8	**9**	**9**	**9**	5	**6**	**6**	8	**9**	**9**	6	**7**	**7**
MUT	0.2	1.00	0.7	**10**	**10**	**10**	**12**	**12**	**12**	**10**	**10**	**10**	**12**	**12**	**12**
			0.8	**4**	**4**	**4**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**
		1.33	0.7	**10**	**10**	**10**	**9**	8	8	**10**	9	9	**8**	**8**	**8**
			0.8	**5**	**5**	**5**	**5**	4	4	**5**	**5**	**5**	**5**	**5**	**5**
		1.67	0.7	**11**	**11**	**11**	**11**	10	10	**11**	10	10	**11**	**11**	**11**
			0.8	**10**	9	9	**9**	7	7	**11**	8	9	**8**	7	7
	0.3	1.00	0.7	**10**	**10**	**10**	**10**	**10**	**10**	**10**	**10**	**10**	**9**	**9**	**9**
			0.8	**5**	**5**	**5**	**6**	**6**	**6**	**6**	5	5	**5**	**5**	**5**
		1.33	0.7	**11**	**11**	**11**	**12**	**12**	**12**	**11**	**11**	**11**	**10**	9	9
			0.8	**8**	**8**	**8**	**8**	**8**	**8**	**8**	7	7	**8**	7	7
		1.67	0.7	**9**	8	8	**10**	**10**	**10**	**9**	8	8	**6**	**6**	**6**
			0.8	**9**	8	8	**8**	7	7	**9**	7	7	**6**	**6**	**6**
	0.4	1.00	0.7	**8**	**8**	**8**	**8**	**8**	**8**	**8**	**8**	**8**	**7**	**7**	**7**
			0.8	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**
		1.33	0.7	**9**	**9**	**9**	**10**	**10**	**10**	**9**	**9**	**9**	**9**	**9**	**9**
			0.8	**7**	**7**	**7**	**7**	**7**	**7**	6	**7**	**7**	**5**	**5**	**5**
		1.67	0.7	**10**	8	8	**9**	**9**	**9**	**10**	8	8	**7**	**7**	**7**
			0.8	**10**	9	9	**9**	8	8	**10**	8	8	**6**	**6**	**6**
			Total	388	**415**	406	309	324	**325**	396	**441**	436	376	**422**	418

upper bound is not available for an instance we do not provide an average gap value but instead use "-" to indicate this situation. If the time or memory limit is reached before proving optimality we use the time limit of 7200 seconds to compute the average CPU time for which a value of "tl" indicates that this average is equal to the time limit. In addition, we show information about the number of instances for which optimality or infeasibility has been shown. The latter only happens for instance set MUT where the numbers of instances that are shown to be infeasible are given in parentheses.

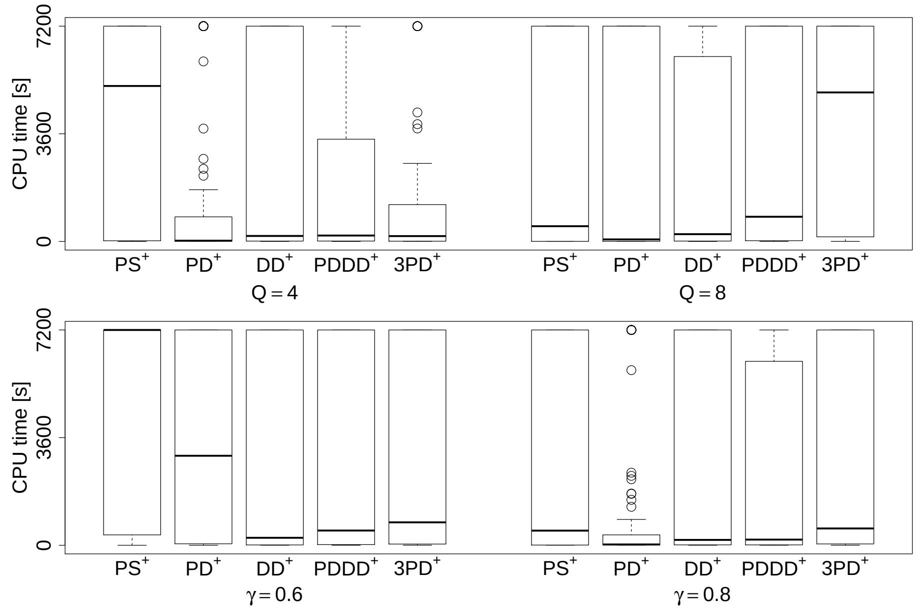

Figure 5: Boxplots comparing the distribution of solution times for instance set GLR1 for different models and different values of Q and γ, respectively.

For instance, with respect to the results on set GLR1, we observe that PD$^+$ outperforms the other models in many cases and thus makes it the overall winner. The position-dependent reformulation is a strong way of modeling the hop constraints without enlarging the model size too much, especially for tight Q values. However, since the path elimination constraints (18) are a quite weak way of ensuring the distance constraints, PD performs bad on instances with tight L values. We observe that 3PD$^+$ performs best when both the hop- and distance constraints are tight while DD$^+$ can be recommended for instances with tight distance but relatively loose hop constraints. PS$^+$ has the worst performance on set GLR1 while PDDD$^+$ does not perform too bad for tight instances on which it is nevertheless outperformed either by DD$^+$ or by 3PD$^+$.

These observations are also supported by Figure 5 which show runtime distributions for instances from GLR1 grouped by hop- or distance limits, respectively.

Similar observations can be made from the results on set GLR2, i.e., from Table 6. These results also confirm that instances for which the black nodes are placed close to the center (GLR2) are significantly harder to solve than those where the black nodes are well distributed (GLR1). Thus, the average CPU times as well as the remaining optimality gaps are higher than for set GLR1. Due to the larger number of unsolved cases we analyze the remaining optimality gaps in more detail. Figure 6 shows their distribution grouped by hop or distance limit, respectively. Surprisingly, it turns out that PDDD$^+$ which did not outperform the other models for a particular instance class seems to provide a quite good compromise. While it is (slightly) outperformed for all analyzed parameter settings, it does not seem to share any of the main weaknesses of PD$^+$ and 3PD$^+$ that are again the clear winners with respect to CPU times and solved instances in general as well as for tightly constrained instances.

While the distance-dependent reformulations achieved a relatively good performance for small values of γ and instance sets GLR1/2 the picture drastically changes when considering instance set MUT, see Tables 7 and 8, where distance values are integer values in $\{1, 2, ..., \lfloor 1000 \cdot \sqrt{2} \rfloor\}$. Clearly, the resulting huge models prohibit a successful application of distance-dependent reformulations. Nevertheless, for several MUT instances infeasibility could be shown by them quite early in the solution process. In

Table 5: Comparison of final optimality gaps, CPU times, and numbers of solved instances of our branch-and-cut algorithms based on different models for instance set GLR1. Bold values denote the best algorithms in a row. (“tl” . . . time limit reached, “-” . . . results not available)

