UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE ESTATÍSTICA E INVESTIGAÇÃO OPERACIONAL

A Comparative Analysis of Formulations for the Hamiltonian p-Median Problem

Francisco Miguel Paulo Canas

Mestrado em Estatística e Investigação Operacional
Especialização em Investigação Operacional

Dissertação orientada por:
Professor Doutor Luís Eduardo Neves Gouveia

2022

First and foremost, I wish to thank my grandparents and my parents for doing everything they could to raise me and help me afford the education I needed to get where I am today.

Secondly, I would like to thank my two siblings, Tiago and João, for being the best brothers (and friends) I could ask for, and for being a source of support and inspiration.

I'm also thankful to all the professors who've accompanied me during my stay in FCUL, and especially to my adviser, professor Luís Eduardo Neves Gouveia. Your guidance, support, friendship and especially patience during these last two years has been invaluable, and I'm very happy I had the opportunity to work with someone I've grown to admire so much. It's been an honour.

Last but most certainly not least, I thank Sofia for inspiring and believing in me, sometimes more than I believed in myself, and for always lifting me up when I needed it the most - words are not enough to describe my gratitude to you. Thank you for sticking with me through thick and thin. I wouldn't be where I am if it weren't for you.

Thank you all. I hope I make you proud.

Resumo

Em otimização, existem problemas que, dadas as suas semelhanças, são agrupados em conjuntos de problemas. Em logística e gestão de operações, por exemplo, os problemas costumam ser agrupados em três grupos: problemas de localização (nos quais o objetivo é determinar as melhores localizações para estabelecimentos como fábricas ou centros de distribuição), problemas de roteamento (nos quais o objetivo é planear rotas, por exemplo, de veículos que efetuem a entrega de produtos a vários clientes) e problemas de gestão de inventário (nos quais o objetivo é determinar as melhores políticas de gestão de inventário). Em todos estes problemas, o objetivo é otimizar alguma medida de desempenho - por exemplo, minimizar o custo - satisfazendo certas restrições.

Ao planear uma cadeia de abastecimento, é habitual resolver problemas de grupos diferentes separadamente. Por exemplo, pode-se optar por localizar centros de distribuição primeiro, e só depois planear as rotas para os veículos de distribuição. Esta abordagem tem as suas vantagens (incluindo a maior facilidade em modelar os problemas e a maior celeridade na sua resolução), mas também tem uma grande desvantagem - a cadeia de abastecimento como um todo pode não ser "ótima". Regressando ao exemplo anterior, se os problemas de localizar centros de distribuição e planear as rotas de distribuição forem resolvidos separadamente, pode acontecer que os centros de distribuição acabem por ser localizados em zonas mais baratas mas também mais distantes dos clientes, resultando em custos de transporte mais elevados devido ao maior consumo de combustível, desperdício de tempo, e possivelmente até maiores custos de manutenção das viaturas de transporte, visto estas necessitarem de percorrer maiores distâncias.

Pode assim ser interessante integrar a otimização de várias áreas da cadeia de abastecimento simultaneamente. Apesar das dificuldades anteriormente referidas, como maior dificuldade de modelação e resolução do problema, o desenho de uma cadeia de abastecimento mais eficiente é um grande incentivo à integração da resolução de problemas de diferentes "grupos". Pode-se, por exemplo, procurar resolver simultaneamente problemas de localização e de roteamento - um problema no qual se integre um problema de localização e um problema de roteamento diz-se um problema de *location-routing*.

Neste trabalho, é estudado o problema da p-Mediana Hamiltoniana (doravante PpMH), um problema de otimização combinatória no qual, dado um grafo não orientado $G = (V, E)$ tal que a cada aresta é associado um custo, o objetivo passa por determinar a forma mais barata de particionar o conjunto dos nodos em exatamente p subconjuntos tais que cada subconjunto é conexo por um único ciclo hamiltoniano. Este é, portanto, um problema que pode ser visto como uma generalização do problema do Caixeiro Viajante (ver, por exemplo, [1]) (doravante PCV) - torna-se neste problema quando $p = 1$ - e pode ser considerado um problema de *location-routing* - se em cada ciclo se considerar um dos nodos um depósito (e muitos dos modelos para este problema podem facilmente ser adaptados para considerar isto mesmo), este problema não é mais do que um problema da p-Mediana no qual, para cada depósito, se planeia a rota de distribuição percorrida pela sua viatura.

Ao considerar modelos em Programação Linear Inteira para o PCV, é comum partir-se de um modelo de afetação que tem como conjunto de soluções admissíveis o conjunto de todas as possíveis partições do grafo num qualquer número de ciclos, sendo depois necessário acrescentar restrições a este modelo que excluam soluções com dois ou mais ciclos. De forma análoga, ao considerar modelos em Programação Linear Inteira para o PpMH, são também necessários conjuntos de restrições que excluam soluções com mais de p ciclos, e estes conjuntos de restrições são muitas vezes semelhantes aos utilizados para o PCV, apesar de precisarem de algumas modificações. No entanto, estes conjuntos não são suficientes para assegurar que qualquer solução admissível para o modelo inclui exatamente p ciclos, visto continuarem a ser admissíveis soluções com menos de p ciclos. De forma a excluir estas soluções, podem ser incluídos conjuntos de restrições adicionais, e serão estes conjuntos de restrições o foco desta dissertação.

Assim, este trabalho começa com uma breve introdução ao problema e uma revisão bibliográfica. São depois apresentadas várias formulações compactas para este problema. A apresentação destes modelos será separada em três partes.

Na primeira parte, são apresentados conjuntos de restrições e variáveis que são comuns a todos os modelos aqui apresentados. Em todos estes modelos são também incluídas variáveis que indicam se um nodo age como depósito do ciclo ao qual pertence - isto mesmo que, na prática, não exista a necessidade de designar certos nodos como depósitos. Nos modelos aqui apresentados é sempre necessário considerar certos nodos como depósitos, pois estes modelos excluem soluções com mais (ou menos) de p ciclos indicando que devem existir exatamente p depósitos e que cada ciclo deve incluir no mínimo (ou no máximo) um depósito.

Na segunda parte, é apresentado um conjunto de restrições que é uma adaptação das restrições apresentadas em [2] para o PCV e que visa excluir soluções com mais de p ciclos. Neste modelo, as variáveis adicionais indicam a posição de um nodo no circuito ao qual este pertence.

Na terceira parte, são apresentados dois modelos distintos (mas semelhantes) que visam excluir soluções com menos de p ciclos. Ambos os modelos utilizam variáveis que associam "etiquetas" a nodos - a diferença entre os modelos está nos valores que estas etiquetas podem tomar. No primeiro modelo, são utilizadas variáveis que associam nodos a depósitos. Por outras palavras, para cada nodo, o valor que a etiqueta toma é o índice do nodo que age como depósito do ciclo ao qual o nodo pertence. É evidente, portanto, que a etiqueta de qualquer nodo que seja um depósito é o índice do próprio nodo. Já no segundo modelo, são utilizadas variáveis que associam nodos a ciclos. Neste modelo, os ciclos são numerados de 1 a p, e para cada nodo, o valor que a sua etiqueta toma é o número associado ao ciclo. Por outras palavras, se um nodo pertence ao k-ésimo ciclo, a sua etiqueta tomará o valor k. Este último modelo parece não ter sido referido na literatura anterior a esta dissertação, ao passo que um modelo bastante semelhante ao primeiro modelo parece ter sido referido pela primeira vez para um problema semelhante ao HpMP em [3]. Para cada modelo foram ainda propostas algumas modificações visando melhorar as relaxações lineares e os tempos computacionais.

Por fim, o trabalho termina com a discussão dos resultados obtidos de uma experiência computacional que visa comparar os modelos aqui apresentados e de possíveis trabalhos futuros relacionados com esses mesmos modelos. A experiência computacional mostra que algumas das modificações propostas para os modelos em que nodos são associados a ciclos resultam em melhorias nos tempos computacionais, mas não todas. Esta experiência permite ainda concluir que apesar das melhorias resultantes de algumas das modificações propostas, os modelos em que os nodos são associados a depósitos têm o melhor desempenho, permitindo obter em várias instâncias tempos de resolução muito mais reduzidos do que os obtidos utilizando os modelos em que nodos são associados a ciclos.

Palavras-chave: Otimização Combinatória, Problema do Caixeiro Viajante, Problema da p-Mediana, Problema da p-Mediana Hamiltoniana, Formulações Compactas

Abstract

In this dissertation we study the Hamiltonian p-Median Problem, a combinatorial optimization problem in which, given an undirected graph $G = (V, E)$ and a cost for each edge, the objective is to find the cheapest way to partition the set of nodes into p subsets with each subset being connected by a single cycle. This is a problem which may therefore be seen as a generalization of the Travelling Salesman Problem (TSP).

When working with MILP models for the TSP, sets of constraints to prevent feasible solutions with more than one cycle are added to an assignment formulation. Similarly, when working with such models for the HpMP, sets of constraints to prevent feasible solutions with more than p cycles can also be added to an assignment formulation, and these are often very similar to sets of constraints already used in models for the TSP, albeit with some modifications. However, these sets are not sufficient to guarantee every feasible solution has exactly p cycles, since it may have fewer than p cycles. To this end, additional sets of constraints for preventing solutions with less than p cycles may be introduced, and these will be the focal point of this work.

The work begins with a brief introduction to the problem and some literature review. After that, several compact formulations for this problem are presented. The presentation of these models will be split into three parts. In the first part, a model upon which all other models are built is presented. The second part focuses on a model used to prevent solutions with more than p cycles, while the third part focuses on two models used to prevent solutions with less than p cycles (in which nodes are assigned to depots or cycles), accompanied by some valid inequalities. Finally, some of the models presented in this work are tested and the results and possibilities for future work are discussed.

Keywords: Combinatorial Optimization, Travelling Salesman Problem, p-Median Problem, Hamiltonian p-Median Problem, Compact Formulations

Contents

List of Figures **vii**

List of Tables **x**

List of Abbreviations **xi**

1 Introduction **1**

2 The Hamiltonian p-Median Problem **5**
2.1 Problem Description 5
2.2 Literature Review 6

3 Formulations **9**
3.1 BASE Formulation 10
3.2 Preventing more than p Cycles - Adapted DL Constraints 12
3.3 Preventing less than p Cycles - Node-Depot Assignment 14
3.3.1 A Valid Inequality 15
3.4 Preventing less than p Cycles - Node-Cycle Assignment 16
3.4.1 A Valid Inequality 17
3.4.2 Improving Constraints 19
3.5 An Overview of the Models 21

4 Computational Experiment **25**
4.1 Hardware / Software Configurations and Test Instances 25
4.2 Test Results 25

5 Conclusions and Future Work **31**
5.1 Main Conclusions 31
5.2 Future Work 32

A Test Results **35**

List of Figures

3.1 Three different solutions for an instance of the HpMP, but only one is feasible 11
3.2 Circuit with no depots, violating some of the adapted DL constraints 13
3.3 Circuit with one depot, respecting all of the adapted DL constraints 13

4.1 Diagram relating the LR bounds of incomplete models 30

List of Tables

3.1 Constraints used in each complete model . . . 22
3.2 Constraints used in each incomplete model . . . 22

4.1 Average gaps for complete models for every value of p . . . 26
4.2 Number of times the model in the row performed better than the model in the column (for complete models) . . . 27
4.3 Number of times the time limit was reached (for complete models) . . . 27
4.4 Average times for tests in which CPLEX never reached the time limit (for complete models) 27
4.5 Average gaps for incomplete models for every value of p . . . 28
4.6 Number of times the model in the row performed better than the model in the column (for incomplete models) . . . 29
4.7 Number of times the time limit was reached (for incomplete models) . . . 29
4.8 Average times for tests in which CPLEX never reached the time limit (for incomplete models) . . . 29

A.1 Test results for complete models (instance: gr24) . . . 36
A.2 Test results for complete models (instance: fri26) . . . 37
A.3 Test results for complete models (instance: bayg29) . . . 38
A.4 Test results for complete models (instance: swiss42) . . . 39
A.5 Test results for complete models (instance: att48) . . . 40
A.6 Test results for complete models (instance: gr48) . . . 41
A.7 Test results for complete models (instance: hk48) . . . 42
A.8 Test results for complete models (instance: eil51) . . . 43
A.9 Test results for complete models (instance: berlin52) . . . 44
A.10 Test results for complete models (instance: brazil58) . . . 45
A.11 Test results for complete models (instance: st70) . . . 46
A.12 Test results for complete models (instance: eil76) . . . 47
A.13 Test results for complete models (instance: pr76) . . . 48
A.14 Test results for complete models (instance: rat99) . . . 49
A.15 Test results for complete models (instance: kroA100) . . . 50
A.16 Test results for complete models (instance: kroB100) . . . 51
A.17 Test results for complete models (instance: kroC100) . . . 52
A.18 Test results for complete models (instance: kroD100) . . . 53
A.19 Test results for complete models (instance: kroE100) . . . 54
A.20 Test results for complete models (instance: rd100) . . . 55
A.21 Test results for incomplete models (instances: gr24, fri26) . . . 56

A.22 Test results for incomplete models (instances: bayg29, swiss42) 57
A.23 Test results for incomplete models (instances: att48, gr48) 58
A.24 Test results for incomplete models (instances: hk48, eil51) 59
A.25 Test results for incomplete models (instances: berlin52, brazil58) 60
A.26 Test results for incomplete models (instances: st70, eil76) 61
A.27 Test results for incomplete models (instances: pr76, rat99) 62
A.28 Test results for incomplete models (instances: kroA100, kroB100) 63
A.29 Test results for incomplete models (instances: kroC100, kroD100) 64
A.30 Test results for incomplete models (instances: kroE100, rd100) 65

List of Abbreviations

ATSP	Asymmetric Travelling Salesman Problem
DL	Desrochers and Laporte
HpMP	Hamiltonian p-Median Problem
LR	Linear Relaxation
MILP	Mixed Integer Linear Programming
SB	Symmetry Breaking
TSP	Travelling Salesman Problem

Chapter 1

Introduction

In optimization, there are problems which, given their similarities, are often grouped into broader sets of problems. For instance, in logistics and operations management (a field which greatly benefits from the application of tools from Operational Research), problems are often split into three groups: location problems (in which the goal is to determine the best locations for facilities, such as distribution centers), routing problems (in which the goal is to plan routes, such as the routes travelled by vehicles delivering products to several clients) and inventory management problems (in which the goal is to determine the best inventory control policies). In all these problems, the goal is to optimize some performance metric - that usually means minimize costs - satisfying certain constraints.

When planning a supply chain, problems from different groups are often solved separately, and therefore, the whole supply chain is not planned simultaneously. For instance, one may choose to locate the facilities first, and after having decided where to locate the facilities, plan the routes for the distribution vehicles. This approach, while having some advantages (such as the ease of mathematically modelling and solving the problems), also presents a remarkable disadvantage - the final solution (that is, the supply chain as a whole) will most likely be suboptimal. Returning to the example of locating facilities before planning routes, for example, if these problems are solved separately, it may be the case that the facilities end up being located in cheap locations but further away from all the other clients to be supplied, meaning the delivery costs will be greater than they could be since more fuel and time will be spent, and the vehicles used to deliver products to clients may require more frequent maintenance as they travel more.

Regardless, in spite of the additional difficulties, it is possible to optimize different areas of a supply chain simultaneously. For example, the authors of [4] consider a location-inventory problem in which the goal is to simultaneously minimize the facility locating and inventory costs.

The TSP (see, for instance, [1]) is one of the most popular combinatorial optimization problems of all time. In this problem, given a set of locations to visit and knowing the cost of travelling between any pair of two different locations, the goal is to determine the cheapest route that visits all locations exactly once and returns to the starting point. This is therefore a routing problem.

On the other hand, the p-Median Problem [5] is a location problem. In this problem, given a set of locations whose demands must be satisfied, the goal is to satisfy the demand of those locations by installing exactly p facilities in some of those locations and assigning each location to exactly one facility. The costs of serving the total demand of any location with a facility in some other location are known, and so are the costs of placing a facility in each location. It is also assumed that if a facility is installed in some location, the demand of that location is fulfilled for free by that same facility. The goal is to determine where to locate the p facilities and which locations to assign to each facility in order to minimize the total cost of locating facilities and fulfilling the demand from all locations.