				avg. optimality gaps in %					avg. CPU times in seconds					# instances solved (out of 5)				
$\lvert V \rvert$	Q	β	γ	PS$^+$	PD$^+$	DD$^+$	PDDD$^+$	3PD$^+$	PS$^+$	PD$^+$	DD$^+$	PDDD$^+$	3PD$^+$	PS$^+$	PD$^+$	DD$^+$	PDDD$^+$	3PD$^+$
20	4	1.00	0.6	1.4	**0.0**	**0.0**	**0.0**	**0.0**	2364	832	105	16	**4**	4	**5**	**5**	**5**	**5**
			0.8	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	135	**3**	64	8	5	**5**	**5**	**5**	**5**	**5**
			1.0	**0.0**	**0.0**	-	-	-	25	**1**	-	-	-	**5**	**5**	-	-	-
		1.33	0.6	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	1215	237	**4**	**4**	5	**5**	**5**	**5**	**5**	**5**
			0.8	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	3	**1**	2	2	3	**5**	**5**	**5**	**5**	**5**
			1.0	**0.0**	**0.0**	-	-	-	**0**	**0**	-	-	-	**5**	**5**	-	-	-
		1.67	0.6	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	10	3	**2**	**2**	3	**5**	**5**	**5**	**5**	**5**
			0.8	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	2	**1**	2	2	3	**5**	**5**	**5**	**5**	**5**
			1.0	**0.0**	**0.0**	-	-	-	**0**	**0**	-	-	-	**5**	**5**	-	-	-
	8	1.00	0.6	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	10	19	**6**	15	66	**5**	**5**	**5**	**5**	**5**
			0.8	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**2**	**2**	10	21	161	**5**	**5**	**5**	**5**	**5**
			1.0	**0.0**	**0.0**	-	-	-	**1**	**1**	-	-	-	**5**	**5**	-	-	-
		1.33	0.6	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	8	24	**2**	5	21	**5**	**5**	**5**	**5**	**5**
			0.8	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**2**	**2**	**2**	6	33	**5**	**5**	**5**	**5**	**5**
			1.0	**0.0**	**0.0**	-	-	-	**0**	**0**	-	-	-	**5**	**5**	-	-	-
		1.67	0.6	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	844	461	**2**	5	12	**5**	**5**	**5**	**5**	**5**
			0.8	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	2	**1**	**1**	2	8	**5**	**5**	**5**	**5**	**5**
			1.0	**0.0**	**0.0**	-	-	-	**0**	**0**	-	-	-	**5**	**5**	-	-	-
40	4	1.00	0.6	16.8	7.9	7.8	**0.0**	**0.0**	tl	5693	tl	3036	**193**	0	2	0	**5**	**5**
			0.8	9.7	**0.0**	5.0	**0.0**	**0.0**	tl	286	6774	1591	**282**	0	**5**	1	**5**	**5**
			1.0	8.0	**0.0**	-	-	-	tl	**8**	-	-	-	0	**5**	-	-	-
		1.33	0.6	5.2	**0.0**	**0.0**	**0.0**	**0.0**	5798	958	93	103	**90**	1	**5**	**5**	**5**	**5**
			0.8	0.9	**0.0**	**0.0**	**0.0**	**0.0**	2206	**15**	119	72	126	4	**5**	**5**	**5**	**5**
			1.0	**0.0**	**0.0**	-	-	-	449	**3**	-	-	-	**5**	**5**	-	-	-
		1.67	0.6	0.6	**0.0**	**0.0**	**0.0**	**0.0**	3554	121	**12**	23	34	4	**5**	**5**	**5**	**5**
			0.8	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	41	**6**	11	14	49	**5**	**5**	**5**	**5**	**5**
			1.0	**0.0**	**0.0**	-	-	-	6	**2**	-	-	-	**5**	**5**	-	-	-
	8	1.00	0.6	6.2	5.2	**0.0**	**0.0**	4.3	tl	6153	**1828**	2349	6772	0	1	**5**	**5**	1
			0.8	1.0	**0.0**	1.2	0.2	1.4	3923	**502**	3652	4148	4879	3	**5**	3	4	3
			1.0	**0.0**	**0.0**	-	-	-	2067	**23**	-	-	-	**5**	**5**	-	-	-
		1.33	0.6	1.7	2.1	**0.0**	**0.0**	**0.0**	4169	3322	**64**	297	1377	3	3	**5**	**5**	**5**
			0.8	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	829	155	**66**	183	902	**5**	**5**	**5**	**5**	**5**
			1.0	**0.0**	**0.0**	-	-	-	5	**3**	-	-	-	**5**	**5**	-	-	-
		1.67	0.6	0.6	0.7	**0.0**	**0.0**	**0.0**	3097	1828	**48**	348	857	4	4	**5**	**5**	**5**
			0.8	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	8	**7**	21	44	367	**5**	**5**	**5**	**5**	**5**
			1.0	**0.0**	**0.0**	-	-	-	**2**	**2**	-	-	-	**5**	**5**	-	-	-
60	4	1.00	0.6	24.9	10.4	13.6	3.5	**0.3**	tl	tl	tl	6154	**2104**	0	0	0	1	**4**
			0.8	17.3	**0.0**	13.4	1.00	**0.0**	tl	**860**	tl	4493	1639	0	**5**	0	3	**5**
			1.0	16.3	**0.0**	-	-	-	tl	**69**	-	-	-	0	**5**	-	-	-
		1.33	0.6	13.0	4.9	0.2	**0.0**	**0.0**	tl	5937	2292	2529	**791**	0	1	4	**5**	**5**
			0.8	3.4	**0.0**	0.7	**0.0**	**0.0**	6324	**120**	3213	833	796	2	**5**	3	**5**	**5**
			1.0	2.4	**0.0**	-	-	-	3949	**27**	-	-	-	3	**5**	-	-	-
		1.67	0.6	3.8	**0.0**	**0.0**	**0.0**	**0.0**	tl	487	**172**	237	326	0	**5**	**5**	**5**	**5**
			0.8	0.5	**0.0**	**0.0**	**0.0**	**0.0**	1580	**20**	359	124	295	4	**5**	**5**	**5**	**5**
			1.0	0.2	**0.0**	-	-	-	1455	**10**	-	-	-	4	**5**	-	-	-
	8	1.00	0.6	12.3	15.7	**6.3**	11.0	9.3	tl	tl	tl	tl	tl	**0**	**0**	**0**	**0**	**0**
			0.8	7.8	**1.3**	4.0	4.2	9.1	tl	**6931**	tl	tl	tl	0	**1**	0	0	0
			1.0	4.4	**0.0**	-	-	-	tl	**1000**	-	-	-	0	**5**	-	-	-
		1.33	0.6	5.6	3.9	**0.0**	1.6	4.1	6105	5147	**836**	3926	6497	1	2	**5**	3	1
			0.8	0.7	**0.0**	**0.0**	0.4	1.7	1772	**358**	924	3246	6022	4	**5**	**5**	4	2
			1.0	**0.0**	**0.0**	-	-	-	345	**25**	-	-	-	**5**	**5**	-	-	-
		1.67	0.6	2.7	1.8	**0.0**	0.2	1.9	6630	5320	**283**	2933	5019	1	2	**5**	4	2
			0.8	**0.0**	**0.0**	**0.0**	**0.0**	1.3	306	**58**	193	641	3742	**5**	**5**	**5**	**5**	3
			1.0	**0.0**	**0.0**	-	-	-	105	**18**	-	-	-	**5**	**5**	-	-	-
80	4	1.00	0.6	30.5	19.0	20.3	13.5	**5.2**	tl	tl	tl	tl	tl	**0**	**0**	**0**	**0**	**0**
			0.8	21.0	**2.4**	18.4	7.6	6.1	tl	**5868**	tl	tl	tl	0	**1**	0	0	0
			1.0	18.9	**0.2**	-	-	-	tl	**2271**	-	-	-	0	**4**	-	-	-
		1.33	0.6	13.0	6.0	1.9	0.8	**0.3**	tl	6056	4095	4953	**3531**	0	1	3	3	**4**
			0.8	5.2	**0.0**	2.0	0.2	0.3	5155	**522**	4186	3775	2497	2	**5**	3	4	4
			1.0	3.3	**0.0**	-	-	-	4507	**63**	-	-	-	2	**5**	-	-	-
		1.67	0.6	8.3	1.6	**0.0**	**0.0**	**0.0**	6675	3500	**463**	1526	1878	1	3	**5**	**5**	**5**
			0.8	1.1	**0.0**	**0.0**	**0.0**	**0.0**	3156	**78**	637	381	841	3	**5**	**5**	**5**	**5**
			1.0	0.5	**0.0**	-	-	-	1648	**24**	-	-	-	4	**5**	-	-	-
	8	1.00	0.6	19.4	19.8	**15.2**	18.1	22.5	tl	tl	tl	tl	tl	**0**	**0**	**0**	**0**	**0**
			0.8	15.5	**8.0**	12.9	14.4	13.2	tl	tl	tl	tl	tl	**0**	**0**	**0**	**0**	**0**
			1.0	11.2	**0.8**	-	-	-	tl	**6864**	-	-	-	0	**1**	-	-	-
		1.33	0.6	9.3	6.8	**1.1**	4.9	11.9	tl	tl	**6890**	tl	tl	0	0	**1**	0	0
			0.8	1.4	**0.0**	1.1	7.4	6.8	5886	**1035**	4790	tl	tl	1	**5**	2	0	0
			1.0	0.2	**0.0**	-	-	-	2514	**156**	-	-	-	4	**5**	-	-	-
		1.67	0.6	6.0	6.8	**0.0**	3.7	5.9	tl	tl	**2359**	tl	tl	0	0	**5**	0	0
			0.8	0.6	0.3	**0.0**	1.1	2.0	3861	**1656**	1682	4380	7105	3	4	**5**	3	1
			1.0	0.1	**0.0**	-	-	-	1681	**531**	-	-	-	4	**5**	-	-	-

Table 6: Comparison of final optimality gaps, CPU times, and numbers of solved instances of our branch-and-cut algorithms based on different models for instance set GLR2. Bold values denote the best algorithms in a row. ("tl" ... time limit reached, "-" ... results not available)

				avg. optimality gaps in %					avg. CPU times in seconds					# instances solved (out of 5)				
$\|V\|$	Q	β	γ	PS$^+$	PD$^+$	DD$^+$	PDDD$^+$	3PD$^+$	PS$^+$	PD$^+$	DD$^+$	PDDD$^+$	3PD$^+$	PS$^+$	PD$^+$	DD$^+$	PDDD$^+$	3PD$^+$
20	4	1.00	0.6	12.2	**0.0**	0.2	**0.0**	**0.0**	tl	179	1627	25	**8**	0	**5**	4	**5**	**5**
			0.8	6.6	**0.0**	1.8	**0.0**	**0.0**	4722	**3**	2938	19	19	3	**5**	3	**5**	**5**
			1.0	7.6	**0.0**	-	-	-	5832	**1**	-	-	-	1	**5**	-	-	-
		1.33	0.6	2.0	**0.0**	**0.0**	**0.0**	**0.0**	1961	5	**3**	**3**	**3**	4	**5**	**5**	**5**	**5**
			0.8	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	188	**1**	8	4	5	**5**	**5**	**5**	**5**	**5**
			1.0	**0.0**	**0.0**	-	-	-	296	**1**	-	-	-	**5**	**5**	-	-	-
		1.67	0.6	0.8	**0.0**	**0.0**	**0.0**	**0.0**	1500	6	**2**	3	3	4	**5**	**5**	**5**	**5**
			0.8	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	77	**1**	5	3	5	**5**	**5**	**5**	**5**	**5**
			1.0	**0.0**	**0.0**	-	-	-	58	**0**	-	-	-	**5**	**5**	-	-	-
	8	1.00	0.6	1.4	2.3	**0.0**	**0.0**	**0.0**	2968	1604	**22**	34	140	3	4	**5**	**5**	**5**
			0.8	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	519	**12**	191	60	181	**5**	**5**	**5**	**5**	**5**
			1.0	**0.0**	**0.0**	-	-	-	997	**3**	-	-	-	**5**	**5**	-	-	-
		1.33	0.6	**0.0**	0.4	**0.0**	**0.0**	**0.0**	1402	1753	**5**	15	34	**5**	4	**5**	**5**	**5**
			0.8	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	21	**5**	**5**	12	59	**5**	**5**	**5**	**5**	**5**
			1.0	**0.0**	**0.0**	-	-	-	3	**1**	-	-	-	**5**	**5**	-	-	-
		1.67	0.6	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	59	22	**3**	5	17	**5**	**5**	**5**	**5**	**5**
			0.8	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	2	**1**	2	4	16	**5**	**5**	**5**	**5**	**5**
			1.0	**0.0**	**0.0**	-	-	-	**1**	**1**	-	-	-	**5**	**5**	-	-	-
40	4	1.00	0.6	32.7	6.7	11.0	0.7	**0.0**	tl	6566	tl	3947	**724**	0	1	0	4	**5**
			0.8	30.7	**0.0**	13.7	**0.0**	**0.0**	tl	**102**	tl	1735	520	0	**5**	0	**5**	**5**
			1.0	28.6	**0.0**	-	-	-	tl	**33**	-	-	-	0	**5**	-	-	-
		1.33	0.6	27.9	4.7	1.3	**0.0**	**0.0**	tl	5840	3130	653	**263**	0	1	4	**5**	**5**
			0.8	23.1	**0.0**	5.0	0.2	0.1	tl	**465**	5885	1787	1525	0	**5**	1	4	4
			1.0	20.8	**0.0**	-	-	-	tl	**162**	-	-	-	0	**5**	-	-	-
		1.67	0.6	17.5	0.7	**0.0**	**0.0**	**0.0**	tl	2377	446	100	**84**	0	4	**5**	**5**	**5**
			0.8	12.5	**0.0**	2.7	**0.0**	**0.0**	tl	**43**	2923	240	184	0	**5**	3	**5**	**5**
			1.0	10.1	**0.0**	-	-	-	5901	**10**	-	-	-	1	**5**	-	-	-
	8	1.00	0.6	15.7	7.7	**0.7**	0.9	2.2	tl	tl	**2624**	4408	tl	0	0	**4**	**4**	0
			0.8	14.5	**0.0**	3.5	1.3	4.1	tl	**1505**	6084	5356	tl	0	**5**	1	2	0
			1.0	11.9	**0.0**	-	-	-	tl	**195**	-	-	-	0	**5**	-	-	-
		1.33	0.6	17.6	9.2	**0.0**	**0.0**	0.2	tl	tl	**242**	1865	4529	0	0	**5**	**5**	4
			0.8	10.7	0.6	0.6	**0.2**	4.4	tl	3455	**3380**	3910	6562	0	3	3	**4**	2
			1.0	8.0	**0.0**	-	-	-	tl	**63**	-	-	-	0	**5**	-	-	-
		1.67	0.6	10.2	3.3	**0.0**	**0.0**	**0.0**	6276	5765	**97**	607	2150	1	1	**5**	**5**	**5**
			0.8	7.0	**0.0**	0.7	0.5	4.1	5773	2539	**2135**	2196	5309	1	**5**	4	4	2
			1.0	3.9	**0.0**	-	-	-	5062	**206**	-	-	-	2	**5**	-	-	-
60	4	1.00	0.6	44.5	7.0	20.1	3.0	**0.2**	tl	tl	tl	tl	**3898**	0	0	0	0	**4**
			0.8	42.9	0.4	25.6	0.8	**0.3**	tl	**1582**	tl	4998	4492	0	**4**	0	3	3
			1.0	42.5	**0.0**	-	-	-	tl	**206**	-	-	-	0	**5**	-	-	-
		1.33	0.6	37.8	4.7	14.7	2.5	**0.4**	tl	tl	tl	5877	**5178**	0	0	0	1	**2**
			0.8	34.2	**0.0**	18.4	0.6	0.2	tl	**1060**	tl	tl	4914	0	**5**	0	0	3
			1.0	33.3	**0.0**	-	-	-	tl	**219**	-	-	-	0	**5**	-	-	-
		1.67	0.6	29.5	4.5	5.1	0.5	**0.0**	tl	tl	tl	3896	**1623**	0	0	0	3	**5**
			0.8	25.9	**0.0**	11.9	0.9	0.4	tl	**1360**	tl	5380	4421	0	**5**	0	3	3
			1.0	24.3	**0.0**	-	-	-	tl	**233**	-	-	-	0	**5**	-	-	-
	8	1.00	0.6	27.0	9.2	8.8	**6.6**	22.3	tl	tl	tl	tl	tl	**0**	**0**	**0**	**0**	**0**
			0.8	24.7	**1.3**	11.8	4.5	20.4	tl	**5852**	tl	tl	tl	0	**1**	0	0	0
			1.0	24.0	**0.1**	-	-	-	tl	**2677**	-	-	-	0	**4**	-	-	-
		1.33	0.6	25.9	10.3	**4.6**	6.2	11.9	tl	tl	**6049**	tl	tl	0	0	**1**	0	0
			0.8	19.6	**3.1**	8.6	5.3	17.9	tl	tl	tl	tl	tl	**0**	**0**	**0**	**0**	**0**
			1.0	18.4	**0.6**	-	-	-	tl	**5354**	-	-	-	0	**3**	-	-	-
		1.67	0.6	24.8	12.9	**2.4**	5.8	9.0	tl	tl	**6890**	tl	tl	0	0	**2**	0	0
			0.8	18.4	**3.0**	7.0	5.1	11.6	tl	**5600**	tl	tl	tl	0	**2**	0	0	0
			1.0	15.7	**0.3**	-	-	-	tl	**2129**	-	-	-	0	**4**	-	-	-
80	4	1.00	0.6	48.8	7.1	26.3	7.0	**4.4**	tl	tl	tl	tl	**6733**	0	0	0	0	**1**
			0.8	48.2	**1.4**	31.0	4.7	3.2	tl	tl	tl	tl	tl	**0**	**0**	**0**	**0**	**0**
			1.0	47.6	**0.3**	-	-	-	tl	**3375**	-	-	-	0	**3**	-	-	-
		1.33	0.6	41.5	5.2	18.1	5.0	**2.1**	tl	tl	tl	tl	tl	**0**	**0**	**0**	**0**	**0**
			0.8	38.8	**0.8**	24.2	4.0	2.2	tl	**4137**	tl	tl	tl	0	**4**	0	0	0
			1.0	38.0	**0.3**	-	-	-	tl	**2456**	-	-	-	0	**4**	-	-	-
		1.67	0.6	34.6	6.2	13.1	4.4	**2.1**	tl	tl	tl	tl	tl	**0**	**0**	**0**	**0**	**0**
			0.8	32.1	**1.2**	19.4	4.0	2.2	tl	**5907**	tl	tl	tl	0	**1**	0	0	0
			1.0	31.2	**0.1**	-	-	-	tl	**2878**	-	-	-	0	**4**	-	-	-
	8	1.00	0.6	32.4	14.3	14.5	**10.2**	35.3	tl	tl	tl	tl	tl	**0**	**0**	**0**	**0**	**0**
			0.8	32.4	**3.2**	16.2	7.9	32.7	tl	tl	tl	tl	tl	**0**	**0**	**0**	**0**	**0**
			1.0	30.8	**1.0**	-	-	-	tl	**5957**	-	-	-	0	**2**	-	-	-
		1.33	0.6	32.3	11.2	11.5	**9.3**	32.3	tl	tl	tl	tl	tl	**0**	**0**	**0**	**0**	**0**
			0.8	28.0	**3.8**	17.1	9.1	29.3	tl	tl	tl	tl	tl	**0**	**0**	**0**	**0**	**0**
			1.0	26.5	**2.5**	-	-	-	tl	tl	-	-	-	**0**	**0**	-	-	-
		1.67	0.6	26.1	12.0	**7.9**	9.9	24.7	tl	tl	tl	tl	tl	**0**	**0**	**0**	**0**	**0**
			0.8	23.0	**4.2**	13.7	9.6	23.5	tl	tl	tl	tl	tl	**0**	**0**	**0**	**0**	**0**
			1.0	22.0	**2.5**	-	-	-	tl	tl	-	-	-	**0**	**0**	-	-	-