It may therefore be of interest to solve both these problems simultaneously - even though this will result in a more complex problem, it should also result in a better solution, allowing for a more efficient supply chain. The HpMP is a location-routing problem which is a natural generalization of the Travelling Salesman and p-Median problems. Given a set of locations that must be visited and knowing the cost of travelling between each pair of locations and the costs of locating facilities (which will be referred to as "depots" throughout the rest of this work), the goal is to determine where to locate each of the p depots, which locations to visit from each depot, and the cheapest route departing from each depot that visits all the locations assigned to that depot exactly once and returns to the depot, making sure that each location is assigned to exactly one depot. This is a problem which arises in practical applications such as school location, multi-depot vehicle routing or the laser multi-scanner problem [6], showing this problem may also be applied outside the domain of logistics.

This problem may also be stated without considering the locations of the depots, at which point it becomes a problem of determining the cheapest set of p cycles such that each location belongs to exactly one cycle. This is the problem covered in this dissertation, and we specifically focus on MILP formulations for the HpMP.

MILP formulations for the TSP are usually built upon an assignment model for which, with no additional constraints, the set of feasible solutions corresponds to the set of all possible partitions of the graph in any number of cycles. Therefore, in order to avoid solutions with more than one cycle, additional constraints are added to the original assignment model. Something similar is done for the HpMP - starting with a formulation very similar to the assignment formulation used for the TSP, additional constraints are added to limit the number of cycles in any feasible solution to exactly p.

Every model presented in this work uses a set of binary edge variables $u_{ij} \in \{0, 1\}$, $\forall (i, j) \in E$, which take value 1 if edge (i, j) is part of any cycle in the solution. In every model presented in this dissertation, one of the nodes in each cycle also acts as the "depot" of that cycle - this is because these models prevent solutions with less than p cycles by stating that there must be exactly p depots and by introducing additional constraints which imply each cycle must include exactly one depot. To this end, every model presented here also uses a set of binary variables $y_i \in \{0, 1\}$, $\forall i \in V$, which indicate if the corresponding node acts as the depot of the cycle it belongs to.

It was previously mentioned that, similarly to what is done for the TSP, additional constraints are added to an assignment formulation in order to set the number of cycles of any feasible solution to exactly p. This is often achieved by combining two (usually independent) sets of constraints - one which sets the maximum number of cycles to p and one which sets the minimum number of cycles to p.

In order to set the maximum number of cycles in a feasible solution to p, one adaptation of a compact model (that is, a model with a polynomial number of variables and constraints) first presented in [2] for the TSP is presented. This model requires the introduction of a new set of integer variables $u_i \in \mathbb{N}$, $\forall i \in V$ - for each node, the corresponding u_i variable indicates the position of the node in the circuit it belongs to (the position of each depot is 0).

The main part of this dissertation, however, focuses on compact models to prevent solutions with less than p cycles. This is motivated by the fact that constraints known for this purpose seem to be less well known than constraints which prevent solutions with more than p cycles. To this end, we present and discuss two such models, from which other "improved" models can be built.

Both models prevent solutions with less than p cycles by assigning "labels" to nodes. The main differences lie in one characteristic of the models - the values these labels take (for a given node i, for instance, they can indicate which cycle node i belongs to, assuming each cycle is numbered from 1 to p, or which node serves as the depot of the cycle node i belongs to).

Following the framework we have just mentioned, the model we begin with also considers a set of integer variables $k_i \in \mathbb{N}$, $\forall i \in V$, which take the value of the label of the corresponding node. In this model, the label of a node is the index of the node that plays the role of depot of the cycle node i belongs to (in other words, $k_i = d$ if and only if node d acts as the depot of the cycle to which node i belongs). This also means that a node d acts as a depot if and only if $k_d = d$. An additional set of valid inequalities for this model are also included.

This model also considers that the depot of any given cycle must always be the node with the lowest index which belongs to it. It is not apparent how to model the problem using these variables and linear constraints without imposing that condition - however, this restriction seems to benefit the model, as it also reduces the number of equivalent solutions. From now on, we will refer to any way of reducing the number of equivalent solutions as a symmetry breaking strategy (SB strategy).

The last model includes instead a set of integer variables $v_i \in \mathbb{N}$, $\forall i \in V$, which, just like the k_i variables, also take the value of the label of the corresponding node. However, this time, the label of a node is the cycle to which the node belongs (in other words, $v_i = k$ if and only if node i belongs to the k-th cycle), instead of being the index of the node that plays the role of depot of the cycle node i belongs to.

We also observe that this model requires both nodes to act as depots and the cycles to be numbered (it is always assumed these are numbered from 1 to p), which results in even more equivalent solutions (a given cycle may be the first, or the second or third cycle, for instance). In this case, as an SB strategy, it is both considered that the depot of a cycle is the node with the lowest index which belongs to it and that all cycles are sorted by ascending order of the indices of their depots. It turns out that, similarly to what happens when considering the first model, it is also not apparent how to prevent less than p cycles using the v_i variables without imposing both these SB strategies. Similarly to the first model, sets of valid inequalities and improvements to known inequalities are presented for this model.

This work begins with a brief introduction to the problem and a literature review, followed by several extended formulations for the HpMP. These formulations are all built upon a “BASE” formulation (the aforementioned model similar to the assignment model used for the TSP), to which two additional sets of constraints are added, as previously described. The presentation of this “BASE” formulation is followed by three sections. The first section includes a set of constraints which is an adaptation of a popular set first introduced in [2] for the TSP. The second and third sections include the previously mentioned models used to prevent solutions with less than p cycles, which assign nodes to depots or to cycles, and some valid inequalities for each model. The work ends with the discussion of results obtained from a computational experiment conducted in order to compare the models presented here and the possibility of further research regarding compact formulations for this problem.

Chapter 2

The Hamiltonian p-Median Problem

In this chapter we define the HpMP. The chapter begins with a description of the problem followed by a brief review of past literature concerning this problem.

2.1 Problem Description

The HpMP is defined on a complete undirected graph $G = (V, E)$, where $V = \{1, \ldots, n\}$ is the vertex set and $E = \{(i, j), \forall i, j \in V : i < j\}$ is the edge set. To every edge $(i, j) \in E$ we associate a cost c_{ij}. In the HpMP, given this graph and set of costs, the objective is to find the cheapest way to partition the vertex set into p subsets of vertices, with each partition being connected by a single cycle.

It is easy to see that, for $p = 1$, the HpMP becomes the classical TSP (since it only asks what the cheapest cycle that goes through all the nodes in the graph is). Since the TSP is NP-hard, the HpMP is also NP-hard. That also means that, for $p = 1$, any models and methods already used for the TSP may also be applied to the HpMP. However, for $p > 1$, this is not the case - those models and algorithms may need to be modified considerably, and different models or algorithms may be necessary.

Three observations are in order: first, throughout this work, it is assumed that any feasible solution must not include cycles with two nodes. This assumption is considered due to some challenges that arise when modelling this problem in an undirected graph allowing for cycles with two nodes. While it is possible to model such problems, some modifications are required (a good example of this may be seen in [7]).

Second, this problem may also be defined on a directed graph. The adaptation of most models included in this work to allow for problems defined on directed graphs is very straightforward. Although considering the problem to be defined on a directed graph allows for a more general version of the problem, considering an undirected graph means the mathematical models for the HpMP may feature less variables, which means these may be easier to solve, and in many practical applications, modelling the problem using an undirected graph is reasonable. These two last observations mean the distinction between symmetrical and asymmetrical problems is important.

The last observation is that assuming the graph is complete is also not restrictive, since, for every non-existent edge, it may be assumed that that edge exists with an arbitrarily large cost.

2.2 Literature Review

For some of the reasons outlined in chapter 1, location-routing problems have been a subject of interest for a long time - as mentioned in [8], some of the earliest articles on location-routing problems are almost 50 years old as of the time of writing this dissertation.

However, the HpMP does not seem to be as old, having been introduced for the first time in [9], where it is presented as a mixed routing location problem embedding the p-Median problem and the TSP. While this is not taken into consideration in this dissertation, many models and solving methods for the HpMP may be easily adapted to consider fixed costs when setting certain nodes as depots - this includes every extended MILP formulation for the HpMP presented in the following pages. The authors of [9] also presented some formulations and several heuristics for the HpMP and for the capacitated HpMP, a generalization of this problem in which each node is considered to have a certain “demand”, and for each depot, the total demand it serves (that is, the demand of all the nodes in the cycle it belongs to) must be within a certain range, and tested the heuristics on instances with up to 100 nodes.

Ten years later, *Glabb and Pott* [6] studied a new formulation for the asymmetric HpMP which uses variables that assign arcs to circuits and investigated the basic properties of the associated Hamiltonian p-Median polytope. This formulation featured an exponential number of constraints, but given the focus of the work was on the polytope associated with that formulation, no branch-and-cut algorithms were presented and no computational experiments were conducted, making this a more theoretical work which differs considerably from most of the work cited in this dissertation. To the best of the author’s knowledge, very little work focusing on the HpMP was published between [9] and [6], and only after [6] did the HpMP start attracting more attention.

In 2011, *Gollowitzer et al.* [10] presented and compared three different models for the HpMP. Just like some of the models presented in this dissertation, *Gollowitzer et al.* [10] also used variables which assign nodes to cycles - although the models presented in [10] to prevent solutions with less than p cycles seem to be related to some of the models presented in this dissertation (those in which nodes are assigned to cycles), these are not the same. However, the focus of *Gollowitzer et al.’s* [10] work is on sets of constraints used to prevent solutions with more than p cycles, while the focus of this dissertation is on sets of constraints used to prevent solutions with less than p cycles.

More recently, *Gollowitzer et al.* [11] presented new formulations for the HpMP and compared these with other models previously presented in other literature. While [11] includes proofs that a set partitioning model based on a similar model first introduced in [9] produces better LR bounds than the other models for the HpMP presented in [11], it does not include any practical results, since this model includes an exponentially large number of variables, and determining the cost of each variable in the objective function implies solving a TSP, which is an NP-hard problem. *Gollowitzer et al.* [11] also find that a p-Median based model seems to have the best LR bounds out of all the models studied in [11] (except for the set partitioning model mentioned earlier), although when it comes to computational times, it seems to perform worse than other natural formulations. One other interesting observation is that the aforementioned p-Median based model also seems to be related to some of the models presented in this dissertation (in particular, to those in which nodes are assigned to depots).

Only two years later, *Erdogan et al.* [12] proposed and tested a branch-and-cut algorithm (based on a model to which we will simply refer as *Erdogan et al.*’s model), a giant tour heuristic and an iterated local search algorithm for the HpMP which showed promising results. In particular, *Erdogan et al.* [12] compared their model with two models presented in [11] (one of them being the aforementioned p-Median model, which *Gollowitzer et al.* denoted by “Model 3”, and the other being a formulation on the natural variable space denoted by "Model 1") and found the branch-and-cut algorithm performed substantially better than those models, providing both better LR bounds and better computational times.

Marzouk et al. [13] presented and tested a branch-and-price algorithm based on a modified version of the set partitioning model presented in [11], and compared it against Model 1 from [11]. For large values of p, this algorithm performed quite well, and was able to solve instances with up to 200 nodes in one hour or less, and for many values of p, it got close to solving instances with up to 318 nodes in the same time limit.

While [3] handles a somewhat different problem, this work is also mentioned in this literature review for one reason: the model presented in section 3.3 is heavily inspired by the model first introduced by *Burger et al.* in [3], which showed promising results. In [3], the authors study the fixed-destination multi-depot multiple-salesmen TSP, a problem in which, given a set of depots (with each depot having a certain number of salesmen available), a set of clients and a set of costs for travelling between any pair of nodes, the objective is to find the cheapest routes each salesman must take such that each client is visited exactly once and each salesman returns to the depot it started from.

Another branch-and-cut algorithm presented by *Bektas et al.* [14] also had interesting results. This algorithm, based on a formulation denoted by "PQR", is compared with a model denoted by "x-v", which is an adaptation of *Erdogan et al.*'s model presented in [12]. Unlike *Erdogan et al.*'s model, however, x-v allows solutions including circuits with only two nodes and may be applied to the asymmetrical HpMP. In many instances, PQR outperformed x-v.

Interestingly, while some work has been dedicated to compact formulations for this problem (for instance, the p-Median model presented in [11]), it seems most of the work conducted so far regarding this problem has either focused on heuristic algorithms or on MILP formulations with an exponential number of variables (such as the set partitioning model presented in [9]) or constraints (such as the PQR model presented in [14] or *Erdogan et al.*'s model presented in [12]), which may then be solved using sophisticated branch-and-price or branch-and-cut algorithms. While these often perform well and have good LR bounds, the implementation of the algorithms used to solve these models means these are not as accessible as other alternatives, such as compact formulations which may be solved using MILP solvers (such as CPLEX or Gurobi) and without implementing specific algorithms to solve pricing or separation problems.

The focus of this work is therefore on the aforementioned compact formulations for the HpMP. Great emphasis is given to sets of constraints aimed at preventing any solutions that feature less than p cycles.

Chapter 3

Formulations

In this chapter, we present and discuss several MILP formulations for the HpMP. Section 3.1 will include a generic formulation upon which all other formulations for the HpMP presented here are built. Section 3.2 includes one set of constraints and variables used to prevent solutions with more than p cycles, while sections 3.3 and 3.4 include sets of constraints and variables used to prevent solutions with less than p cycles.

The model covered in section 3.1 is very similar to the assignment formulations from which most models for the TSP are built, and similarly, the constraints presented in section 3.2 are a modified version of the constraints first presented in [2] for the TSP.

Sections 3.3 and 3.4, however, are the focal point of this dissertation. This focus on models used to prevent solutions with less than p cycles is motivated by the fact that constraints known for this purpose seem to be less well known than inequalities which prevent solutions with more than p cycles.

Both those models prevent solutions with less than p cycles by assigning "labels" to nodes. The main differences lie in one characteristic of the models - what these labels are (for a given node i, for instance, they can indicate which cycle node i belongs to, assuming each cycle is numbered from 1 to p, or which node serves as the depot of the cycle node i belongs to).

One observation is in order: in this work, only extended formulations (as opposed to natural formulations) for the HpMP are studied. A formal definition of natural formulations for combinatorial optimization problems may be found in [15]. More informally, the difference between natural and extended formulations is that natural formulations include minimal sets of variables, while extended formulations include additional sets of variables that may not be strictly necessary to model the problem, but usually make it easier to accurately model the problem. For instance, a natural formulation for the HpMP only includes one variable per edge. Although the smaller number of variables in natural formulations may be seen as an advantage, natural formulations are not without their drawbacks - these often include exponentially large sets of constraints (which extended formulations often do not), meaning they cannot be solved in a regular computer the same way many extended formulations can. Instead, more sophisticated branch-and-cut algorithms are required.

The extended formulations studied here, then, although having larger sets of variables, have the advantage of often being more accessible and easier to implement. Unlike the aforementioned natural formulations (which usually feature exponentially large sets of constraints and require specific algorithms to be solvable in any ordinary computer), most extended formulations (including all the formulations presented in this work) can simply be implemented in a modelling language (such as OPL or Mosel) and solved using a commercial MILP solver, allowing for the usage of generic branch-and-bound (or branch-and-cut) algorithms which could be used to solve any MILP.

3.1 BASE Formulation

We start by presenting the generic formulation upon which all other formulations in this work are built. All models presented here are for the symmetric HpMP defined on a complete undirected graph $G = (V, E)$, where $E = \{(i, j), \forall i, j \in V : i < j\}$ and solutions containing cycles with only two nodes are not allowed. This is a consequence of the additional complexity of modelling cycles with two nodes on undirected graphs (although this is not impossible to model - a good example of this may be seen in [7]). In this formulation (which we will denote by "BASE"), binary variables $u_{ij} \in \{0, 1\}, \forall (i, j) \in E$, will take value 1 if the edge $(i, j) \in E$ belongs to one of the p cycles and 0 otherwise, while the binary variables $y_i \in \{0, 1\}, \forall i \in V$ indicate whether the corresponding node acts as the depot of the cycle it belongs to or not. To simplify the notation in the following pages while maintaining rigour, similarly to what is done in [12], we denote the ordered vertex pairs of every edge $(i, j) \in E$ as $\gamma(i, j) = \{(i, j), (j, i)\}$. We also denote by $\delta(i) = \{(i, j), \forall j \in V : (i, j) \in E\} \cup \{(j, i), \forall j \in V : (j, i) \in E\}$ the set of all edges between a node i and any other node in the graph. The BASE formulation is as follows:

$$\text{Min.} \sum_{(i,j)\in E} c_{ij} u_{ij} \tag{3.1}$$

$$\text{s.t.:} \sum_{e\in\delta(i)} u_e = 2, \qquad \forall i \in V \tag{3.2}$$

$$\sum_{i\in V} y_i = p, \tag{3.3}$$

$$\{(i, j) \in E : u_{ij} = 1\} \text{ forms at most } p \text{ cycles,} \tag{3.4}$$

$$\{(i, j) \in E : u_{ij} = 1\} \text{ forms at least } p \text{ cycles,} \tag{3.5}$$

$$u_{ij} \in \{0, 1\}, \qquad \forall (i, j) \in E \tag{3.6}$$

$$y_i \in \{0, 1\}, \qquad \forall i \in V \tag{3.7}$$

Constraints (3.2) are similar to the degree constraints also found in most models for the TSP, stating that each node is included in exactly one cycle and that exactly two edges featuring this node must be used. Constraints (3.6) define the u_{ij} variables as binary. A solution to the model described by the u_{ij} variables alone is composed of several disjoint cycles covering all nodes of the graph such that each cycle includes at least three nodes - this is an important detail, since it implies that any feasible solution to any model built upon this will also be composed of several disjoint cycles covering all nodes of the graph.