Table 7: Comparison of final optimality gaps, CPU times, and numbers of solved and infeasible instances of our branch-and-cut algorithms based on different models for instances with $|V| \in \{20, 40\}$ from set MUT. Bold values denote the best algorithms in a row. (“tl” . . . time limit reached, “-” . . . results not available)

				avg. optimality gaps in %					avg. CPU times in seconds					# instances solved (inf.) (out of 5)				
$\lvert V\rvert$	α	β	γ	PS$^+$	PD$^+$	DD$^+$	PDDD$^+$	3PD$^+$	PS$^+$	PD$^+$	DD$^+$	PDDD$^+$	3PD$^+$	PS$^+$	PD$^+$	DD$^+$	PDDD$^+$	3PD$^+$
20	0.2	1.00	0.7	-	-	-	-	-	45	**0**	2	**0**	**0**	**5(5)**	**5(5)**	**5(5)**	**5(5)**	**5(5)**
			0.8	16.7	**0.0**	1.5	**0.0**	**0.0**	1462	**7**	1483	66	14	4(3)	**5(3)**	4(3)	**5(3)**	**5(3)**
			1.0	8.3	**0.0**	-	-	-	3101	**1**	-	-	-	3(0)	**5(0)**	-	-	-
		1.33	0.7	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	107	**1**	80	11	25	**5(4)**	**5(4)**	**5(4)**	**5(4)**	**5(4)**
			0.8	**0.0**	**0.0**	3.2	**0.0**	**0.0**	350	**2**	1440	93	58	**5(3)**	**5(3)**	4(3)	**5(3)**	**5(3)**
			1.0	**0.0**	**0.0**	-	-	-	31	**1**	-	-	-	**5(0)**	**5(0)**	-	-	-
		1.67	0.7	-	-	-	-	-	**0**	**0**	**0**	**0**	**0**	**5(5)**	**5(5)**	**5(5)**	**5(5)**	**5(5)**
			0.8	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	17	1	**0**	**0**	**0**	**5(4)**	**5(4)**	**5(4)**	**5(4)**	**5(4)**
			1.0	**0.0**	**0.0**	-	-	-	1	**0**	-	-	-	**5(0)**	**5(0)**	-	-	-
	0.3	1.00	0.7	-	-	-	-	-	**0**	**0**	**0**	**0**	**0**	**5(5)**	**5(5)**	**5(5)**	**5(5)**	**5(5)**
			0.8	-	-	-	-	-	11	19	27	**9**	673	**5(5)**	**5(5)**	**5(5)**	**5(5)**	**5(5)**
			1.0	**0.0**	**0.0**	-	-	-	7	**1**	-	-	-	**5(0)**	**5(0)**	-	-	-
		1.33	0.7	-	-	-	-	-	**9**	165	10	28	18	**5(5)**	**5(5)**	**5(5)**	**5(5)**	**5(5)**
			0.8	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**5**	44	6	**5**	19	**5(4)**	**5(4)**	**5(4)**	**5(4)**	**5(4)**
			1.0	**0.0**	**0.0**	-	-	-	**0**	**0**	-	-	-	**5(0)**	**5(0)**	-	-	-
		1.67	0.7	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0**	**0**	1	1	4	**5(4)**	**5(4)**	**5(4)**	**5(4)**	**5(4)**
			0.8	7.4	**0.0**	**0.0**	**0.0**	**0.0**	1440	386	**2**	20	14	4(2)	**5(3)**	**5(3)**	**5(3)**	**5(3)**
			1.0	**0.0**	**0.0**	-	-	-	**0**	**0**	-	-	-	**5(0)**	**5(0)**	-	-	-
	0.4	1.00	0.7	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0**	4	67	47	113	**5(4)**	**5(4)**	**5(4)**	**5(4)**	**5(4)**
			0.8	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**2**	49	33	39	158	**5(3)**	**5(3)**	**5(3)**	**5(3)**	**5(3)**
			1.0	**0.0**	**0.0**	-	-	-	**0**	**0**	-	-	-	**5(0)**	**5(0)**	-	-	-
		1.33	0.7	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	5	25	7	20	21	**5(4)**	**5(4)**	**5(4)**	**5(4)**	**5(4)**
			0.8	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	3	61	**2**	8	34	**5(4)**	**5(4)**	**5(4)**	**5(4)**	**5(4)**
			1.0	**0.0**	**0.0**	-	-	-	**0**	**0**	-	-	-	**5(0)**	**5(0)**	-	-	-
		1.67	0.7	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0**	**0**	1	1	3	**5(4)**	**5(4)**	**5(4)**	**5(4)**	**5(4)**
			0.8	8.1	7.1	**0.0**	**0.0**	**0.0**	1440	1440	**3**	8	61	4(2)	4(2)	**5(3)**	**5(3)**	**5(3)**
			1.0	**0.0**	**0.0**	-	-	-	**0**	**0**	-	-	-	**5(0)**	**5(0)**	-	-	-
40	0.2	1.00	0.7	15.5	**7.7**	14.7	10.5	28.7	4321	4372	3951	**2890**	2973	2(2)	2(2)	**3(3)**	**3(3)**	**3(3)**
			0.8	16.4	**12.5**	24.5	21.6	31.3	tl	tl	tl	tl	tl	**0(0)**	**0(0)**	**0(0)**	**0(0)**	**0(0)**
			1.0	5.5	**0.0**	-	-	-	4396	**71**	-	-	-	2(0)	**5(0)**	-	-	-
		1.33	0.7	15.5	17.4	**3.9**	11.9	14.7	3067	4323	**2488**	2921	2939	3(2)	2(1)	**4(4)**	3(3)	3(3)
			0.8	23.3	22.4	**14.8**	18.9	22.5	5855	**5777**	7180	tl	tl	**1(0)**	**1(0)**	**1(1)**	0(0)	0(0)
			1.0	1.9	**0.0**	-	-	-	1453	**12**	-	-	-	4(0)	**5(0)**	-	-	-
		1.67	0.7	3.2	1.2	**0.0**	0.4	**0.0**	1443	1442	1340	1473	**922**	4(1)	4(1)	**5(1)**	4(1)	**5(1)**
			0.8	**0.0**	8.2	0.2	0.7	5.7	**260**	1444	2818	3868	4339	**5(1)**	4(0)	4(1)	3(1)	2(1)
			1.0	**0.0**	**0.0**	-	-	-	**1**	2	-	-	-	**5(0)**	**5(0)**	-	-	-
	0.3	1.00	0.7	18.5	13.0	**11.2**	13.0	19.2	**2882**	3126	2910	2904	4320	**3(3)**	**3(3)**	**3(3)**	**3(3)**	2(2)
			0.8	14.4	**11.8**	14.0	24.2	-	tl	**5783**	tl	tl	tl	0(0)	**1(0)**	0(0)	0(0)	0(0)
			1.0	1.9	**0.0**	-	-	-	3143	**83**	-	-	-	3(0)	**5(0)**	-	-	-
		1.33	0.7	16.9	20.0	**0.0**	**0.0**	17.9	2082	2880	**354**	656	2880	4(3)	3(3)	**5(4)**	**5(4)**	3(3)
			0.8	**4.6**	6.9	7.9	6.0	12.8	4131	**3673**	3715	4321	4321	**3(2)**	**3(2)**	**3(3)**	2(2)	2(2)
			1.0	**0.0**	**0.0**	-	-	-	112	**4**	-	-	-	**5(0)**	**5(0)**	-	-	-
		1.67	0.7	19.8	21.4	**16.9**	20.0	37.5	4397	**4329**	4332	4357	5760	**2(1)**	**2(1)**	**2(2)**	**2(2)**	1(1)
			0.8	12.8	13.3	**3.7**	5.4	11.9	**2791**	3333	4401	4476	5768	**4(1)**	3(0)	2(2)	2(2)	1(1)
			1.0	**0.0**	**0.0**	-	-	-	6	**3**	-	-	-	**5(0)**	**5(0)**	-	-	-
	0.4	1.00	0.7	**1.0**	5.9	9.0	10.1	-	**2159**	4794	5858	5843	tl	**4(1)**	2(0)	1(1)	1(1)	0(0)
			0.8	**3.9**	5.3	15.3	15.2	-	**3664**	4502	tl	tl	tl	**3(0)**	2(0)	0(0)	0(0)	0(0)
			1.0	2.2	**0.0**	-	-	-	1935	**358**	-	-	-	4(0)	**5(0)**	-	-	-
		1.33	0.7	**11.9**	14.8	16.0	17.7	31.6	**5760**	**5760**	**5760**	**5760**	**5760**	**1(1)**	**1(1)**	**1(1)**	**1(1)**	**1(1)**
			0.8	**2.4**	3.5	8.5	7.6	3.3	**3002**	3664	4576	4697	tl	**3(1)**	**3(1)**	2(1)	2(1)	0(0)
			1.0	**0.0**	**0.0**	-	-	-	63	**12**	-	-	-	**5(0)**	**5(0)**	-	-	-
		1.67	0.7	**9.3**	14.4	11.9	12.2	40.1	**2531**	2920	4338	4359	5760	**4(2)**	3(1)	2(2)	2(2)	1(1)
			0.8	10.9	10.7	2.6	**2.1**	10.4	**1504**	2346	3025	4571	5769	**4(1)**	**4(1)**	3(2)	2(2)	1(1)
			1.0	**0.0**	**0.0**	-	-	-	**1**	**1**	-	-	-	**5(0)**	**5(0)**	-	-	-

Table 8: Comparison of final optimality gaps, CPU times, and numbers of solved and infeasible instances of our branch-and-cut algorithms based on different models for instances with $|V| \in \{60, 80\}$ from set MUT. Bold values denote the best algorithms in a row. (“tl” …time limit reached, “-” …results not available)

				avg. optimality gaps in %					avg. CPU times in seconds					# instances solved (inf.) (out of 5)				
$\lvert V\rvert$	α	β	γ	PS$^+$	PD$^+$	DD$^+$	PDDD$^+$	3PD$^+$	PS$^+$	PD$^+$	DD$^+$	PDDD$^+$	3PD$^+$	PS$^+$	PD$^+$	DD$^+$	PDDD$^+$	3PD$^+$
60	0.2	1.00	0.7	22.7	**15.9**	29.7	35.7	-	5760	5760	5760	5760	**4339**	1(1)	1(1)	1(1)	1(1)	**2(2)**
			0.8	18.5	**11.3**	34.4	34.4	-	tl	tl	tl	tl	tl	**0(0)**	**0(0)**	**0(0)**	**0(0)**	**0(0)**
			1.0	6.9	**0.5**	-	-	-	6517	**2060**	-	-	-	1(0)	**4(0)**	-	-	-
		1.33	0.7	13.3	12.2	**10.7**	16.3	-	5770	**5266**	5792	5840	tl	1(0)	**2(0)**	1(1)	1(1)	0(0)
			0.8	4.2	**2.0**	10.8	12.1	-	5782	**5691**	tl	tl	tl	1(0)	**2(0)**	0(0)	0(0)	0(0)
			1.0	3.3	**0.0**	-	-	-	5772	**1647**	-	-	-	1(0)	**5(0)**	-	-	-
		1.67	0.7	**3.8**	5.0	15.5	16.7	-	**5013**	5760	5760	5760	5760	**2(1)**	1(1)	1(1)	1(1)	1(1)
			0.8	**0.4**	0.8	16.9	17.3	-	**1906**	3096	tl	tl	tl	**4(0)**	3(0)	0(0)	0(0)	0(0)
			1.0	**0.0**	**0.0**	-	-	-	647	**316**	-	-	-	**5(0)**	**5(0)**	-	-	-
	0.3	1.00	0.7	**7.5**	9.0	28.4	28.4	-	**4320**	**4320**	**4320**	**4320**	**4320**	**2(2)**	**2(2)**	**2(2)**	**2(2)**	**2(2)**
			0.8	**6.2**	7.6	12.8	14.8	-	**5805**	tl	tl	tl	tl	**1(0)**	0(0)	0(0)	0(0)	0(0)
			1.0	**0.0**	**0.0**	-	-	-	1162	**199**	-	-	-	**5(0)**	**5(0)**	-	-	-
		1.33	0.7	**12.6**	14.8	30.9	33.8	-	tl	tl	tl	tl	tl	**0(0)**	**0(0)**	**0(0)**	**0(0)**	**0(0)**
			0.8	**2.1**	4.7	20.1	20.8	-	**4191**	5822	tl	tl	tl	**3(0)**	1(0)	0(0)	0(0)	0(0)
			1.0	**0.0**	**0.0**	-	-	-	51	**41**	-	-	-	**5(0)**	**5(0)**	-	-	-
		1.67	0.7	13.4	14.7	22.7	23.0	10.8	**4364**	4712	5766	5771	tl	**2(1)**	**2(1)**	1(1)	1(1)	0(0)
			0.8	**5.3**	7.6	19.7	17.8	-	**5761**	5765	7196	tl	tl	**1(0)**	**1(0)**	**1(1)**	0(0)	0(0)
			1.0	**0.0**	**0.0**	-	-	-	758	**415**	-	-	-	**5(0)**	**5(0)**	-	-	-
	0.4	1.00	0.7	13.8	**10.5**	32.5	32.5	-	4325	**4324**	5760	5760	5760	**2(1)**	**2(1)**	1(1)	1(1)	1(1)
			0.8	11.0	**7.2**	18.6	18.6	-	5762	**5761**	tl	tl	tl	**1(0)**	**1(0)**	0(0)	0(0)	0(0)
			1.0	6.2	**2.2**	-	-	-	5760	**4638**	-	-	-	1(0)	**2(0)**	-	-	-
		1.33	0.7	**13.2**	14.7	32.3	32.3	-	**2880**	**2880**	**2880**	**2880**	**2880**	**3(3)**	**3(3)**	**3(3)**	**3(3)**	**3(3)**
			0.8	**11.2**	13.7	37.4	37.4	-	tl	tl	tl	tl	tl	**0(0)**	**0(0)**	**0(0)**	**0(0)**	**0(0)**
			1.0	**0.0**	**0.0**	-	-	-	337	**134**	-	-	-	**5(0)**	**5(0)**	-	-	-
		1.67	0.7	**5.4**	7.0	21.1	19.8	-	**4342**	5463	5760	5760	5760	**2(1)**	**2(1)**	1(1)	1(1)	1(1)
			0.8	**3.3**	4.4	14.4	15.6	-	**3637**	5461	5867	6593	tl	**3(1)**	2(0)	1(1)	1(1)	0(0)
			1.0	**0.4**	0.5	-	-	-	**1445**	1460	-	-	-	**4(0)**	**4(0)**	-	-	-
80	0.2	1.00	0.7	24.4	**21.6**	30.0	30.0	-	5760	5760	5760	5760	**4374**	1(1)	1(1)	1(1)	1(1)	**2(2)**
			0.8	20.4	**18.7**	-	-	-	tl	tl	tl	tl	tl	**0(0)**	**0(0)**	**0(0)**	**0(0)**	**0(0)**
			1.0	7.8	**3.1**	-	-	-	tl	tl	-	-	-	**0(0)**	**0(0)**	-	-	-
		1.33	0.7	15.6	**13.9**	39.8	39.9	-	**5958**	tl	tl	tl	tl	**1(0)**	0(0)	0(0)	0(0)	0(0)
			0.8	8.2	**6.5**	40.9	40.9	-	**5816**	6883	tl	tl	tl	**1(0)**	**1(0)**	0(0)	0(0)	0(0)
			1.0	3.3	**0.6**	-	-	-	4355	**2035**	-	-	-	2(0)	**4(0)**	-	-	-
		1.67	0.7	9.2	**7.5**	21.2	19.1	-	**6081**	tl	tl	tl	tl	**1(0)**	0(0)	0(0)	0(0)	0(0)
			0.8	5.2	**3.3**	21.8	22.9	-	7112	**6347**	tl	tl	tl	**1(0)**	**1(0)**	0(0)	0(0)	0(0)
			1.0	2.8	**1.0**	-	-	-	3623	**3233**	-	-	-	**3(0)**	**3(0)**	-	-	-
	0.3	1.00	0.7	**10.9**	13.1	-	-	-	tl	tl	tl	tl	tl	**0(0)**	**0(0)**	**0(0)**	**0(0)**	**0(0)**
			0.8	**8.8**	9.2	-	-	-	**7165**	tl	tl	tl	tl	**1(0)**	0(0)	0(0)	0(0)	0(0)
			1.0	0.7	**0.5**	-	-	-	**3274**	4485	-	-	-	**4(0)**	3(0)	-	-	-
		1.33	0.7	**7.1**	7.6	28.7	29.6	-	**5596**	5760	5760	5760	5760	**2(1)**	1(1)	1(1)	1(1)	1(1)
			0.8	4.1	6.3	**4.0**	11.8	-	**5884**	6671	tl	tl	tl	**1(0)**	**1(0)**	0(0)	0(0)	0(0)
			1.0	1.3	**1.0**	-	-	-	**1524**	1871	-	-	-	**4(0)**	**4(0)**	-	-	-
		1.67	0.7	**16.8**	17.0	31.4	37.6	-	**5792**	6496	tl	tl	tl	**1(0)**	**1(0)**	0(0)	0(0)	0(0)
			0.8	**5.8**	7.1	35.9	35.9	-	**5770**	5787	tl	tl	tl	**1(0)**	**1(0)**	0(0)	0(0)	0(0)
			1.0	1.0	**0.8**	-	-	-	2899	**1665**	-	-	-	3(0)	**4(0)**	-	-	-
	0.4	1.00	0.7	**6.0**	8.4	-	-	-	**5401**	5760	5760	5760	5760	**2(2)**	1(1)	1(1)	1(1)	1(1)
			0.8	**6.3**	10.7	-	-	-	tl	tl	tl	tl	tl	**0(0)**	**0(0)**	**0(0)**	**0(0)**	**0(0)**
			1.0	1.5	**1.3**	-	-	-	**2939**	2973	-	-	-	**3(0)**	**3(0)**	-	-	-
		1.33	0.7	**11.4**	12.0	17.5	17.5	-	tl	tl	tl	tl	tl	**0(0)**	**0(0)**	**0(0)**	**0(0)**	**0(0)**
			0.8	**8.3**	9.1	-	-	-	tl	tl	tl	tl	tl	**0(0)**	**0(0)**	**0(0)**	**0(0)**	**0(0)**
			1.0	**0.5**	0.7	-	-	-	**3069**	4409	-	-	-	**4(0)**	2(0)	-	-	-
		1.67	0.7	**6.6**	9.7	-	-	-	**5907**	tl	tl	tl	tl	**1(0)**	0(0)	0(0)	0(0)	0(0)
			0.8	**3.0**	5.4	-	-	-	**4739**	7192	tl	tl	tl	**2(0)**	1(0)	0(0)	0(0)	0(0)
			1.0	**0.8**	**0.8**	-	-	-	**1641**	1759	-	-	-	**4(0)**	**4(0)**	-	-	-