Constraint (3.3) states that there must be exactly p depots (exactly one depot per cycle) and constraints (3.7) simply state the y_i variables are binary. While it may seem unnecessary to designate some nodes as depots (since the problem we are considering simply consists of finding the lowest cost partition of the graph in exactly p cycles), this is necessary for every model presented in this dissertation. This happens because the model presented to prevent solutions with more than p cycles does so by stating that each cycle must include at least one depot, and likewise, the models presented to prevent solutions with less than p cycles do so by stating that each cycle must include at most one depot. This, when combined with constraint (3.3), implies each feasible solution must have exactly p cycles.

However, setting some nodes as depots raises an issue: if the choice of depots is irrelevant (which is not always the case - for instance, if there are fixed costs for considering a node to be a depot), this assignment results in many equivalent solutions, which may worsen the computational times. To exemplify, consider a cycle with m nodes. If any node in the cycle can act as a depot, using the y_i

variables, this cycle can be represented in m different ways - the only difference from one representation of this cycle to another is the node which acts as a depot. For instance, assuming the cycle with m nodes includes nodes $i_1, \ldots, i_m$, one representation can consider node i_1 to be the depot (and therefore consider $y_{i_1} = 1$ and $y_i = 0$ for every other node i in the cycle), whereas another representation can consider node i_2 to be the depot (and therefore consider $y_{i_2} = 1$ and $y_i = 0$ for every other node i in the cycle), and so on. These different representations of what is, in practice, the same solution, are what we refer to as "equivalent solutions", and since these representations are what comprises the set of feasible solutions for the mathematical models, having multiple different representations for the same solution means the set of feasible solutions is much larger than it should be, which can worsen the performance of the model. It is therefore of interest to consider ways of reducing the number of equivalent solutions as much as possible - we refer to such strategies as SB strategies. Every model presented in this dissertation features at least one SB strategy - only the node with the lowest index in a cycle can act as the depot of that cycle. As a matter of fact, it is not apparent how to adapt the models presented in sections 3.3 and 3.4 to allow solutions in which this SB strategy is violated.

It must also be noted that depending on the problem which is being modelled, this may or may not be irrelevant. In particular, if the choice of depots is relevant, the adoption of this SB strategy does not necessarily mean these formulations cannot be used to model the problem. Suppose, for example, that there are fixed costs for setting nodes as depots of their cycles. While this seems to invalidate these models, since one cannot pick the depots of a given cycle, this might not be true - assuming the goal is to simply pick the "cheapest" solution, the nodes can be sorted by ascending order of fixed costs. In that case, given any possible cycle, the node with the lowest index (which acts as a depot) and the node with the lowest fixed cost will always coincide, which means these models may be applied to such situations.

Constraints (3.4) and (3.5) are generic sets of constraints which can be written in many different ways - in sections 3.2, 3.3 and 3.4, some ways of writing such constraints are studied in depth. In this work, however, greater emphasis is given to sets aimed at preventing solutions with less than p cycles.

We conclude this section with the following example, which helps illustrate the set of feasible solutions for an instance of this problem. Consider a complete undirected graph with ten nodes, and consider the HpMP defined on that graph for $p = 2$. From the solutions seen in figure 3.1, only solution 3 is feasible (since it includes exactly p cycles). Solution 1 is not feasible (since it includes three cycles) and neither is solution 2 (since it only includes one cycle).

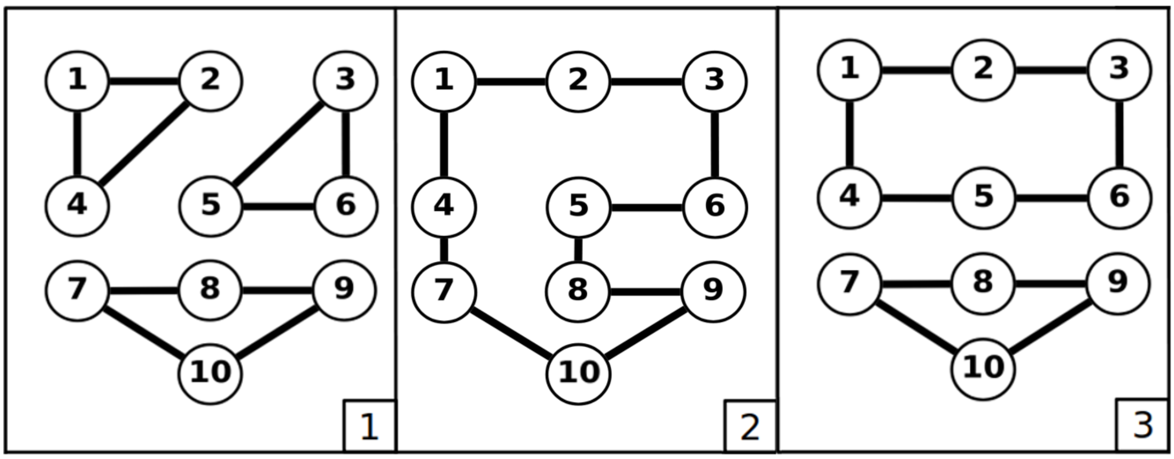

Figure 3.1: Three different solutions for an instance of the HpMP, but only one is feasible

3.2 Preventing more than p Cycles - Adapted DL Constraints

In this section we present a set of constraints and variables which is an adaptation of a set of constraints first introduced in [2] for the TSP and may be used to prevent solutions with more than p cycles. Many other sets of constraints used to model the TSP can also be adapted to prevent solutions featuring mode than p cycles for this problem. This set of constraints can therefore replace the generic constraints (3.4).

Before resuming, an observation must be made. It is not clear how to adapt the DL constraints to prevent solutions with more than p cycles using the u_{ij} variables previously presented. Therefore, to be able to utilize these constraints, a new set of variables must be used. Let $x_{ij} \in \{0, 1\}$, $\forall i, j \in V$ be a binary variable indicating whether nodes i and j belong to the same cycle and node j immediately follows node i (these variables may also be seen as simply indicating whether the arc (i, j) is used, in case the graph is directed). In that case, if these variables are added to the BASE model, two new sets of constraints must be added relating the u_{ij} and the x_{ij} variables and stating the x_{ij} variables are binary, as follows:

$$u_{ij} = x_{ij} + x_{ji}, \quad \forall (i, j) \in E \tag{3.8}$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, j \in V \tag{3.9}$$

This relation is clearly true: if node i immediately follows node j or node j immediately follows node i, then, the edge (i, j) must be used, and if neither happens, then the edge (i, j) is not being used.

The x_{ij} variables can also entirely replace the u_{ij} variables - if the u_{ij} variables are removed, constraints (3.2), (3.6) and (3.8) must be removed and replaced by the following:

$$x_{ij} + x_{ji} \leq 1, \quad \forall (i, j) \in E \tag{3.10}$$

$$\sum_{j \in V: j \neq i} x_{ij} = 1, \quad \forall i \in V \tag{3.11}$$

$$\sum_{j \in V: j \neq i} x_{ji} = 1, \quad \forall i \in V \tag{3.12}$$

(3.9)

In every other constraint presented in this dissertation and in the objective function, considering $(i, j) \in E$ and $i < j$, it is enough to replace each u_{ij} variable with $x_{ij} + x_{ji}$. Constraints (3.11) and (3.12) are often found on MILP formulations for the ATSP, and constraints (3.10) are a consequence of constraints (3.6) and (3.8). One observation regarding the replacement of variables u_{ij} with x_{ij} is in order - although constraints (3.10) should be included, these are always implied by other constraints presented later on, and will therefore never be included in the models tested in this work, since any of these models always includes constraints that imply these.

The nomenclature used throughout the remainder of this section reflects the fact that this adaptation consists of turning a problem defined on an undirected graph into a problem defined on a directed graph (in particular, the terms "cycle" and "edge" are replaced by "circuit" and "arc" respectively).

Other than the x_{ij} variables, the set of constraints presented in this subsection also uses non-negative integer variables $u_i \in \mathbb{N}$, $i \in V$ with the property that $u_j \geq u_i + 1$ when $x_{ij} = 1$ and $y_j = 0$, and $u_j = u_i + 1$ when $x_{ij} = 1$ and $y_i = y_j = 0$. These variables can be interpreted as indicating the position of a node in the circuit it is in (assuming the depot's position is 0). Consider the following inequalities:

$$u_j \geq u_i + 1 - M(1 - x_{ij} + y_j) + (M-2)x_{ji}, \quad \forall i, j \in V \tag{3.13}$$

$$1 - y_i \leq u_i \leq (1 - y_i)(M-1), \quad \forall i \in V \tag{3.14}$$

In inequalities (3.13) and (3.14), the value M equals $n - 3(p-1)$ (or $n - 2(p-1)$ if 2-circuits are allowed), which is the size of the largest circuit in any feasible solution. Inequalities (3.13) are an adaptation of the well-known DL inequalities presented in [2], guaranteeing that the position of two consecutive nodes in a circuit must increase by at least 1 unless the node that follows is a depot. The adaptation includes the y_j variable in order to guarantee this last condition. Inequalities (3.14) state that if a node acts as a depot, its position must equal 0, and otherwise, it must be some value between 1 and $M-1$. Although not strictly necessary, these constraints are used to possibly improve the LR of the model and simplify the interpretation of the u_i variables.

These inequalities, when added to the BASE model, eliminate solutions with more than p circuits. To see this, consider a solution with more than p circuits. At least one of these circuits does not have a depot, and thus, for all the arcs in that circuit, inequalities (3.13) become the standard DL inequalities. By using these inequalities in a circular fashion along the circuit, we obtain $u_j \geq u_i + 1$ for each arc (i, j) in the circuit, leading to a contradiction. An example of this can be seen in figure 3.2 - in this circuit, we assume no node is considered to be a depot. We notice we arrive at a contradiction if, for instance, we start from node 0 and keep considering the respective constraints for each arc in the circuit. Eventually, we find that $u_0 \geq u_0 + m$, which is obviously false, given m is a positive integer.

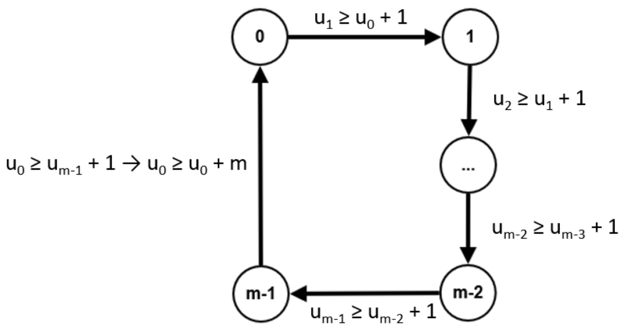

Figure 3.2: Circuit with no depots, violating some of the adapted DL constraints

Reusing the example from before, it is easy to see the introduction of a depot (for instance, node 0) changes this and allows this circuit to exist. This is illustrated in figure 3.3. Now, the lower bound on the value of u_0 remains 0 and no contradiction results from this.

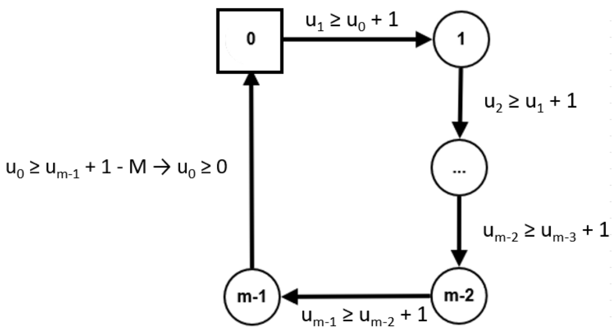

Figure 3.3: Circuit with one depot, respecting all of the adapted DL constraints

3.3 Preventing less than p Cycles - Node-Depot Assignment

The first formulation aimed at preventing solutions with less than p cycles (and which can therefore replace the generic constraints (3.5)) includes the integer variables $k_j \in \mathbb{N}, \forall j \in V$, which indicate the "label" of node j. In this case, we consider that the "label" of node j is given by the index of the node that plays the role of depot of the cycle node j belongs to (for example, if node j belongs to a cycle with depot d, then, $k_j = d$). These variables can be used to formulate a valid model to prevent solutions with less than p cycles, which may be achieved if the following constraints are added:

$$k_j \leq j - 1 + y_j, \qquad \forall j \in V \tag{3.15}$$

$$k_j \geq (j-1)y_j + 1, \qquad \forall j \in V \tag{3.16}$$

$$k_a \leq k_b + (a-1)(1-u_{ij}), \qquad \forall (i,j) \in E, \forall (a,b) \in \gamma(i,j) \tag{3.17}$$

$$1 \leq k_j \leq j, \qquad \forall j \in V \tag{3.18}$$

Inequalities (3.15) state that the label of node j must be less than or equal to $j-1$, unless j acts as the depot of the cycle it belongs to, in which case it may also equal j. Inequalities (3.16) state that the label of node j must be greater than or equal to 1, unless j acts as the depot of the cycle it belongs to, in which case it must be greater than or equal to j. Together, these constraints imply $k_j = j$ if and only if j acts as the depot of the cycle it belongs to. When $y_j = 0$, these two constraints guarantee that $1 \leq k_j \leq j - 1$ - this reflects the SB strategy mentioned when the BASE formulation was first presented (the only node which can act as the depot of a cycle is the node with the lowest index which belongs to that cycle). To model the HpMP using these variables without imposing that SB strategy, it seems necessary to find a way to state $k_j \neq j$ if $y_j = 0$, which is not obvious using linear constraints.

Inequalities (3.17) guarantee that the label of two adjacent nodes is the same (in other words, the depot of the cycle to which two adjacent nodes belong must be the same for both nodes). The coefficient "$(a-1)$" in these inequalities follows from the SB strategy - if this strategy was not considered, this coefficient would have to be increased to "$(|V|-1)$". Inequalities (3.18) simply define lower and upper bounds for the k_j variables - the upper bounds are also a consequence of the SB strategy. Observe that these k_j variables can be defined as continuous. This is a consequence of constraints (3.15) and (3.16) (which imply the label of each node d that acts as a depot must be exactly d, which is always an integer value) combined with constraints (3.17) (which, as shown below, imply the label of any two nodes in the same cycle must be the same). As previously mentioned, any feasible solution for this formulation considers that the only node which can act as the depot of a cycle is the node with the lowest index which belongs to it.

We show next that the formulation composed by BASE augmented with constraints (3.15), (3.16), (3.17) and (3.18) prevents solutions with less than p cycles. We do so by first observing that given a cycle with nodes $\{i_1, \ldots, i_m\}$, the following holds:

$$k_{i_j} = k_{i_k}, \quad \forall j, k \in \{1, \ldots, m\}$$

This is an immediate consequence of the fact that, given two adjacent nodes i_j, i_{j+1}, constraints (3.17) imply $k_{i_j} = k_{i_{j+1}}$, which in turn implies the equality above for the entire cycle.

To show that BASE augmented with constraints (3.15)-(3.18) prevents solutions with less than p cycles, we begin by considering a solution with less than p cycles. In such a solution, there will be at least one cycle with at least two depots (this follows from (3.3)) - for instance, nodes i, j such that $i \neq j$. Constraints (3.15) and (3.16) imply $k_i = i$ and $k_j = j$. However, this contradicts what was observed before - because since i and j are in the same cycle, $k_i = k_j$ should hold. This contradiction is a consequence of having multiple depots in a cycle, and therefore, this set of constraints prevents solutions with less than p cycles.