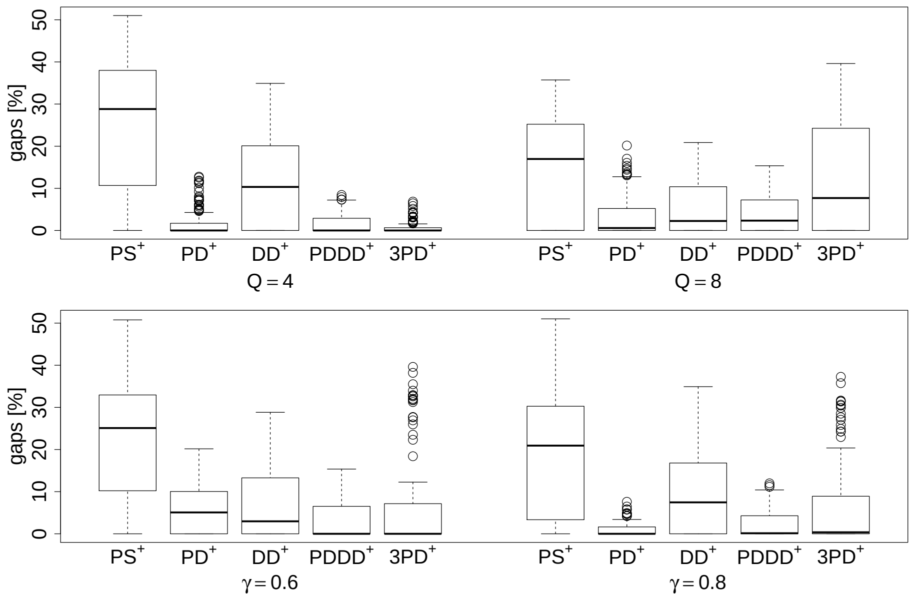

Figure 6: Boxplots comparing the distribution of final optimality gaps for instance set GLR2 for different models and different values of Q and γ, respectively.

general, however, these instances have a rather different characteristic in which tight modeling of either the hop- or distance constraints does not seem to be too relevant. This can be observed from the fact that the simplest model PS$^+$ often outperforms the others, in particular on larger instances. PD$^+$ is, however, a good alternative as well.

Overall, the following findings and conclusions can be made: PS$^+$ should be preferred for loosely constrained instances such as the large MUT instances whose results are given in Table 8. PD$^+$ performs well throughout all the instance sets except for extremely tight distance constraints. It is precisely for these cases that PDDD$^+$ and 3PD$^+$ obtain good results. The reason is that these formulations model the extra requirements by the additional indices and there is no need for detecting infeasible paths. DD$^+$ outperforms the other variants only for cases with loose hop constraints and tight distance constraints which can be easily explained by the fact that DD$^+$ is a good model for the distance constraint but includes only weak path elimination inequalities to ensure the hop constraints. To summarize, most of the models are specialized to work well for particular instance characteristics leading to a set of solution approaches which yield promising results for all instances—if applied appropriately.

8 Conclusions

In this paper we have studied ILP models and developed branch-and-cut algorithms to solve the BWTSP. The proposed formulations are based on a strategy that is positioned in between two strategies proposed in the literature. Following this strategy, several variants of position- and distance-dependent reformulations together with corresponding layered graph representations have been proposed, developed and studied. The results from of our extensive computational study show that the proposed models work well for particular and different instance characteristics. Overall, if applied appropriately, the models proposed in this paper can be used to successfully tackle most instance characteristics.

Our results confirm that modeling ideas that yield theoretically strong, but very large-size formulations such as the side-by-side approach of the model PDDD or the three-dimensional approach of model 3PD pay off for tightly constrained instances when compared to standard resource-indexed reformulations. We also gave some empirical evidence that “on the fly” reformulation ideas such as the ones of model PDPS in which the basic models are reformulated with additional problem-dependent information (in this case, information about the different path segments) does not pay off if the strength improvement is too small (in practice). Finally, our results indicate a strong need for a better understanding how to efficiently separate additional layered graph inequalities without increasing the model size too much.

Acknowledgements

This work is supported by National Funding from FCT - Fundação para a Ciência e a Tecnologia, under the project UID/MAT/04561/2013 and by the Vienna Science and Technology Fund (WWTF) through project ICT15-014. Part of this research has been performed while M. Leitner was a research fellow at the Department of Computer Science, Université Libre de Bruxelles (Brussels, Belgium) where he was supported by the Interuniversity Attraction Poles Programme initiated by the Belgian Science Policy Office. These supports are greatly acknowledged.

References

[1] H. Abeledo, R. Fukasawa, A. Pessoa, and E. Uchoa. The time dependent traveling salesman problem: polyhedra and algorithm. *Mathematical Programming Computation*, 5(1):27–55, 2013.

[2] D. L. Applegate, R. E. Bixby, V. Chvátal, and W. J. Cook. *The Traveling Salesman Problem: A Computational Study*. Princeton University Press, 2007.

[3] N. Ascheuer, M. Fischetti, and M. Grötschel. Solving the asymmetric travelling salesman problem with time windows by branch-and-cut. *Mathematical Programming*, 90(3):475–506, 2001. ISSN 0025-5610. doi: 10.1007/PL00011432.

[4] Q. Botton, B. Fortz, L. Gouveia, and M. Poss. Benders decomposition for the hop-constrained survivable network design problem. *INFORMS Journal on Computing*, 25(1):13–26, 2013.