3.3.1 A Valid Inequality

Constraints (3.17) can be improved for $a < b$, and therefore, these can be replaced by:

$$k_i \leq k_j + (i-1)(1-u_{ji}), \qquad \forall (j,i) \in E \tag{3.19}$$

$$k_i \leq k_j + (i-1)(1-u_{ij}) - (j-1)y_j, \quad \forall (i,j) \in E \tag{3.20}$$

Constraints (3.19) were already included in constraints (3.17), but constraints (3.20) are new and clearly imply constraints (3.17) for $a < b$ (as demonstrated below). These may be interpreted as both stating that two adjacent nodes must be assigned to the same depot and that a node j which acts as a depot cannot be adjacent to any node i such that $i < j$. This happens because, if j is a depot and it is adjacent to some smaller node i, the corresponding constraint (3.20) for this pair becomes $k_i \leq 1$, which is false (recall that i and j are adjacent and, since j is a depot, $k_i = j$ should hold).

Proposition 1. *Constraints (3.20) are valid for the HpMP.*

Proof. Considering a pair of nodes i, j such that $i < j$, we begin by observing that $y_j = 0$ results in constraints already assumed to be valid, and therefore, it will always be assumed that $y_j = 1$. As previously mentioned, this implies $k_j = j$ (this is a consequence of constraints (3.15) and (3.16)). Considering the SB strategy, i and j cannot be adjacent if j is a depot, since that would imply j is the depot of a cycle containing nodes with indexes smaller than j. This means $u_{ij} = 0$. The constraint therefore becomes:

$$k_i \leq i$$

But this is simply the upper bound featured in constraints (3.18), meaning constraints (3.20) are valid if $y_j = 1$, and they were already valid when $y_j = 0$. Therefore, constraints (3.20) are valid. □

Lemma 1. *Constraints (3.7) and (3.20) imply constraints (3.17) for $a < b$.*

Proof. Consider constraint (3.20) for some edge $(i, j) \in E$. This constraint can be written as:

$$k_i + (j-1)y_j \leq k_j + (i-1)(1-u_{ij})$$

Since constraints (3.7) imply $y_j \geq 0$ and $j - 1 \geq 0$ also holds, the inequality above implies:

$$k_i \leq k_j + (i-1)(1-u_{ij})$$

Therefore, for any edge $(i, j) \in E$, the corresponding constraint (3.20), combined with constraints (3.7), implies the corresponding constraint (3.17) for $a < b$. Therefore, constraints (3.20) and (3.7) imply constraints (3.17) for $a < b$. □

It can also be proven that constraints (3.16) are implied by constraints (3.20):

Lemma 2. *Constraints (3.15), (3.18) and (3.20) imply constraints (3.16).*

Proof. To see this, consider some node j. For node j, the corresponding constraints (3.16) and (3.20) are:

$$k_j \geq (j-1)y_j + 1,$$
$$k_j \geq (j-1)y_j + k_i - (i-1)(1-u_{ij}), \quad \forall i \in V : i < j$$

Considering constraint (3.20) for node j and $i = 1$, and knowing $k_i = 1$ (which is implied by constraints (3.15) and (3.18)), we observe that it equals constraint (3.16) for node j. Therefore, constraints (3.16) are redundant if constraints (3.15), (3.18) and (3.20) are used. □

3.4 Preventing less than p Cycles - Node-Cycle Assignment

Consider now the additional integer variables $v_i \in \mathbb{N}, \forall i \in V$, indicating the "label" of node i. In these models, we consider the "label" of a node to be given by the cycle to which node i belongs (for instance, $v_i = k$ if and only if node i is in the k-th cycle). These variables can be used to create the following constraints, which prevent solutions with less than p cycles (and can therefore replace the generic constraints (3.5)):

$$v_a \leq v_b + (p-1)(1-u_{ij}), \qquad \forall (i,j) \in E, \forall (a,b) \in \gamma(i,j) \tag{3.21}$$
$$v_i \geq \sum_{j=1}^{i} y_j - (p-1)(1-y_i), \quad \forall i \in V \tag{3.22}$$
$$v_i \leq \sum_{j=1}^{i} y_j, \qquad \forall i \in V \tag{3.23}$$
$$1 \leq v_i \leq p, \qquad \forall i \in V \tag{3.24}$$

Constraints (3.21) indicate that the labels of adjacent nodes must have the same value (that is, adjacent nodes must belong to the same cycle). Constraints (3.22) and (3.23) relate the v_i variables with the y_i variables, respectively indicating that if a node is a depot, then the label of that node must be greater than or equal to the number of depots among all nodes before and including it, and that regardless of whether a node is a depot or not, the label of that node can never exceed the number of depots before and including it. Constraints (3.24) define lower and upper bounds for the v_i variables. Observe that these variables may be continuous (the reasoning for this is similar to that for the models presented in the previous section).

We show next that the formulation composed by BASE augmented with constraints (3.21), (3.22), (3.23) and (3.24) prevents solutions with less than p cycles. We do so by first observing that given a cycle with nodes $\{i_1, \ldots, i_m\}$, the following holds:

$$v_{i_j} = v_{i_k}, \quad \forall j, k \in \{1, \ldots, m\}$$

This is an immediate consequence of the fact that, given two adjacent nodes i_j, i_{j+1}, constraints (3.21) imply $v_{i_j} = v_{i_{j+1}}$, which in turn implies the equality above for the entire cycle.

To show that BASE augmented with constraints (3.21)-(3.24) prevents solutions with less than p cycles, we begin by considering a solution with less than p cycles. In such a solution, there will be at least one cycle with at least two depots (this follows from (3.3)) - for instance, nodes i, j such that $i > j$. Constraints (3.22) and (3.23) imply $v_i = \sum_{k=1}^{i} y_k \geq 1 + \sum_{k=1}^{j} y_k = 1 + v_j$. However, this contradicts what was observed before - because since i and j are in the same cycle, $v_i = v_j$ should hold. This contradiction is a consequence of having multiple depots in a cycle, and therefore, this set of constraints prevents solutions with less than p cycles.

Similarly to the model presented in the previous section, this model also considers, as an SB strategy, that the only node which may act as the depot of a cycle is the node with the lowest index which belongs to it. And also similarly to the model presented in section 3.3, it is not apparent how to model the problem using these variables without imposing this SB strategy. However, we also observe that, unlike the first model (in which the cycles are not numbered), this model requires the cycles to be numbered (it is always assumed these are numbered from 1 to p), which may result in multiple equivalent solutions (a given cycle may be the first, or the second or third cycle, for instance). In this case, as an SB strategy, it is also considered that all cycles are sorted by ascending order of the indices of their depots. It turns out that it is also not apparent how to prevent less than p cycles using the v_i variables without imposing this SB strategy.

Curiously, the argument used to show that the formulation composed by BASE augmented with constraints (3.21), (3.22), (3.23) and (3.24) prevents solutions with less than p cycles also shows why these SB strategies are implied by this model. Regarding the strategy which consists of considering the node with the lowest index in a given cycle to be the only possible depot for that cycle, recall that if two nodes belong to the same cycle, these are assigned the same label. Consider now some pair i, j such that $i > j$ and i acts as a depot - then, $v_i = \sum_{k=1}^{i} y_k \geq 1 + \sum_{k=1}^{j} y_k > v_j$ holds. But if $v_i > v_j$, these nodes cannot have the same label, and therefore, they cannot belong to the same cycle. Similarly, to show that this model implies the second SB strategy (which consists of sorting the cycles by ascending order of the indices of their depots), consider two depots i, j such that $i > j$. For these two depots, constraints (3.22) and (3.23) imply $v_i = \sum_{k=1}^{i} y_k \geq 1 + \sum_{k=1}^{j} y_k = 1 + v_j > v_j$. In other words, the label of a depot with index i must be larger than the label of a depot with some index j such that $j < i$, implying the second SB strategy.

3.4.1 A Valid Inequality

The SB strategy used in this model allows for the improvement of some of the inequalities shown in the previous subsection - in particular, constraints (3.21) may be replaced by the following:

$$v_i \leq v_j + (p-1)(1 - u_{ij}) - p y_j, \quad \forall (i, j) \in E \tag{3.25}$$

$$v_i \leq v_j + (p-1)(1 - u_{ji}), \quad \forall (j, i) \in E \tag{3.26}$$

Constraints (3.26) are clearly valid - they are a subset of constraints (3.21), which are valid. But the same is not so clear for constraints (3.25), which imply constraints (3.21) for $a < b$ (as demonstrated below).

Proposition 2. *Constraints (3.25) are valid for the HpMP.*

Proof. We begin by observing that, for a given pair of nodes i, j such that $i < j$, assuming $y_j = 0$ results in constraints already considered to be valid. We therefore assume $y_j = 1$. But if node j acts as a depot, the SB strategy states that it cannot be adjacent to any node i such that $i < j$, which means $u_{ij} = 0$. All this means that, in order to prove the validity of these constraints, it is only necessary to prove that the following inequality is valid under the assumptions mentioned previously:

$$v_i \leq v_j - 1$$

This is clearly valid. To see this, observe that under these assumptions ($y_j = 1$ and $u_{ij} = 0$), constraints (3.22) and (3.23) imply the following:

$$v_j = \sum_{l=1}^{j} y_l = \sum_{l=1}^{j-1} y_l + 1 \geq \sum_{l=1}^{i} y_l + 1 \geq v_i + 1$$

Therefore, assuming $y_j = 1$ and $u_{ij} = 0$ implies $v_i \leq v_j - 1$, and constraints (3.25) are valid. □

Lemma 3. *Constraints (3.7) and (3.25) imply constraints (3.21) for $a < b$.*

Proof. Consider constraint (3.25) for some edge $(i, j) \in E$. This constraint can be written as:

$$v_i + py_j \leq v_j + (p - 1)(1 - u_{ij})$$

Since constraints (3.7) imply $y_j \geq 0$ and $p - 1 \geq 0$ also holds, the inequality above implies:

$$v_i \leq v_j + (p - 1)(1 - u_{ij})$$

Therefore, for any edge $(i, j) \in E$, the corresponding constraint (3.25), combined with constraints (3.7), implies the corresponding constraint (3.21) for $a < b$. Therefore, constraints (3.25) and (3.7) imply constraints (3.21) for $a < b$. □

It can also be proven that constraints (3.25) and (3.26) are sufficient to prevent feasible solutions with less than p cycles. In other words, when these constraints are used, constraints (3.22) and (3.23) are no longer necessary to model the problem. To prove this, we first observe that similarly to what happens with the model using constraints (3.21), given a cycle with nodes $\{i_1, \ldots, i_m\}$, the following holds:

$$v_{i_j} = v_{i_k}, \quad \forall j, k \in \{1, \ldots, m\}$$

To see this, observe that, given two adjacent nodes i_j, i_{j+1}, constraints (3.25) and (3.26) imply respectively (it is assumed here that $i_j > i_{j+1}$, but it is simple to see that the same can be done for $i_j < i_{j+1}$):

$$v_{i_j} \geq v_{i_{j+1}} + py_{i_j}$$
$$v_{i_{j+1}} \geq v_{i_j}$$

Since $v_{i_{j+1}} \geq v_{i_j}$, we can see that $v_{i_j} \geq v_{i_j} + py_{i_j}$, and therefore, these inequalities imply $y_{i_j} = 0$ and also imply $v_{i_j} = v_{i_{j+1}}$. Since this equality holds for any pair of adjacent nodes, we can conclude that it also holds for any pair of nodes belonging to the same cycle.

To show that constraints (3.25) and (3.26) alone prevent solutions with less than p cycles, we begin by considering a solution with less than p cycles. In such a solution, there will be at least one cycle with at least two depots (this follows from (3.3)) - for instance, nodes i, j such that $i < j$. The corresponding constraint (3.25) for this pair implies that $v_i \leq v_j + (p - 1)(1 - u_{ij}) - py_j$, and since $y_j = 1$, this inequality implies $v_i \leq v_j - 1$. However, this contradicts what was observed before - because since i

and j are in the same cycle, $v_i = v_j$ should hold. This contradiction is a consequence of having multiple depots in a cycle, and therefore, constraints (3.25) and (3.26) prevent solutions with less than p cycles.

3.4.2 Improving Constraints

For the following improvements, it is important to observe that constraints (3.7) and (3.23) imply $v_i \leq i$. We now resume with the improvements, modifying constraints (3.21):

$$v_a \leq v_b + \min\{p-1, a-1\}(1-u_{ij}), \quad \forall (i,j) \in E, \forall (a,b) \in \gamma(i,j) \tag{3.27}$$

Constraints (3.21) and (3.27) are very similar - the only difference is the term $(p-1)$ in (3.21) is replaced by $\min\{p-1, a-1\}$ in (3.27).

Proposition 3. *Constraints (3.27) are valid for the HpMP.*

Proof. In this proof, it may be assumed that $a < p$, since $a \geq p$ turns these constraints into constraints already assumed to be valid. It may also be considered that $u_{ij} = 0$, since u_{ij} is a binary variable and $u_{ij} = 1$ makes the proposed modification irrelevant, since the only modified constant is multiplied by 0. Therefore, it is only necessary to prove the following constraint is valid under all these assumptions:

$$v_a \leq v_b + a - 1$$

But $v_b \geq 1$, and therefore, $v_b + a - 1 \geq a \geq v_a$, proving the validity of $v_a \leq v_b + a - 1$ under these assumptions, and therefore, the validity of constraints (3.27). □

Lemma 4. *Constraints (3.6) and (3.27) imply constraints (3.21).*

Proof. Given any edge $(i,j) \in E$ and pair $(a,b) \in \gamma(i,j)$, constraints (3.27) and (3.21) are equivalent if $a \geq p$. Therefore, it is now only necessary to prove that constraints (3.27) and (3.6) imply constraints (3.21) if $a < p$. We begin by considering some edge $(i,j) \in E$ and the corresponding constraints (3.6) and a constraint (3.27) such that $a < p$. Since $a < p$, the latter constraint becomes:

$$v_a \leq v_b + (a-1)(1-u_{ij})$$

Constraints (3.6) imply $0 \leq 1 - u_{ij}$. If this inequality is added $(p-a)$ times to the corresponding constraint (3.27) (recall that $p - a$ is a positive integer), the following inequality is obtained:

$$v_a + (p-a)0 \leq v_b + (a-1)(1-u_{ij}) + (p-a)(1-u_{ij}) \Leftrightarrow$$

$$\Leftrightarrow v_a \leq v_b + (p-1)(1-u_{ij})$$

Therefore, since constraints (3.27) are equivalent to constraints (3.21) for $a \geq p$ and, combined with constraints (3.6), also imply constraints (3.21) for $a < p$, this means constraints (3.27) and (3.6) imply constraints (3.21). □

Likewise, constraints (3.22) can also be modified:

$$v_i \geq \sum_{j=1}^{i} y_j - \min\{p-1, i-2\}(1-y_i), \quad \forall i \in V \tag{3.28}$$

Proposition 4. *Constraints (3.28) are valid for the HpMP.*

Proof. In this proof, it is considered that $i-2 < p-1$, since considering otherwise turns these constraints into constraints already assumed to be valid. It may also be considered that $y_i = 0$, since considering $y_i = 1$, once again, results in inequalities already known to be valid under that assumption. To prove the validity of constraints (3.28), then, it is only necessary to prove the validity of the following inequality under all these assumptions:

$$v_i \geq \sum_{j=1}^{i-1} y_j - (i-2)$$

But since $y_j \leq 1, \forall j \in V$, this inequality is implied by $v_i \geq 1$, which is already assumed to be valid. Therefore, constraints (3.28) are valid. □

Lemma 5. *Constraints (3.7) and (3.28) imply constraints (3.22).*

Proof. Given any node $i \in V$, constraints (3.28) and (3.22) are equivalent if $i-2 \geq p-1$, and therefore, it is only necessary to prove that constraints (3.28) and (3.7) imply constraints (3.22) if $i-2 < p-1$.

We begin by considering some node $i \in V$ such that $i-2 < p-1$. Since $i-2 < p-1$, the corresponding constraint (3.28) becomes:

$$v_i \geq \sum_{j=1}^{i} y_j - (i-2)(1-y_i)$$

Constraints (3.7) imply $0 \geq -(1-y_i)$. If this inequality is added $(p-i+1)$ times to the corresponding constraint (3.28) (recall that $(p-i+1)$ is a positive integer), the following inequality is obtained:

$$v_i + (p-i+1)0 \geq \sum_{j=1}^{i} y_j - (i-2)(1-y_i) - (p-i+1)(1-y_i) \Leftrightarrow$$

$$\Leftrightarrow v_i \geq \sum_{j=1}^{i} y_j - (p-1)(1-y_i)$$

Therefore, since constraints (3.28) are equivalent to constraints (3.22) for $i-2 \geq p-1$ and, combined with constraints (3.7), also imply constraints (3.22) for $i-2 < p-1$, this means constraints (3.28) and (3.7) imply constraints (3.22). □

And finally, very similarly to what was done for constraints (3.27), constraints (3.25) can also be improved as follows:

$$v_i \leq v_j + \min\{p-1, i-1\}(1-u_{ij}) - \min\{p, i\}y_j, \quad \forall (i,j) \in E \tag{3.29}$$

Proposition 5. *Constraints (3.29) are valid for the HpMP.*

Proof. In this proof, it is considered that $i < p$, since considering otherwise would result in constraints already known to be valid. It is also assumed that $y_j = 1$, since considering $y_j = 0$ turns these constraints into (3.27), which are valid. According to the SB strategy employed in these models, if $y_j = 1$ and $i < j$, then, i and j must not belong to the same cycle (otherwise the depot of the cycle to which j belongs would be some node with an index which is less than or equal to i). If i and j are not in the same cycle, they are not adjacent, and therefore, $u_{ij} = 0$. Therefore, to prove the validity of these constraints, all that is necessary is to prove, under all these assumptions, the validity of the following inequality:

$$v_i \leq v_j - 1$$

Under these assumptions this inequality has already been shown to be valid in the proof of proposition 2. Therefore, constraints (3.29) are valid for the HpMP. □

It must be observed that, unlike what happened with constraints (3.27) and (3.28), constraints (3.29) do not seem to imply constraints (3.25). However, these imply constraints (3.27):

Lemma 6. *Constraints* (3.7) *and* (3.29) *imply constraints* (3.27).

Proof. Consider constraint (3.29) for some edge $(i, j) \in E$. This constraint can be written as:

$$v_i + \min\{p, i\} y_j \leq v_j + \min\{p - 1, i - 1\}(1 - u_{ij})$$

Since constraints (3.7) imply $y_j \geq 0$ and $\min\{p, i\} \geq 0$ also holds, the inequality above implies:

$$v_i \leq v_j + \min\{p - 1, i - 1\}(1 - u_{ij})$$

Therefore, for any edge $(i, j) \in E$, the corresponding constraint (3.29), combined with constraints (3.7), implies the corresponding constraint (3.27). Therefore, constraints (3.29) and (3.7) imply constraints (3.27). □

3.5 An Overview of the Models

We can now build models using the variables and constraints presented in the previous pages. Given the focus of this work is on constraints used to prevent solutions with less than p cycles, we consider both models which include constraints aimed at preventing solutions with more than p cycles (referred to as "complete models") and models which do not feature those constraints (referred to as "incomplete models"). Although this means the feasible solutions for the latter models feature p or more cycles, and not only p cycles, this methodology does not seem inadequate, as the purpose of this dissertation is to compare the constraints used to prevent solutions with less than p cycles, which does not seem reliant on the addition of the adapted DL constraints (or any other set of constraints which serves the same purpose).

The tables below feature every model built with variables and constraints presented throughout this dissertation. Each row corresponds to a different model, and the tables include two columns. The first column includes the name of the model, and the second column includes the constraints that comprise the corresponding model. Observe that since every model includes the depot variables y_i and constraints (3.3) and (3.7), these constraints are not included in the table. It is also not necessary to explicitly state which variables are included in each model, since that is implicit in the constraints.

In these models, "D" stands for "**D**esrochers and Laporte", "NC" stands for "**N**ode-**C**ycle Assignment", "SNC" stands for "**S**trengthened **NC**", "ND" stands for "**N**ode-**D**epot Assignment", "SND" stands for "**S**trengthened **ND**" and "(SB)" stands for "(**S**ymmetry **B**reaking)".

Models featuring "ND" include the k_i variables and some of the constraints presented in section 3.3, and models featuring "NC" include the v_i variables and some of the constraints presented in section 3.4.

Model	Constraints
DNC	(3.11); (3.12); (3.13); (3.21); (3.22); (3.23); (3.9); (3.14); (3.24)
SDNC-	(3.11); (3.12); (3.13); (3.25); (3.26); (3.9); (3.14); (3.24)
SDNC	(3.11); (3.12); (3.13); (3.25); (3.26); (3.22); (3.23); (3.9); (3.14); (3.24)
DNC(SB)	(3.11); (3.12); (3.13); (3.27); (3.28); (3.23); (3.9); (3.14); (3.24)
SDNC-(SB)	(3.11); (3.12); (3.13); (3.27) (for $a > b$); (3.29); (3.9); (3.14); (3.24)
SDNC(SB)	(3.11); (3.12); (3.13); (3.27) (for $a > b$); (3.29); (3.28); (3.23); (3.9); (3.14); (3.24)
DND	(3.11); (3.12); (3.13); (3.15); (3.16); (3.17); (3.9); (3.14); (3.18)
SDND	(3.11); (3.12); (3.13); (3.15); (3.19); (3.20); (3.9); (3.14); (3.18)

Table 3.1: Constraints used in each complete model

Model	Constraints
NC	(3.2); (3.21); (3.22); (3.23); (3.6); (3.24)
SNC-	(3.2); (3.25); (3.26); (3.6); (3.24)
SNC	(3.2); (3.25); (3.26); (3.22); (3.23); (3.6); (3.24)
NC(SB)	(3.2); (3.27); (3.28); (3.23); (3.6); (3.24)
SNC-(SB)	(3.2); (3.27) (for $a > b$); (3.29); (3.6); (3.24)
SNC(SB)	(3.2); (3.27) (for $a > b$); (3.29); (3.28); (3.23); (3.6); (3.24)
ND	(3.2); (3.15); (3.16); (3.17); (3.6); (3.18)
SND	(3.2); (3.15); (3.19); (3.20); (3.6); (3.18)

Table 3.2: Constraints used in each incomplete model

If a model is given a name which includes "D" without an "N" before it (for instance, "**D**NC" or "S**D**ND"), the u_{ij} variables have been replaced by the x_{ij} variables, with some constraints also being replaced as stated in section 3.2. Compared to their counterparts without "D", these models also include the u_i variables and constraints (3.13) and (3.14). The difference between a model with "D" and a model without "D", therefore, is that the former only considers feasible solutions with exactly p cycles, whereas the latter also considers feasible solutions with more than p cycles.

Regarding the "Strengthened" models (those with names starting with "S"), if they use variables which assign nodes to cycles, these replace constraints (3.21) (or (3.27) if the name of the model ends with "(SB)") for $i < j$ with constraints (3.25) (or (3.29) if the name of the model ends with "(SB)"). Additionally, if the "Strengthened" models use variables which assign nodes to depots, constraints (3.16) and (3.17) are replaced by (3.19) and (3.20).

As the listed constraints show, the only differences between a model and its "(SB)" counterpart are the replacement of constraints (3.21), (3.22) and (3.25) with constraints (3.27), (3.28) and (3.29) where applicable. The naming, therefore, reflects the fact that these modifications are a consequence of the SB strategy.

As mentioned after the introduction of the improved constraints featured in the "Strengthened" models in which nodes are assigned to cycles, a model featuring constraints (3.25) and (3.26) no longer requires constraints (3.22) and (3.23) to prevent solutions with less than p cycles. The difference between an "NC" model and the corresponding model with a "-" in the name, then, is that the model with "-" does not include (3.22) (or (3.28), if the name of the model includes "(SB)") and (3.23), whereas the original model does.

We can now formally relate some of the models presented in this dissertation. While the lemmas presented in the previous sections relating different constraints were only presented for the inequalities found in the incomplete models, it is simple to see these also apply to the inequalities found in the complete models (where the u_{ij} variables have been replaced by the x_{ij} variables as instructed in section 3.2), and therefore, the following propositions also apply to the corresponding complete models. This happens because, for each implication in which it is necessary to consider $u_{ij} \leq 1$ (and therefore, in the complete models, $x_{ij} + x_{ji} \leq 1$), the complete models said to implicate other complete models always include inequalities which imply $x_{ij} + x_{ji} \leq 1, \forall (i, j) \in E$, and therefore, the implications are also valid for the corresponding complete models.

Proposition 6. *The LR bound obtained with SNC is always greater than or equal to the LR bound obtained with NC.*

Proof. We prove this by showing every constraint found in NC is implied by constraints found in SNC. If this implication holds, that implies that every feasible solution for the LR of SNC is also feasible for the LR of NC, implying the LR bound of SNC is always greater than or equal to the LR bound of NC.

Every constraint in NC other than constraints (3.21) is also found in SNC. Therefore, it is only necessary to prove the constraints found in SNC imply constraints (3.21). Constraints (3.21) for $b < a$ are featured in SNC (they are constraints (3.26)) and constraints (3.21) for $a < b$ are implied by constraints (3.25) and (3.7), which are both also featured in SNC. Therefore, constraints (3.21) are implied by constraints found in SNC, and thus, the LR bound obtained with SNC is always greater than or equal to the LR bound obtained with NC. □

Proposition 7. *The LR bound obtained with SNC is always greater than or equal to the LR bound obtained with SNC-.*

Proof. Every constraint found in SNC- can also be found in SNC. This means every solution which is feasible for the LR of SNC is also feasible for the LR of SNC-, which means the LR bound obtained with SNC is always greater than or equal to the LR bound obtained with SNC-. □

Proposition 8. *The LR bound obtained with SNC(SB) is always greater than or equal to the LR bound obtained with NC(SB).*

Proof. We prove this by showing every constraint found in NC(SB) is implied by constraints found in SNC(SB). If this implication holds, that implies that every feasible solution for the LR of SNC(SB) is also feasible for the LR of NC(SB), implying the LR bound of SNC(SB) is always greater than or equal to the LR bound of NC(SB).

Every constraint in NC(SB) other than constraints (3.27) is also found in SNC(SB). Therefore, it is only necessary to prove the constraints found in SNC(SB) imply constraints (3.27). Constraints (3.27) for $b < a$ are featured in SNC(SB) and constraints (3.27) for $a < b$ are implied by constraints (3.29) and (3.7), which are both also featured in SNC(SB). Therefore, constraints (3.27) are implied by constraints found in SNC(SB), and thus, the LR bound obtained with SNC(SB) is always greater than or equal to the LR bound obtained with NC(SB). □

Proposition 9. *The LR bound obtained with SNC(SB) is always greater than or equal to the LR bound obtained with SNC-(SB).*

Proof. Every constraint found in SNC-(SB) can also be found in SNC(SB). This means every solution which is feasible for the LR of SNC(SB) is also feasible for the LR of SNC-(SB), which means the LR bound obtained with SNC(SB) is always greater than or equal to the LR bound obtained with SNC-(SB). □

Proposition 10. *The LR bound obtained with NC(SB) is always greater than or equal to the LR bound obtained with NC.*

Proof. We prove this by showing every constraint found in NC is implied by constraints found in NC(SB). If this implication holds, that implies that every feasible solution for the LR of NC(SB) is also feasible for the LR of NC, implying the LR bound of NC(SB) is always greater than or equal to the LR bound of NC.

Every constraint in NC other than constraints (3.21) and (3.22) is also found in NC(SB). Therefore, it is only necessary to prove the constraints found in NC(SB) imply constraints (3.21) and (3.22). Constraints (3.27) and (3.6) imply constraints (3.21), whereas constraints (3.28) and (3.7) imply constraints (3.22). In conclusion, every constraint found in NC is implied by constraints found in NC(SB), which means the LR bound obtained with NC(SB) is always greater than or equal to the LR bound obtained with NC. □

Proposition 11. *The LR bound obtained with SND is always greater than or equal to the LR bound obtained with ND.*

Proof. We prove this by showing every constraint found in ND is implied by constraints found in SND. If this implication holds, that implies that every feasible solution for the LR of SND is also feasible for the LR of ND, implying the LR bound of SND is always greater than or equal to the LR bound of ND.

Every constraint in ND other than constraints (3.16) and (3.17) is also found in SND. Therefore, it is only necessary to prove the constraints found in SND imply constraints (3.16) and (3.17). Constraints (3.16) are implied by constraints (3.15), (3.18) and (3.20), whereas constraints (3.17) for $b < a$ are the same as constraints (3.19) and constraints (3.17) for $a < b$ are implied by constraints (3.7) and (3.20). Therefore, every constraint found in ND is implied by constraints found in SND, which implies the LR bound obtained with SND is always greater than or equal to the LR bound obtained with ND. □

Chapter 4

Computational Experiment

In this chapter we conduct a computational experiment with the goal of evaluating the performance and comparing all of the compact formulations for the HpMP listed in section 3.5.

4.1 Hardware / Software Configurations and Test Instances

All tests were run on a computer with an Intel® Core™ i7-4790 CPU, 8GB of DDR3-1600 RAM running Windows 10 Pro, Version 21H2, within which CPLEX 20.1.0 Concert Technology for C++ was used.

The models are tested on a set of instances available for free from TSPLIB (http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/) - these are gr24 (24 nodes), fri26 (26 nodes), bayg29 (29 nodes), swiss42 (42 nodes), **att48** (48 nodes), gr48 (48 nodes), hk48 (48 nodes), **eil51** (51 nodes), **berlin52** (52 nodes), brazil58 (58 nodes), **st70** (70 nodes), **eil76** (76 nodes), **pr76** (76 nodes), **rat99** (99 nodes), **kroA100** (100 nodes), **kroB100** (100 nodes), **kroC100** (100 nodes), **kroD100** (100 nodes), **kroE100** (100 nodes), **rd100** (100 nodes). The instances whose names are bold had their edge weight functions changed from the original functions to the euclidian distance (without rounding to integer values) - this was done to remain consistent with recent literature (for instance, [12] and [14]).

Each one of the models presented in section 3.5 is tested in each one of these instances and for different values of p. The complete models are tested for five values of p for every instance (assuming n is the number of nodes in the graph, the values of p for which the tests were carried out are $p_1 = \lfloor \frac{n}{10} \rfloor$, $p_2 = \lfloor \frac{n}{7} \rfloor$, $p_3 = \lfloor \frac{n}{5} \rfloor$, $p_4 = \lfloor \frac{n}{4} \rfloor$ and $p_5 = \lfloor \frac{n}{3} \rfloor$), and the incomplete models are tested for three values of p for every instance (assuming n is the number of nodes in the graph, the values of p for which the tests were carried out are $p_1^* = \lfloor \frac{n}{4} \rfloor$, $p_2^* = \lceil \frac{p_1^*+p_3^*}{2} \rceil$ and $p_3^* = \lfloor \frac{n}{3} \rfloor$). Every test has a time limit of one hour - if the instance is not solved within the first hour, an optimal value is not obtained, being instead replaced by an interval to which the optimal value is guaranteed to belong.

4.2 Test Results

Before presenting the test results, we recall that the LR gap of a model can be calculated as $GAP = 1 - \frac{LR}{OPT}$, where OPT is the optimal value of the instance and LR is the LR bound of the model for that instance.

Each one of the tables with test results (which may be found in appendix A) has 8 columns. From left to right, they indicate the number of cycles of each feasible solution ("p"), the optimal value of the corresponding instance ("OPT"), the model which was tested ("Model"), the LR bound ("LR"), the lower bound on the optimal value ("LB"), the upper bound on the optimal value ("UB"), the computational time (in seconds) ("Time (s)") and the LR gap ("GAP (%)"). The computational times to determine the LR bounds are not included as these were always negligible.

In each one of these tables, a set of test results is also preceded by a row with an instance name (meaning the test results that follow were obtained for that instance). Each instance name is also accompanied by a constant (denoted by p'), which represents the value of p for which the optimal value is lowest for the corresponding graph (in other words, the optimal solution for the problem considering only constraints (3.2) and (3.6)).

Regarding the OPT values, an observation is in order - for some of the instances, no optimal value was known at the time of writing this dissertation. Therefore, some of the OPT values are simply the value of the best solution found in either [12] or [14] without proof of optimality (if the OPT value is followed by "*").

Before resuming, we recall that the models studied here are compared considering two metrics - the quality of their LR bounds (higher is better) and the computational times necessary to solve the problems (lower is better). If, when comparing two models, CPLEX was unable to solve an instance with either model within the time limit, the models are instead compared considering the proximity of the corresponding LB and UB values (closer values are better).