[5] M. Bourgeois, G. Laporte, and F. Semet. Heuristics for the black and white traveling salesman problem. *Computers & Operations Research*, 30:75–85, 2003.

[6] B. V. Cherkassky and A. V. Goldberg. On implementing push-relabel method for the maximum flow problem. *Algorithmica*, 19:390–410, 1994.

[7] G. Ghiani, G. Laporte, and F. Semat. The black and white traveling salesman problem. *Operations Research*, 54(2):366–378, 2006.

[8] M. T. Godinho, L. Gouveia, and P. Pesneau. Natural and extended formulations for the time-dependent traveling salesman problem. *Discrete Applied Mathematics*, 164(1):138–153, 2014. ISSN 0166218X.

[9] L. Gouveia. A 2n constraint formulation for the capacitated minimal spanning tree problem. *Operations Research*, 43(1):130–141, 1995.

[10] L. Gouveia and M. Leitner. Design of survivable networks with vulnerability constraints. *European Journal of Operational Research*, 258(1):89–103, 2017.

[11] L. Gouveia and M. Ruthmair. Load-dependent and precedence-based models for pickup and delivery problems. *Computers & Operations Research*, 63:56–71, 2015. doi: 10.1016/j.cor.2015.04.008.

[12] L. Gouveia and S. Voß. A classification of formulations for the (time-dependent) traveling salesman problem. *European Journal of Operational Research*, 83(1):69–82, 1995.

[13] L. Gouveia, A. Paias, and D. Sharma. Modeling and solving the rooted distance-constrained minimum spanning tree problem. *Computers & Operations Research*, 35(2):600–613, 2008. ISSN 03050548. doi: 10.1016/j.cor.2006.03.022.

[14] L. Gouveia, L. G. Simonetti, and E. Uchoa. Modeling hop-constrained and diameter-constrained minimum spanning tree problems as steiner tree problems over layered graphs. *Mathematical Programming*, 128(1):123–148, 2011. ISSN 0025-5610.

[15] L. Gouveia, M. Leitner, and I. Ljubić. Hop constrained steiner trees with multiple root nodes. *European Journal of Operational Research*, 236(1):100–112, 2014.

[16] L. Gouveia, M. Leitner, and I. Ljubić. The two-level diameter constrained spanning tree problem. *Mathematical Programming*, 150(1):49–78, 2015.

[17] G. Gutin and A. P. Punen, editors. *The Traveling Salesman Problem and Its Variations*, volume 12 of *Combinatorial Optimization.* Springer US, 2007.

[18] P. Hansen and N. Mladenović. Variable neighborhood search: Principles and applications. *European Journal of Operational Research*, 130(3):449–467, 2001.

[19] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys. *The traveling salesman problem: a guided tour of combinatorial optimization.* Wiley New York, 1985.

[20] M. Leitner. Layered graph models and exact algorithms for the generalized hop-constrained minimum spanning tree problem. *Computers & Operations Research*, 65:1–18, 2016.

[21] V. Mak and N. Boland. Heuristic approaches ot the asymmetric travelling salesman problem with replenishment arcs. *International Transactions in Operational Research*, 7:431–447, 2000.

[22] I. Muter. A new formulation and approach for the black and white traveling salesman problem. *Computers & Operations Research*, 53:96–106, 2015.

[23] M. Padberg and G. Rinaldi. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM review*, 33(1):60–100, 1991.

[24] J. C. Picard and M. Queyranne. The time-dependent traveling salesman problem and its application to the tardiness problem in one-machine scheduling. *Operations Research*, 26(1):86–110, 1978.

[25] M. Ruthmair and G. R. Raidl. A layered graph model and an adaptive layers framework to solve delay-constrained minimum tree problems. In O. Günlük and G. Woeginger, editors, *Proceedings of the 15th Conference on Integer Programming and Combinatorial Optimization (IPCO XV)*, volume 6655 of *LNCS*, pages 376–388. Springer, 2011. doi: 10.1007/978-3-642-20807-2_30.

[26] M. Ruthmair and G. R. Raidl. On solving the rooted delay- and delay-variation-constrained steiner tree problem. In A. Mahjoub et al., editors, *Proceedings of the 2nd International Symposium on Combinatorial Optimization*, volume 7422 of *LNCS*, pages 225–236. Springer, 2012. doi: 10.1007/978-3-642-32147-4_21.

[27] K. T. Talluri. The four-day aircraft maintenance routing problem. *Transportation Science*, 32(1): 43–53, 1998.

[28] E. Uchoa. Cuts over extended formulations by flow discretization. In A. R. Mahjoub, editor, *Progress in Combinatorial Optimization*, pages 255–282. ISTE-Wiley, 2011.

[29] O. J. Wasem. An algorithm for designing rings for survivable fiber networks. *IEEE Transactions on Reliability*, 40(4):428–432, 1991.

A Computational Comparison to Ghiani et al. [7]

We first note that despite several attempts, we have not been able to obtain the instances from Ghiani et al. [7]. To allow, at least, a partial comparison, we generated new random instances following their description and used identical parameters in the associated computational experiments. We set the memory limit to 5 GB and the time limit to 10000 seconds.

Tables 9–11 report numbers of solved instances and average CPU times for our algorithms on these instances. We do not report these values for the branch-and-cut algorithm introduced by Ghiani et al. [7] as such a direct comparison would be misleading for three main reasons:

- Even though the instances have been generated with the same parameters, their difficulty may be significantly different from those used in Ghiani et al. [7]. One main reason is the (randomly chosen) distribution of black nodes (as also demonstrated with our instance sets GLR1 and GLR2 in Section 7.1). To this end, we note that we observed runtimes that differ by one to two orders of magnitude for two instances with the same numbers of nodes and identical values for α and β.

- Ghiani et al. [7] only report average solution times for successfully solved instances. Results from test runs reaching memory or time limit are ignored in their paper.

- Significantly different hardware and software versions have been used.

Thus, a direct comparison between the results in [7] and the ones in Tables 9–11 cannot lead to statistically significant conclusions. We observe, however, that their testbed only contains instances where costs and distances are correlated (identical values) and rather loose hop constraints (high values for α) are imposed. Thus, small sized (often weak) models with few variables and few (initial) constraints are naturally favorable here. This is confirmed by the fact that the branch-and-cut approach by Ghiani et al. [7] and the model PS^+ perform well for these instances and seem also "comparable" to each other with respect to the numbers of solved instances and solution runtimes. However, these loosely-restricted instances are not appropriate for the strong modeling of hop and distance constraints associated to the layered graph models presented in this paper. The reason is that the benefit (if any) may not compensate the increase in size of the models.

Table 9: Comparison of numbers of solved instances and average CPU times (only for solved instances) of our branch-and-cut algorithms based on different models for instances generated in the same way as in Ghiani et al. [7] for $\gamma = 0.85$. Bold values denote the best algorithms in a row. ("-" ... time or memory limit reached)