Considering this, given an instance, when comparing LR gaps, some model is said to perform better than some other model if the LR gap is lower (in other words, if the LR bound is larger). When comparing computational times, some model is said to perform better than some other model if the corresponding computational time is at least 0.1 seconds lower. If, for both models, CPLEX was unable to solve the problem within one hour (in other words, if the computational time is 3600 seconds for both models), some model A is said to perform better than some other model B if $UB_A - LB_A \leq UB_B - LB_B - 0.1$. These small tolerances when comparing computational times mean that, in some instances, there can be draws between pairs of models. When comparing two models it is also considered that if CPLEX ran out of memory when solving the problem using one of the models but not the other, the model for which CPLEX did not run out of memory performed better. If CPLEX ran out of memory with both models, it is considered a draw.

We begin by analysing the test results obtained for every complete model (these results may be found in appendix A), and for those, we start with the LR gaps - the following table features the average LR gaps for every complete model presented in table 3.1 for each value of p:

	DNC	SDNC-	SDNC	DNC(SB)	SDNC-(SB)	SDNC(SB)	DND	SDND
p_1	1.52%	1.52%	1.52%	1.52%	1.52%	1.52%	1.52%	1.52%
p_2	1.18%	1.18%	1.18%	1.18%	1.18%	1.18%	1.18%	1.18%
p_3	1.85%	1.85%	1.85%	1.85%	1.85%	1.84%	1.85%	1.84%
p_4	3.16%	3.16%	3.15%	3.16%	3.16%	3.15%	3.16%	3.13%
p_5	9.62%	9.55%	9.48%	9.62%	9.53%	9.46%	9.61%	9.33%

Table 4.1: Average gaps for complete models for every value of p

As this table shows, the differences between the LR gaps of all the models are small. The differences between the gaps are greater for larger values of p, and even for $p_5 = \lfloor \frac{n}{3} \rfloor$, the largest difference between gap averages is 0.29% (between DNC and SDND). It must also be observed that, while the LR bounds improve as p grows, this improvement is not very substantial. For some models, the LR bounds always remain the same regardless of p (this seems to apply to DNC and DNC(SB)), while for every other model, the LR bounds only improve slightly as p grows closer to $\lfloor \frac{n}{3} \rfloor$ (and sometimes do not improve at all).

Interestingly, the "(SB)" models have very similar gaps to their counterparts without "improved" constraints, with very negligible improvements even for p_5. One observation is in order - in some instances (for example, rat99 for $p = 33$), some "(SB)" models actually have worse LR gaps than their corresponding models without "(SB)". This can only happen for SDNC- and SDNC, and is to be expected, since the "(SB)" counterparts of these models are obtained by replacing, among others, constraints (3.25) with (3.29), and these two constraints cannot be related, unlike most other constraints found in the "(SB)" models which imply the constraints they replace.

The addition of constraints (3.25) (or (3.29)) also does not seem to result in large improvements to the LR bounds. However, models including these constraints and not including constraints (3.22) (or (3.28)) and (3.23) seem to have slightly better LR bounds than models which include constraints (3.22) (or (3.28)) and (3.23) but do not include constraints (3.25) (or (3.29)).

We now proceed to analyse the performance of these models in regards to computational times. The following tables are relevant for this analysis: the first table (4.2) indicates, for each pair of complete models, the number of times the first model performed better than the second, while table 4.3 indicates the number of tests CPLEX was unable to solve within the time limit for each complete model and each value of p. Finally, table 4.4 features the average times for tests in which CPLEX did not reach the time limit with any complete model.

	DNC	SDNC-	SDNC	DNC(SB)	SDNC-(SB)	SDNC(SB)	DND	SDND
DNC	N/A	10	4	44	8	9	6	3
SDNC-	83	N/A	33	84	52	46	22	18
SDNC	90	55	N/A	90	56	51	26	23
DNC(SB)	48	12	7	N/A	8	8	7	4
SDNC-(SB)	88	40	34	86	N/A	38	17	11
SDNC(SB)	88	47	39	88	50	N/A	24	11
DND	87	66	67	86	76	68	N/A	34
SDND	92	71	69	93	81	77	52	N/A

Table 4.2: Number of times the model in the row performed better than the model in the column (for complete models)

	DNC	SDNC-	SDNC	DNC(SB)	SDNC-(SB)	SDNC(SB)	DND	SDND
p_1	4	3	3	4	3	4	4	2
p_2	4	2	1	2	2	1	2	1
p_3	9	8	7	9	8	8	5	5
p_4	11	9	9	11	10	9	7	7
p_5	16	16	16	15	15	15	15	15

Table 4.3: Number of times the time limit was reached (for complete models)

	DNC	SDNC-	SDNC	DNC(SB)	SDNC-(SB)	SDNC(SB)	DND	SDND
p_1	270.9	205	148.7	205.5	147.2	123.5	101.8	89.9
p_2	370.5	96.4	65	311.6	98.4	37.3	104.2	30.4
p_3	144.1	77.9	150.3	180.9	74.3	71.6	19.9	17.9
p_4	361	46.7	47.8	919.9	66.5	61.8	15.4	14.3
p_5	38.6	8.3	5.5	107.7	38.6	25.7	10.1	9.7

Table 4.4: Average times for tests in which CPLEX never reached the time limit (for complete models)

As far as computational times are concerned, none of the complete models presented in this work seems to perform well for large instances and large values of p - most models reached the one hour time limit for most tests featuring large instances, especially for large values of p. Regardless, some perform much better than others.

Comparing these models with each other, we can begin by observing something which is made evident by tables 4.2, 4.3 and 4.4 - models DNC and DNC(SB) are clearly the worst performers of the group, often performing worse than most other models. Not only do they result in the worst LR bounds, then, they also result in the worst computational times out of all the models.

We also observe that the "(SB)" models do not consistently perform better than their original counterparts. DNC performs much better than DNC(SB) in instances such as att48 (for $p = 12$) and DNC(SB) performs much better than DNC in instances such as hk48 (for $p = 16$). The conclusions are similar when comparing SDNC- with SDNC-(SB) and SDNC with SDNC(SB).

Comparing SDNC and SDNC- shows the former performs slightly better than the latter, although SDNC- outperforms SDNC in some cases (for example, gr48 (for $p = 9$) and hk48 (for $p = 12$)). A similar conclusion is arrived at when comparing SDNC-(SB) and SDNC(SB).

Considering DND and SDND, we conclude that these models seem to perform the best out of all complete models. When compared with one another, SDND seems to perform somewhat better, although DND also performs better than SDND in many instances, sometimes having a substantial lead over its "strengthened" counterpart (for example, rd100 (for $p = 20$)).

In conclusion, SDND appears to be the best model, followed by DND and SDNC. DNC and DNC(SB), on the other hand, perform the worst out of all the models.

We now proceed to analyse the performance of the incomplete models (the results are found in appendix A). We begin, once again, by analysing the LR gaps - the following table features the average LR gaps for every incomplete model presented here and for every value of p.

	NC	SNC-	SNC	NC(SB)	SNC-(SB)	SNC(SB)	ND	SND
p_1^*	3.16%	3.16%	3.15%	3.16%	3.16%	3.15%	3.16%	3.13%
p_2^*	4.86%	4.83%	4.81%	4.86%	4.83%	4.80%	4.86%	4.75%
p_3^*	9.62%	9.55%	9.48%	9.62%	9.53%	9.46%	9.61%	9.33%

Table 4.5: Average gaps for incomplete models for every value of p

In regards to LR gaps, the results are very similar to the results obtained when testing the complete models. There seem to be no additional observations to be made.

Similarly to what was done for the complete models, we analyse the computational times obtained with the incomplete models with the aid of tables 4.6, 4.7 and 4.8. Table 4.6 indicates, for each pair of incomplete models, the number of times the first model performed better than the second. Table 4.7 indicates the number of tests CPLEX was unable to solve within the time limit for each incomplete model and each value of p. Finally, table 4.8 features the average times for tests in which CPLEX did not reach the time limit with any incomplete model.

As far as computational times are concerned, although they perform much better than their complete counterparts, none of the incomplete models presented in this work still seem to perform well for the largest instances and the largest values of p - most models reached the one hour time limit for most tests featuring large instances and very large values of p. However, the incomplete models must still be compared with each other.

	NC	SNC-	SNC	NC(SB)	SNC-(SB)	SNC(SB)	ND	SND
NC	N/A	13	8	30	25	11	3	4
SNC-	44	N/A	23	41	39	25	4	7
SNC	49	32	N/A	47	41	31	8	9
NC(SB)	28	16	10	N/A	26	16	6	4
SNC-(SB)	33	17	15	32	N/A	20	3	1
SNC(SB)	48	30	23	40	35	N/A	9	4
ND	55	53	49	52	50	47	N/A	28
SND	55	50	48	54	54	51	26	N/A

Table 4.6: Number of times the model in the row performed better than the model in the column (for incomplete models)

	NC	SNC-	SNC	NC(SB)	SNC-(SB)	SNC(SB)	ND	SND
p_1^*	3	1	2	2	3	3	1	0
p_2^*	8	7	4	8	6	7	2	3
p_3^*	13	11	11	12	14	12	11	11

Table 4.7: Number of times the time limit was reached (for incomplete models)

	NC	SNC-	SNC	NC(SB)	SNC-(SB)	SNC(SB)	ND	SND
p_1^*	328.8	286.9	132.8	322.5	291.3	95.1	9.6	7.9
p_2^*	149.1	66.2	83.7	190.0	270.9	87.2	18.3	19.6
p_3^*	272.8	88.7	64.1	287	165.1	196.4	82	101.5

Table 4.8: Average times for tests in which CPLEX never reached the time limit (for incomplete models)

Similarly to what happened for the complete models, we observe that models NC and NC(SB) are the worst performers of the incomplete models, often performing worse than most other models. Interestingly, though, NC and NC(SB) are not the only poor performers among the incomplete models. Although CPLEX ran out of memory once with the NC(SB) model, it also ran out of memory in nine different tests with the SNC-(SB) model. This makes SNC-(SB) the second model with which CPLEX ran out of memory or reached the time limit the most, only behind NC.

We also observe that the “(SB)” models do not consistently perform better than their original counterparts, and this seems to depend on the instance (for instance, NC performs much better than NC(SB) in instances such as rat99 (for $p = 24$) and NC(SB) performs much better than NC in instances such as kroD100 (for $p = 25$)). Curiously, the “(SB)” models seem to perform slightly worse on average than their original counterparts.

Comparing SNC and SNC- shows that, similarly to what happened when comparing SDNC and SDNC-, the former performs slightly better than the latter, although SNC- outperforms SNC considerably in some cases (for example, pr76 (for $p = 25$) and kroE100 (for $p = 25$)). A similar conclusion is arrived at when comparing SNC-(SB) and SNC(SB).

Considering ND and SND, we conclude that these models perform the best out of all incomplete models, and unlike what happened with the complete models (where SDND seemed to have a slight edge over DND), it seems hard to pick one model as the best.

In conclusion, ND and SND appear to be the best models, followed by SNC. NC, NC(SB) and SNC-(SB), on the other hand, perform the worst out of all the models.

We finish this chapter by observing that the values of the LR bounds allow us to state that the LR bounds of some models cannot be related. These observations are only made for the incomplete models, but they also apply to the corresponding complete models.

We begin with the LR bounds of SNC- and SNC-(SB), which cannot be related, since there are instances for which SNC-(SB) has a strictly better LR bound than SNC- (such as eil51 for $p = 17$) and there are instances for which SNC- has a strictly better LR bound than SNC-(SB) (such as rat99 for $p = 33$).

The LR bounds of SNC- and ND also cannot be related, since there are instances for which ND has a strictly better LR bound than SNC- (such as eil51 for $p = 17$) and there are instances for which SNC- has a strictly better LR bound than ND (such as berlin52 for $p = 17$).

Additionally, the LR bounds of SNC-(SB) and ND cannot be related, since there are instances for which ND has a strictly better LR bound than SNC-(SB) (such as eil51 for $p = 17$) and there are instances for which SNC-(SB) has a strictly better LR bound than ND (such as berlin52 for $p = 17$).

Finally, the LR bounds of SNC and SNC(SB) cannot be related as there are instances for which SNC(SB) has a strictly better LR bound than SNC (such as eil51 for $p = 17$) and there are instances for which SNC has a strictly better LR bound than SNC(SB) (such as rat99 for $p = 33$).

The diagram below illustrates the relations that have been established between the LR bounds of different incomplete models, both from the observations above and from the propositions at the end of section 3.5. This diagram also applies to the corresponding complete models. In this diagram, if there exists a full arrow going from the name of some model A to the name of some other model B, then, the LR bound of model B is always at least as good as that of model A. However, if there is a dashed line between a pair of models A and B, then, the LR bounds of both models cannot be related. In other words, there are instances for which model A has a strictly better LR bound than model B, and there are instances for which model B has a strictly better LR bound than model A.

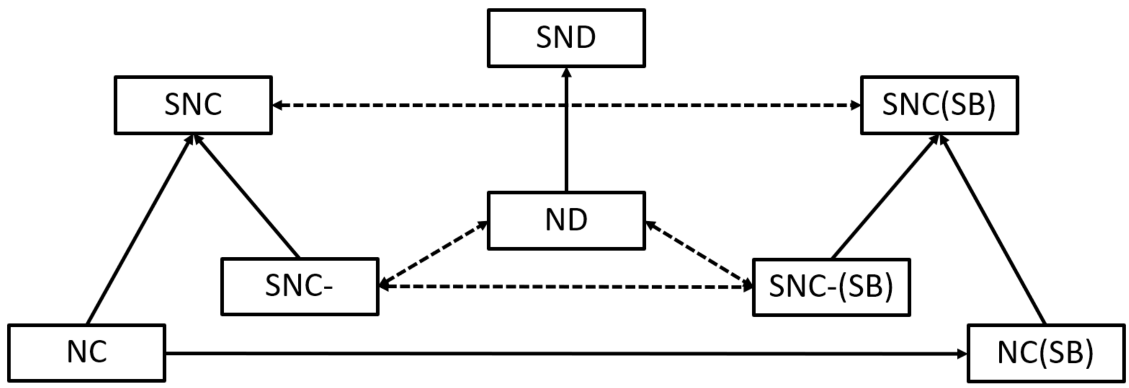

Figure 4.1: Diagram relating the LR bounds of incomplete models

Chapter 5

Conclusions and Future Work

This chapter, which marks the end of this dissertation, starts with an overview of the purpose and contributions of this dissertation, followed by conclusions drawn from the computational study. The chapter ends with a brief discussion of possible future work.

5.1 Main Conclusions

In this thesis, we defined the HpMP, a location-routing problem which combines the p-Median and Travelling Salesman problems, and in which the goal is to, given an undirected graph, determine the cheapest way to partition the graph in exactly p cycles (this problem can also clearly be defined on directed graphs, and the models presented in this work are very simple to adapt for this scenario). The purpose of this work was to study valid compact formulations which prevent solutions featuring less than p cycles.

We began by presenting a formulation upon which all models for the HpMP presented in this work are built, followed by one set of constraints which is an adaptation of the constraints presented by *Desrochers and Laporte* [2] for the TSP used to prevent solutions with more than p cycles and some sets of constraints used to prevent solutions with less than p cycles. To this end, we began by presenting a formulation in which nodes are assigned to depots (which is very similar to a formulation first presented in [3]). This was followed by a valid inequality which results in slight improvements to LR bounds. This model was followed by a new formulation in which nodes are assigned to cycles. Additional valid inequalities and improvements to some of the constraints for this model were also presented, with the potential to improve LR bounds and computational times.

All these models were implemented using Concert Technology for C++ and a set of instances was downloaded from TSPLIB (http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/). Each instance from this set was then solved using CPLEX (version 20.1), for many different numbers of cycles.

Finally, the results from the aforementioned computational experiment were analysed. These results show that some of the modifications presented for the model in which nodes are assigned to cycles result in better computational times, while others do not result in substantial improvements. The results also show that the models in which nodes are assigned to depots result in the lowest computational times out of all the models presented in this dissertation. These results also allowed us to draw more conclusions regarding the LR bounds of the models - namely, that the LR bounds of some models cannot be related.

5.2 Future Work

Regarding future work, there is still much to be done, both regarding compact formulations for this problem in general and regarding the formulations presented in this dissertation. Some compact formulations presented in other literature - in particular, the formulation presented in [10] to prevent solutions with less than p cycles and the p-Median model presented in [11] - may be related to the compact formulations presented in this dissertation, and it may be worth studying the aforementioned models and comparing these to the (more) compact formulations presented in this dissertation.

It may also be interesting to perform a computational experiment in which the models presented in this dissertation are tested for asymmetric instances, and for either symmetric or asymmetric instances allowing cycles with two nodes. The modification of most models to allow for cycles (or circuits) with two nodes is very straightforward - for instance, if the models have been modified to replace the u_{ij} variables with the x_{ij} variables as instructed in section 3.2, all that is necessary is to remove constraints (3.10), remove the term $(M-2)x_{ji}$ from constraints (3.13), modify the constant M used in constraints (3.13) and (3.14) (turning it into $n-2(p-1)$) and replace $x_{ij}+x_{ji}$ with x_{ij} in constraints (3.17), (3.19), (3.20), (3.21), (3.25), (3.26), (3.27) and (3.29).

Additionally, and especially considering the SB strategies considered throughout this work (which depend on the ordering of the nodes), it may also be of interest to test whether the ordering of the nodes has a significant impact on the computational times.

Bibliography

[1] D. L. Applegate, R. E. Bixby, V. Chvátal, and W. J. Cook, *The Traveling Salesman Problem: A Computational Study*. Princeton University Press, 2011, ISBN: 9781400841103.

[2] M. Desrochers and G. Laporte, "Improvements and extensions to the miller-tucker-zemlin subtour elimination constraints," *Operations Research Letters*, vol. 10, no. 1, pp. 27–36, 1991.

[3] M. Burger, Z. Su, and B. De Schutter, "A node current-based 2-index formulation for the fixed-destination multi-depot travelling salesman problem," *European Journal of Operational Research*, vol. 265, no. 2, pp. 463–477, 2018.

[4] Z.-J. M. Shen, C. Coullard, and M. S. Daskin, "A joint location-inventory model," *Transportation Science*, vol. 37, pp. 40–55, 2003.

[5] M. S. Daskin and K. L. Maass, "The p-median problem," in *Location Science*, G. Laporte, S. Nickel, and F. Saldanha da Gama, Eds. Springer International Publishing, 2015, pp. 21–45.

[6] H. Glaab and A. Pott, "The hamiltonian p-median problem," *The Electronic Journal of Combinatorics*, vol. 7, no. 1, R42, 2000.

[7] G. Laporte, Y. Nobert, and P. Pelletier, "Hamiltonian location problems," *European Journal of Operational Research*, vol. 12, pp. 82–89, 1983.

[8] G. Nagy and S. Salhi, "Location-routing: Issues, models and methods," *European Journal of Operational Research*, vol. 177, no. 2, pp. 649–672, 2007.

[9] I. M. Branco and J. D. Coelho, "The hamiltonian p-median problem," *European Journal of Operational Research*, vol. 47, pp. 86–95, 1990.

[10] S. Gollowitzer, D. L. Pereira, and A. Wojciechowski, "New models for and numerical tests of the hamiltonian p-median problem," in *Network Optimization*, J. Pahl, T. Reiners, and S. Voß, Eds., Springer Berlin Heidelberg, 2011, pp. 385–394, ISBN: 978-3-642-21527-8.

[11] S. Gollowitzer, L. Gouveia, G. Laporte, D. L. Pereira, and A. Wojciechowski, "A comparison of several models for the hamiltonian p-median problem," *Networks*, vol. 63, no. 4, pp. 350–363, 2014.

[12] G. Erdoğan, G. Laporte, and A. M. R. Chía, "Exact and heuristic algorithms for the hamiltonian p-median problem," *European Journal of Operational Research*, vol. 253, pp. 280–289, 2016.

[13] A. M. Marzouk, E. Moreno-Centeno, and H. Üster, "A branch-and-price algorithm for solving the hamiltonian p-median problem," *INFORMS Journal on Computing*, vol. 28, no. 4, pp. 674–686, 2016.

[14] T. Bektaş, L. Gouveia, and D. Santos, "Revisiting the hamiltonian p-median problem: A new formulation on directed graphs and a branch-and-cut algorithm," *European Journal of Operational Research*, vol. 276, pp. 40–64, 2019.

[15] W. Pulleyblank, "Chapter v polyhedral combinatorics," in *Optimization*, ser. Handbooks in Operations Research and Management Science, vol. 1, Elsevier, 1989, pp. 371–446.

Appendix A

Test Results

The test results for all complete models are presented in tables A.1 through to A.20, followed by the test results for all incomplete models, which are presented in tables A.21 through to A.30. Each one of the tables with test results has 8 columns. From left to right, they indicate the number of cycles of each feasible solution ("p"), the optimal value of the corresponding instance ("OPT"), the model which was tested ("Model"), the LR bound ("LR"), the lower bound on the optimal value ("LB"), the upper bound on the optimal value ("UB"), the computational time (in seconds) ("Time (s)") and the LR gap ("GAP (%)"). The computational times to determine the LR bounds are not included as these were always negligible.

In each one of these tables, a set of test results is also preceded by a row with an instance name (meaning the test results that follow were obtained for that instance). Each instance name is also accompanied by a constant (denoted by p'), which represents the value of p for which the optimal value is lowest for the corresponding graph.

Regarding the OPT values, an observation is in order - for some of the instances, no optimal value was known at the time of writing this dissertation. Therefore, some of the OPT values are simply the value of the best solution found in either [12] or [14] without proof of optimality (if the OPT value is followed by "*").

p	OPT	Model	LR	LB	UB	Time (s)	GAP (%)
		gr24 ($p' = 4$)					
2	1238	DNC	1224.5	1238	1238	0.3	1.09%
		SDNC-	1224.5	1238	1238	0.3	1.09%
		SDNC	1224.5	1238	1238	0.2	1.09%
		DNC(SB)	1224.5	1238	1238	0.3	1.09%
		SDNC-(SB)	1224.5	1238	1238	0.2	1.09%
		SDNC(SB)	1224.5	1238	1238	0.1	1.09%
		DND	1224.5	1238	1238	0.3	1.09%
		SDND	1224.5	1238	1238	0.1	1.09%
3	1227	DNC	1224.5	1227	1227	0.1	0.20%
		SDNC-	1224.5	1227	1227	0.1	0.20%
		SDNC	1224.5	1227	1227	0.1	0.20%
		DNC(SB)	1224.5	1227	1227	0.1	0.20%
		SDNC-(SB)	1224.5	1227	1227	0.1	0.20%
		SDNC(SB)	1224.5	1227	1227	0.1	0.20%
		DND	1224.5	1227	1227	0.1	0.20%
		SDND	1224.5	1227	1227	0.1	0.20%
4	1227	DNC	1224.5	1227	1227	0.2	0.20%
		SDNC-	1224.5	1227	1227	0.1	0.20%
		SDNC	1224.5	1227	1227	0.1	0.20%
		DNC(SB)	1224.5	1227	1227	0.1	0.20%
		SDNC-(SB)	1224.5	1227	1227	0.1	0.20%
		SDNC(SB)	1224.5	1227	1227	0.1	0.20%
		DND	1224.5	1227	1227	0.1	0.20%
		SDND	1224.5	1227	1227	0.1	0.20%
6	1266	DNC	1224.5	1266	1266	1.9	3.28%
		SDNC-	1224.5	1266	1266	1.4	3.28%
		SDNC	1224.5	1266	1266	1.3	3.28%
		DNC(SB)	1224.5	1266	1266	1.5	3.28%
		SDNC-(SB)	1224.5	1266	1266	0.9	3.28%
		SDNC(SB)	1224.5	1266	1266	0.9	3.28%
		DND	1224.5	1266	1266	1	3.28%
		SDND	1224.61	1266	1266	0.7	3.27%
8	1317	DNC	1224.5	1317	1317	3.3	7.02%
		SDNC-	1226.5	1317	1317	1.6	6.87%
		SDNC	1228.06	1317	1317	1.3	6.75%
		DNC(SB)	1224.5	1317	1317	19.2	7.02%
		SDNC-(SB)	1227.33	1317	1317	1.7	6.81%
		SDNC(SB)	1228.38	1317	1317	1.6	6.73%
		DND	1224.5	1317	1317	1.2	7.02%
		SDND	1231.34	1317	1317	1.1	6.50%

Table A.1: Test results for complete models (instance: gr24)

p	OPT	Model	LR	LB	UB	Time (s)	GAP (%)
		fri26 ($p' = 7$)					
2	911	DNC	880	911	911	0.6	3.40%
		SDNC-	880	911	911	0.6	3.40%
		SDNC	880	911	911	0.5	3.40%
		DNC(SB)	880	911	911	0.4	3.40%
		SDNC-(SB)	880	911	911	0.7	3.40%
		SDNC(SB)	880	911	911	0.5	3.40%
		DND	880	911	911	0.6	3.40%
		SDND	880	911	911	0.6	3.40%
3	903	DNC	880	903	903	0.6	2.55%
		SDNC-	880	903	903	1.3	2.55%
		SDNC	880	903	903	0.6	2.55%
		DNC(SB)	880	903	903	0.7	2.55%
		SDNC-(SB)	880	903	903	0.7	2.55%
		SDNC(SB)	880	903	903	0.8	2.55%
		DND	880	903	903	0.7	2.55%
		SDND	880	903	903	1	2.55%
5	893	DNC	880	893	893	0.6	1.46%
		SDNC-	880	893	893	0.6	1.46%
		SDNC	880	893	893	0.6	1.46%
		DNC(SB)	880	893	893	0.8	1.46%
		SDNC-(SB)	880	893	893	0.4	1.46%
		SDNC(SB)	880	893	893	0.5	1.46%
		DND	880	893	893	0.5	1.46%
		SDND	880	893	893	0.5	1.46%
6	886	DNC	880	886	886	0.5	0.68%
		SDNC-	880	886	886	0.1	0.68%
		SDNC	880	886	886	0.3	0.68%
		DNC(SB)	880	886	886	0.5	0.68%
		SDNC-(SB)	880	886	886	0.2	0.68%
		SDNC(SB)	880	886	886	0.3	0.68%
		DND	880	886	886	0.4	0.68%
		SDND	880	886	886	0.3	0.68%
8	885	DNC	880	885	885	0.8	0.56%
		SDNC-	881.29	885	885	0.1	0.42%
		SDNC	881.47	885	885	0.2	0.40%
		DNC(SB)	880	885	885	0.5	0.56%
		SDNC-(SB)	881.4	885	885	0.1	0.41%
		SDNC(SB)	881.59	885	885	0.1	0.38%
		DND	880	885	885	0.2	0.56%
		SDND	881.02	885	885	0.1	0.45%

Table A.2: Test results for complete models (instance: fri26)

p	OPT	Model	LR	LB	UB	Time (s)	GAP (%)
		bayg29 ($p' = 3$)					
2	1562	DNC	1546	1562	1562	0.3	1.02%
		SDNC-	1546	1562	1562	0.3	1.02%
		SDNC	1546	1562	1562	0.3	1.02%
		DNC(SB)	1546	1562	1562	0.5	1.02%
		SDNC-(SB)	1546	1562	1562	0.4	1.02%
		SDNC(SB)	1546	1562	1562	0.3	1.02%
		DND	1546	1562	1562	0.3	1.02%
		SDND	1546	1562	1562	0.3	1.02%
4	1549	DNC	1546	1549	1549	0.2	0.19%
		SDNC-	1546	1549	1549	0.2	0.19%
		SDNC	1546	1549	1549	0.2	0.19%
		DNC(SB)	1546	1549	1549	0.8	0.19%
		SDNC-(SB)	1546	1549	1549	0.2	0.19%
		SDNC(SB)	1546	1549	1549	0.3	0.19%
		DND	1546	1549	1549	0.4	0.19%
		SDND	1546	1549	1549	0.2	0.19%
5	1555	DNC	1546	1555	1555	0.7	0.58%
		SDNC-	1546	1555	1555	0.3	0.58%
		SDNC	1546	1555	1555	0.4	0.58%
		DNC(SB)	1546	1555	1555	0.6	0.58%
		SDNC-(SB)	1546	1555	1555	0.3	0.58%
		SDNC(SB)	1546	1555	1555	0.3	0.58%
		DND	1546	1555	1555	0.7	0.58%
		SDND	1546	1555	1555	0.3	0.58%
7	1618	DNC	1546	1618	1618	19.1	4.45%
		SDNC-	1546	1618	1618	7.9	4.45%
		SDNC	1546	1618	1618	6.1	4.45%
		DNC(SB)	1546	1618	1618	18.2	4.45%
		SDNC-(SB)	1546	1618	1618	18	4.45%
		SDNC(SB)	1546.03	1618	1618	19.5	4.45%
		DND	1546	1618	1618	20.7	4.45%
		SDND	1546.26	1618	1618	12.2	4.43%
9	1676	DNC	1546	1676	1676	24.8	7.76%
		SDNC-	1546	1676	1676	15.1	7.76%
		SDNC	1546	1676	1676	11.5	7.76%
		DNC(SB)	1546	1676	1676	122.6	7.76%
		SDNC-(SB)	1546.56	1676	1676	141.1	7.72%
		SDNC(SB)	1546.97	1676	1676	81.2	7.70%
		DND	1546.09	1676	1676	30.6	7.75%
		SDND	1547.61	1676	1676	28	7.66%

Table A.3: Test results for complete models (instance: bayg29)

p	OPT	Model	LR	LB	UB	Time (s)	GAP (%)
swiss42 ($p' = 7$)							
4	1232	DNC	1214.5	1232	1232	2.4	1.42%
		SDNC-	1214.5	1232	1232	2.3	1.42%
		SDNC	1214.5	1232	1232	1.2	1.42%
		DNC(SB)	1214.5	1232	1232	3.8	1.42%
		SDNC-(SB)	1214.5	1232	1232	1.9	1.42%
		SDNC(SB)	1214.5	1232	1232	3.4	1.42%
		DND	1214.5	1232	1232	0.9	1.42%
		SDND	1214.5	1232	1232	1.5	1.42%
6	1231	DNC	1214.5	1231	1231	1.8	1.34%
		SDNC-	1214.5	1231	1231	1.3	1.34%
		SDNC	1214.5	1231	1231	1.2	1.34%
		DNC(SB)	1214.5	1231	1231	2.5	1.34%
		SDNC-(SB)	1214.5	1231	1231	1.9	1.34%
		SDNC(SB)	1214.5	1231	1231	1.9	1.34%
		DND	1214.5	1231	1231	1.7	1.34%
		SDND	1214.5	1231	1231	1.8	1.34%
8	1231	DNC	1214.5	1231	1231	3.2	1.34%
		SDNC-	1214.5	1231	1231	1.1	1.34%
		SDNC	1214.5	1231	1231	1.3	1.34%
		DNC(SB)	1214.5	1231	1231	2.5	1.34%
		SDNC-(SB)	1214.5	1231	1231	0.9	1.34%
		SDNC(SB)	1214.5	1231	1231	1.7	1.34%
		DND	1214.5	1231	1231	1.5	1.34%
		SDND	1214.5	1231	1231	1.7	1.34%
10	1238	DNC	1214.5	1238	1238	5.6	1.90%
		SDNC-	1214.5	1238	1238	2.9	1.90%
		SDNC	1214.5	1238	1238	1.7	1.90%
		DNC(SB)	1214.5	1238	1238	16.5	1.90%
		SDNC-(SB)	1214.5	1238	1238	2.7	1.90%
		SDNC(SB)	1214.5	1238	1238	3.5	1.90%
		DND	1214.5	1238	1238	2.2	1.90%
		SDND	1214.5	1238	1238	3.2	1.90%
14	1292	DNC	1214.5	1292	1292	125.6	6.00%
		SDNC-	1214.5	1292	1292	16.2	6.00%
		SDNC	1215.56	1292	1292	9	5.92%
		DNC(SB)	1214.5	1292	1292	288.5	6.00%
		SDNC-(SB)	1214.67	1292	1292	11.6	5.99%
		SDNC(SB)	1215.89	1292	1292	19.9	5.89%
		DND	1214.5	1292	1292	8.3	6.00%
		SDND	1219.87	1292	1292	9.7	5.58%