			# instances solved (out of 5)					avg. CPU times in seconds				
α	β	$\|V\|$	PS^+	PD^+	DD^+	$PDDD^+$	$3PD^+$	PS^+	PD^+	DD^+	$PDDD^+$	$3PD^+$
0.20	1.00	20	4	**5**	**5**	**5**	**5**	342	**17**	366	76	48
		40	2	**4**	3	3	2	**0**	1316	705	3177	**0**
		60	2	2	**3**	**3**	2	**0**	**0**	2633	3232	**0**
		80	**0**	**0**	**0**	**0**	**0**	-	-	-	-	-
		100	**2**	**2**	0	0	0	**8**	13	-	-	-
	1.33	20	4	**5**	**5**	**5**	**5**	1045	1968	261	47	**19**
		40	2	**4**	**4**	3	2	**57**	1415	2273	3081	211
		60	**1**	**1**	0	0	0	7413	**3443**	-	-	-
		80	**0**	**0**	**0**	**0**	**0**	-	-	-	-	-
		100	**0**	**0**	**0**	**0**	**0**	-	-	-	-	-
	1.67	20	**5**	**5**	**5**	**5**	**5**	326	35	34	7	**4**
		40	3	3	**4**	**4**	**4**	272	**75**	374	570	1189
		60	**3**	2	0	0	0	849	**278**	-	-	-
		80	**1**	**1**	0	0	0	**300**	1107	-	-	-
		100	**0**	**0**	**0**	**0**	**0**	-	-	-	-	-
0.35	1.00	20	**5**	**5**	**5**	**5**	**5**	**1**	4	294	349	1626
		40	**5**	**5**	3	3	2	278	105	76	94	**0**
		60	**4**	**4**	3	3	3	1971	2358	**0**	**0**	**0**
		80	**5**	**5**	**5**	**5**	**5**	**0**	**0**	**0**	**0**	**0**
		100	**4**	**4**	3	3	3	**0**	1	**0**	**0**	**0**
	1.33	20	**5**	**5**	**5**	**5**	**5**	2	**1**	18	22	38
		40	**4**	**4**	0	0	0	**766**	1395	-	-	-
		60	**2**	1	0	0	0	3180	**1041**	-	-	-
		80	**1**	0	0	0	0	**1509**	-	-	-	-
		100	**0**	**0**	**0**	**0**	**0**	-	-	-	-	-
	1.67	20	**5**	**5**	**5**	**5**	**5**	412	1912	30	**17**	39
		40	**4**	**4**	1	0	0	**8**	24	69	-	-
		60	**3**	1	0	0	0	**2645**	8510	-	-	-
		80	**4**	1	0	0	0	**582**	5381	-	-	-
		100	**3**	0	0	0	0	**1107**	-	-	-	-
0.50	1.00	20	**5**	**5**	**5**	4	4	**0**	4	669	**0**	**0**
		40	**5**	**5**	**5**	**5**	**5**	**0**	**0**	**0**	**0**	**0**
		60	**5**	**5**	**5**	**5**	**5**	**0**	**0**	**0**	**0**	**0**
		80	**5**	**5**	3	3	3	**0**	1	**0**	**0**	**0**
		100	**5**	**5**	4	4	4	**0**	1	**0**	**0**	**0**
	1.33	20	**5**	**5**	**5**	**5**	4	**12**	183	418	422	1659
		40	**5**	3	0	0	0	**564**	1151	-	-	-
		60	**5**	2	0	0	0	**162**	4436	-	-	-
		80	**5**	0	0	0	0	**471**	-	-	-	-
		100	**4**	1	0	0	0	**1994**	3034	-	-	-
	1.67	20	**5**	**5**	**5**	**5**	**5**	207	666	**83**	166	1426
		40	**5**	3	0	0	0	**296**	1590	-	-	-
		60	**5**	3	0	0	0	**125**	2404	-	-	-
		80	**5**	0	0	0	0	**318**	-	-	-	-
		100	**0**	**0**	**0**	**0**	**0**	-	-	-	-	-

Table 10: Comparison of numbers of solved instances and average CPU times (only for solved instances) of our branch-and-cut algorithms based on different models for instances generated in the same way as in Ghiani et al. [7] for $\gamma = 0.95$. Bold values denote the best algorithms in a row. ("-" . . . time or memory limit reached)

			# instances solved (out of 5)					avg. CPU times in seconds				
α	β	$\|V\|$	PS$^+$	PD$^+$	DD$^+$	PDDD$^+$	3PD$^+$	PS$^+$	PD$^+$	DD$^+$	PDDD$^+$	3PD$^+$
0.20	1.00	20	4	**5**	4	**5**	**5**	1639	**1**	26	223	138
		40	2	**4**	1	1	1	1807	**349**	6945	5973	8686
		60	0	**1**	0	0	0	-	**2197**	-	-	-
		80	**0**	**0**	**0**	**0**	**0**	-	-	-	-	-
		100	**0**	**0**	**0**	**0**	**0**	-	-	-	-	-
	1.33	20	**5**	**5**	4	**5**	**5**	805	**2**	3	69	267
		40	3	**4**	2	2	1	**110**	126	3188	2203	3454
		60	**3**	**3**	0	0	0	922	**188**	-	-	-
		80	1	**2**	0	0	0	7790	**7775**	-	-	-
		100	**0**	**0**	**0**	**0**	**0**	-	-	-	-	-
	1.67	20	**5**	**5**	**5**	**5**	**5**	487	23	122	40	**17**
		40	4	4	**5**	**5**	4	40	**8**	1420	3066	4767
		60	**5**	**5**	0	0	0	3082	**464**	-	-	-
		80	**3**	**3**	0	0	0	2883	**1856**	-	-	-
		100	**1**	**1**	0	0	0	4408	**1055**	-	-	-
0.35	1.00	20	4	**5**	**5**	**5**	**5**	**48**	134	304	373	1863
		40	4	**5**	0	0	0	1429	**654**	-	-	-
		60	**5**	**5**	2	2	2	221	605	**0**	**0**	**0**
		80	**3**	1	1	1	1	887	**0**	**0**	**0**	**0**
		100	**1**	0	0	0	0	**618**	-	-	-	-
	1.33	20	4	**5**	**5**	**5**	**5**	**6**	742	308	254	862
		40	**5**	**5**	0	0	0	1516	**403**	-	-	-
		60	**5**	3	0	0	0	1387	**633**	-	-	-
		80	**3**	0	0	0	0	**2277**	-	-	-	-
		100	**1**	0	0	0	0	**184**	-	-	-	-
	1.67	20	**5**	**5**	**5**	**5**	**5**	25	**17**	291	272	254
		40	**4**	**4**	1	1	0	**5**	9	109	4703	-
		60	**4**	**4**	0	0	0	**90**	910	-	-	-
		80	**5**	2	0	0	0	**448**	2398	-	-	-
		100	**3**	1	0	0	0	**197**	1538	-	-	-
0.50	1.00	20	**5**	**5**	**5**	**5**	4	**0**	1	869	1216	1197
		40	**5**	**5**	1	1	1	272	379	**0**	**0**	**0**
		60	**4**	**4**	3	3	3	481	1164	**0**	**0**	**0**
		80	**5**	3	2	2	2	379	5	**0**	**0**	**0**
		100	**5**	**5**	3	3	3	5	987	**0**	**0**	**0**
	1.33	20	**5**	4	**5**	**5**	3	702	**3**	573	1547	2371
		40	**5**	4	0	0	0	**56**	290	-	-	-
		60	**5**	4	0	0	0	**72**	1582	-	-	-
		80	**5**	1	0	0	0	**155**	7658	-	-	-
		100	**5**	2	0	0	0	**189**	3825	-	-	-
	1.67	20	4	4	**5**	**5**	4	**1**	**1**	379	859	795
		40	**5**	**5**	0	0	0	**34**	1582	-	-	-
		60	**5**	4	0	0	0	**66**	361	-	-	-
		80	**5**	3	0	0	0	**141**	3494	-	-	-
		100	**5**	0	0	0	0	**1952**	-	-	-	-

Table 11: Comparison of numbers of solved instances and average CPU times (only for solved instances) of our branch-and-cut algorithms based on different models for instances generated in the same way as in Ghiani et al. [7] for $L = \infty$. Bold values denote the best algorithms in a row. ("-" ... time or memory limit reached)

			# solved		avg. $t[s]$	
α	β	$\lvert V\rvert$	PS$^+$	PD$^+$	PS$^+$	PD$^+$
0.20	1.00	20	4	**5**	459	**1**
		40	1	**5**	93	**37**
		60	0	**5**	-	**2091**
		80	**0**	**0**	-	-
		100	**0**	**0**	-	-
	1.33	20	**5**	**5**	268	**1**
		40	4	**5**	**35**	40
		60	4	**5**	2365	**1572**
		80	**3**	**3**	1612	**566**
		100	0	**2**	-	**6161**
	1.67	20	**5**	**5**	76	**0**
		40	**5**	**5**	19	**2**
		60	**5**	**5**	814	**48**
		80	4	**5**	**190**	1734
		100	2	**5**	**1989**	3303
0.35	1.00	20	**5**	**5**	144	**1**
		40	**5**	**5**	588	**367**
		60	**5**	**5**	1311	**601**
		80	**5**	3	**2109**	4815
		100	**3**	0	**1350**	-
	1.33	20	**5**	**5**	494	**3**
		40	4	**5**	**19**	124
		60	**5**	**5**	**1002**	1441
		80	**5**	**5**	**335**	2760
		100	**3**	0	**1000**	-
	1.67	20	**5**	**5**	1	**0**
		40	4	**5**	**1**	182
		60	**5**	**5**	994	**925**
		80	**5**	**5**	**225**	2377
		100	**3**	**3**	**64**	1541
0.50	1.00	20	**5**	**5**	171	**1**
		40	4	**5**	**44**	84
		60	**5**	4	1122	**776**
		80	**3**	1	**488**	2838
		100	**4**	1	**455**	1498
	1.33	20	**5**	**5**	**0**	**0**
		40	**5**	**5**	**8**	56
		60	**5**	**5**	**5**	48
		80	**5**	**5**	**30**	1863
		100	**5**	3	**70**	1768
	1.67	20	**5**	**5**	**0**	**0**
		40	**5**	**5**	**13**	44
		60	**5**	**5**	**8**	52
		80	**5**	**5**	**31**	1547
		100	**5**	3	**75**	620