Table A.4: Test results for complete models (instance: swiss42)

p	OPT	Model	LR	LB	UB	Time (s)	GAP (%)
		att48 ($p' = 5$)					
4	31903.3	DNC	31669.3	31903.3	31903.3	1.8	0.73%
		SDNC-	31669.3	31903.3	31903.3	1	0.73%
		SDNC	31669.3	31903.3	31903.3	1	0.73%
		DNC(SB)	31669.3	31903.3	31903.3	2.4	0.73%
		SDNC-(SB)	31669.3	31903.3	31903.3	1.4	0.73%
		SDNC(SB)	31669.3	31903.3	31903.3	1.6	0.73%
		DND	31669.3	31903.3	31903.3	0.9	0.73%
		SDND	31669.3	31903.3	31903.3	0.9	0.73%
6	31836.1	DNC	31669.3	31836.1	31836.1	2.1	0.52%
		SDNC-	31669.3	31836.1	31836.1	0.6	0.52%
		SDNC	31669.3	31836.1	31836.1	0.9	0.52%
		DNC(SB)	31669.3	31836.1	31836.1	1.8	0.52%
		SDNC-(SB)	31669.3	31836.1	31836.1	1.2	0.52%
		SDNC(SB)	31669.3	31836.1	31836.1	0.8	0.52%
		DND	31669.3	31836.1	31836.1	0.6	0.52%
		SDND	31669.3	31836.1	31836.1	0.6	0.52%
9	32195.5	DNC	31669.3	32195.5	32195.5	11.4	1.63%
		SDNC-	31669.3	32195.5	32195.5	7.4	1.63%
		SDNC	31669.3	32195.5	32195.5	6.1	1.63%
		DNC(SB)	31669.3	32195.5	32195.5	83.3	1.63%
		SDNC-(SB)	31669.3	32195.5	32195.5	30.4	1.63%
		SDNC(SB)	31669.3	32195.5	32195.5	14.9	1.63%
		DND	31669.3	32195.5	32195.5	5.6	1.63%
		SDND	31669.3	32195.5	32195.5	9.1	1.63%
12	32742.9	DNC	31669.3	32742.9	32742.9	88.7	3.28%
		SDNC-	31669.3	32742.9	32742.9	24.4	3.28%
		SDNC	31669.5	32742.9	32742.9	17.7	3.28%
		DNC(SB)	31669.3	32742.9	32742.9	918.5	3.28%
		SDNC-(SB)	31669.3	32742.9	32742.9	173.6	3.28%
		SDNC(SB)	31669.8	32742.9	32742.9	98.9	3.28%
		DND	31669.3	32742.9	32742.9	38.6	3.28%
		SDND	31671.1	32742.9	32742.9	23.8	3.27%
16	37068.82	DNC	31669.3	34598.2	37329.1	3600	14.57%
		SDNC-	31669.3	35688.3	37068.8	3600	14.57%
		SDNC	31693.9	34443.9	37068.8	3600	14.50%
		DNC(SB)	31669.3	33002.3	37561.9	3600	14.57%
		SDNC-(SB)	31673.7	35601.8	37068.8	3600	14.55%
		SDNC(SB)	31710.8	34731.2	37068.8	3600	14.45%
		DND	31669.7	35512	37068.8	3600	14.57%
		SDND	31780.7	34802	37329.1	3600	14.27%

Table A.5: Test results for complete models (instance: att48)

p	OPT	Model	LR	LB	UB	Time (s)	GAP (%)
		gr48 ($p' = 6$)					
4	4841	DNC	4769	4841	4841	12.3	1.49%
		SDNC-	4769	4841	4841	12.6	1.49%
		SDNC	4769	4841	4841	9.3	1.49%
		DNC(SB)	4769	4841	4841	9.9	1.49%
		SDNC-(SB)	4769	4841	4841	6.9	1.49%
		SDNC(SB)	4769	4841	4841	13.5	1.49%
		DND	4769	4841	4841	4.2	1.49%
		SDND	4769	4841	4841	7.2	1.49%
6	4805	DNC	4769	4805	4805	1.9	0.75%
		SDNC-	4769	4805	4805	0.7	0.75%
		SDNC	4769	4805	4805	0.8	0.75%
		DNC(SB)	4769	4805	4805	1.4	0.75%
		SDNC-(SB)	4769	4805	4805	1.2	0.75%
		SDNC(SB)	4769	4805	4805	0.9	0.75%
		DND	4769	4805	4805	0.6	0.75%
		SDND	4769	4805	4805	0.6	0.75%
9	4926	DNC	4769	4926	4926	899.5	3.19%
		SDNC-	4769	4926	4926	703.3	3.19%
		SDNC	4769	4926	4926	1598.3	3.19%
		DNC(SB)	4769	4926	4926	1412.5	3.19%
		SDNC-(SB)	4769	4926	4926	703.2	3.19%
		SDNC(SB)	4769	4926	4926	677.2	3.19%
		DND	4769	4926	4926	171.2	3.19%
		SDND	4769	4926	4926	151.5	3.19%
12	5011	DNC	4769	4936.62	5011	3600	4.83%
		SDNC-	4769	5011	5011	523.6	4.83%
		SDNC	4769	4940.43	5011	3600	4.83%
		DNC(SB)	4769	4922.22	5011	3600	4.83%
		SDNC-(SB)	4769	4956.07	5011	3600	4.83%
		SDNC(SB)	4769	5011	5011	1812.8	4.83%
		DND	4769	5011	5011	826	4.83%
		SDND	4769	5011	5011	574.2	4.83%
16	5445	DNC	4769	5072.72	5445	3600	12.42%
		SDNC-	4769	5258.41	5445	3600	12.42%
		SDNC	4769	5392.45	5445	3600	12.42%
		DNC(SB)	4769	5057.99	5445	3600	12.42%
		SDNC-(SB)	4769	5121.84	5445	3600	12.42%
		SDNC(SB)	4769.49	5140.28	5445	3600	12.41%
		DND	4769	5329.13	5445	3600	12.42%
		SDND	4776.57	5261.85	5445	3600	12.28%

Table A.6: Test results for complete models (instance: gr48)

p	OPT	Model	LR	LB	UB	Time (s)	GAP (%)
		hk48 ($p' = 6$)					
4	11271	DNC	11197	11271	11271	9.9	0.66%
		SDNC-	11197	11271	11271	6.8	0.66%
		SDNC	11197	11271	11271	10.9	0.66%
		DNC(SB)	11197	11271	11271	11	0.66%
		SDNC-(SB)	11197	11271	11271	8.2	0.66%
		SDNC(SB)	11197	11271	11271	6.6	0.66%
		DND	11197	11271	11271	6.9	0.66%
		SDND	11197	11271	11271	18.1	0.66%
6	11197	DNC	11197	11197	11197	0.3	0.00%
		SDNC-	11197	11197	11197	0.2	0.00%
		SDNC	11197	11197	11197	0.3	0.00%
		DNC(SB)	11197	11197	11197	0.2	0.00%
		SDNC-(SB)	11197	11197	11197	0.2	0.00%
		SDNC(SB)	11197	11197	11197	0.2	0.00%
		DND	11197	11197	11197	0.1	0.00%
		SDND	11197	11197	11197	0.2	0.00%
9	11292	DNC	11197	11292	11292	68.7	0.84%
		SDNC-	11197	11292	11292	30.5	0.84%
		SDNC	11197	11292	11292	13.6	0.84%
		DNC(SB)	11197	11292	11292	54.4	0.84%
		SDNC-(SB)	11197	11292	11292	36.6	0.84%
		SDNC(SB)	11197	11292	11292	17.4	0.84%
		DND	11197	11292	11292	7.3	0.84%
		SDND	11197	11292	11292	7.8	0.84%
12	11450	DNC	11197	11450	11450	1193.7	2.21%
		SDNC-	11197	11450	11450	99.3	2.21%
		SDNC	11197	11450	11450	216.3	2.21%
		DNC(SB)	11197	11450	11450	3428.2	2.21%
		SDNC-(SB)	11197	11450	11450	141.2	2.21%
		SDNC(SB)	11197	11450	11450	257.8	2.21%
		DND	11197	11450	11450	23.3	2.21%
		SDND	11197.1	11450	11450	44.3	2.21%
16	12215	DNC	11197	12031.8	12215	3600	8.33%
		SDNC-	11197	11969.3	12215	3600	8.33%
		SDNC	11197.2	12115.8	12215	3600	8.33%
		DNC(SB)	11197	12215	12215	2612.7	8.33%
		SDNC-(SB)	11198.5	12215	12215	2029.7	8.32%
		SDNC(SB)	11199.7	12215	12215	1286.6	8.31%
		DND	11197	12215	12215	1583.7	8.33%
		SDND	11202.7	12215	12215	1265.1	8.29%

Table A.7: Test results for complete models (instance: hk48)

p	OPT	Model	LR	LB	UB	Time (s)	GAP (%)
		eil51 ($p' = 3$)					
5	422.323	DNC	418.69	422.32	422.32	5.4	0.86%
		SDNC-	418.69	422.32	422.32	4.1	0.86%
		SDNC	418.69	422.32	422.32	2.9	0.86%
		DNC(SB)	418.69	422.32	422.32	6.2	0.86%
		SDNC-(SB)	418.69	422.32	422.32	2.8	0.86%
		SDNC(SB)	418.69	422.32	422.32	1.4	0.86%
		DND	418.69	422.32	422.32	2.8	0.86%
		SDND	418.69	422.32	422.32	1.6	0.86%
7	424.356	DNC	418.69	424.36	424.36	35.1	1.34%
		SDNC-	418.69	424.36	424.36	61.7	1.34%
		SDNC	418.69	424.36	424.36	6.1	1.34%
		DNC(SB)	418.69	424.36	424.36	68.3	1.34%
		SDNC-(SB)	418.69	424.36	424.36	10.8	1.34%
		SDNC(SB)	418.69	424.36	424.36	10.5	1.34%
		DND	418.69	424.36	424.36	7.2	1.34%
		SDND	418.73	424.36	424.36	10.6	1.32%
10	432.489	DNC	418.69	428.32	432.49	3600	3.19%
		SDNC-	418.69	427.88	432.49	3600	3.19%
		SDNC	418.7	432.49	432.49	1379.3	3.19%
		DNC(SB)	418.69	427.08	432.49	3600	3.19%
		SDNC-(SB)	418.71	431.42	432.49	3600	3.19%
		SDNC(SB)	418.83	432.49	432.49	1889.7	3.16%
		DND	418.75	432.49	432.49	1941.9	3.18%
		SDND	419.02	432.49	432.49	910.5	3.11%
12	436.587	DNC	418.69	425.57	436.59	3600	4.10%
		SDNC-	418.69	431.71	436.59	3600	4.10%
		SDNC	418.84	436.59	436.59	3286.5	4.07%
		DNC(SB)	418.69	429.04	436.59	3600	4.10%
		SDNC-(SB)	418.78	433.15	436.59	3600	4.08%
		SDNC(SB)	419.06	434.02	436.59	3600	4.02%
		DND	418.87	436.59	436.59	2796.2	4.06%
		SDND	419.28	436.59	436.59	3024.4	3.96%
17	473.977	DNC	418.69	438.02	475.89	3600	11.67%
		SDNC-	418.76	450.05	475.56	3600	11.65%
		SDNC	419.69	449.3	473.98	3600	11.45%
		DNC(SB)	418.69	442.38	473.98	3600	11.67%
		SDNC-(SB)	419.2	444.1	473.98	3600	11.56%
		SDNC(SB)	419.99	448.66	475.89	3600	11.39%
		DND	419.32	454.71	473.98	3600	11.53%
		SDND	420.25	449.75	475.56	3600	11.34%

Table A.8: Test results for complete models (instance: eil51)

p	OPT	Model	LR	LB	UB	Time (s)	GAP (%)
		berlin52 ($p' = 7$)					
5	7182.23	DNC	7166.3	7182.23	7182.23	7.2	0.22%
		SDNC-	7166.3	7182.23	7182.23	5.6	0.22%
		SDNC	7166.3	7182.23	7182.23	4.3	0.22%
		DNC(SB)	7166.3	7182.23	7182.23	4.9	0.22%
		SDNC-(SB)	7166.3	7182.23	7182.23	5.5	0.22%
		SDNC(SB)	7166.3	7182.23	7182.23	5.1	0.22%
		DND	7166.3	7182.23	7182.23	5	0.22%
		SDND	7166.3	7182.23	7182.23	3.3	0.22%
7	7167.2	DNC	7166.3	7167.2	7167.2	1.4	0.01%
		SDNC-	7166.3	7167.2	7167.2	1	0.01%
		SDNC	7166.3	7167.2	7167.2	0.7	0.01%
		DNC(SB)	7166.3	7167.2	7167.2	2.1	0.01%
		SDNC-(SB)	7166.3	7167.2	7167.2	0.4	0.01%
		SDNC(SB)	7166.3	7167.2	7167.2	0.4	0.01%
		DND	7166.3	7167.2	7167.2	0.6	0.01%
		SDND	7166.3	7167.2	7167.2	0.5	0.01%
10	7206.7	DNC	7166.3	7206.7	7206.7	86.6	0.56%
		SDNC-	7166.3	7206.7	7206.7	60.5	0.56%
		SDNC	7166.3	7206.7	7206.7	12.2	0.56%
		DNC(SB)	7166.3	7206.7	7206.7	60.1	0.56%
		SDNC-(SB)	7166.3	7206.7	7206.7	12.2	0.56%
		SDNC(SB)	7166.32	7206.7	7206.7	24.4	0.56%
		DND	7166.3	7206.7	7206.7	7.4	0.56%
		SDND	7166.32	7206.7	7206.7	4.2	0.56%
13	7298.63	DNC	7166.3	7298.63	7298.63	1155.4	1.81%
		SDNC-	7166.3	7298.63	7298.63	91.4	1.81%
		SDNC	7166.4	7298.63	7298.63	107.9	1.81%
		DNC(SB)	7166.3	7298.63	7298.63	2475.9	1.81%
		SDNC-(SB)	7166.32	7298.63	7298.63	131.1	1.81%
		SDNC(SB)	7166.43	7298.63	7298.63	134.3	1.81%
		DND	7166.3	7298.63	7298.63	29	1.81%
		SDND	7166.46	7298.63	7298.63	24.8	1.81%
17	7800.77	DNC	7166.3	7248.42	17148.4	3600	8.13%
		SDNC-	7166.48	7515.56	7800.77	3600	8.13%
		SDNC	7168.01	7557.95	7800.77	3600	8.11%
		DNC(SB)	7166.3	7372.17	7800.77	3600	8.13%
		SDNC-(SB)	7166.65	7447.52	7800.77	3600	8.13%
		SDNC(SB)	7168.34	7353.14	9436.62	3600	8.11%
		DND	7166.33	7590.22	7800.77	3600	8.13%
		SDND	7173.26	7557.56	7800.77	3600	8.04%

Table A.9: Test results for complete models (instance: berlin52)

p	OPT	Model	LR	LB	UB	Time (s)	GAP (%)
brazil58 ($p' = 12$)							
5	21744	DNC	20896	21353.6	21744	3600	3.90%
		SDNC-	20896	21364.3	21744	3600	3.90%
		SDNC	20896	21378.7	21744	3600	3.90%
		DNC(SB)	20896	21389.6	21744	3600	3.90%
		SDNC-(SB)	20896	21331.4	21744	3600	3.90%
		SDNC(SB)	20896	21364.5	21744	3600	3.90%
		DND	20896	21384.8	21937	3600	3.90%
		SDND	20896	21455.2	21744	3600	3.90%
8	21289	DNC	20896	21289	21289	1015.6	1.85%
		SDNC-	20896	21289	21289	670.8	1.85%
		SDNC	20896	21289	21289	631	1.85%
		DNC(SB)	20896	21289	21289	892.6	1.85%
		SDNC-(SB)	20896	21289	21289	597.8	1.85%
		SDNC(SB)	20896	21289	21289	238.2	1.85%
		DND	20896	21289	21289	1517.3	1.85%
		SDND	20896	21289	21289	328	1.85%
11	21080	DNC	20896	21080	21080	7.7	0.87%
		SDNC-	20896	21080	21080	3	0.87%
		SDNC	20896	21080	21080	2.7	0.87%
		DNC(SB)	20896	21080	21080	4	0.87%
		SDNC-(SB)	20896	21080	21080	2.9	0.87%
		SDNC(SB)	20896	21080	21080	2.4	0.87%
		DND	20896	21080	21080	4.2	0.87%
		SDND	20896	21080	21080	2.2	0.87%
14	21221	DNC	20896	21221	21221	74.7	1.53%
		SDNC-	20896	21221	21221	10.6	1.53%
		SDNC	20896	21221	21221	13.5	1.53%
		DNC(SB)	20896	21221	21221	120.3	1.53%
		SDNC-(SB)	20896	21221	21221	54.7	1.53%
		SDNC(SB)	20896	21221	21221	7	1.53%
		DND	20896	21221	21221	5.1	1.53%
		SDND	20896	21221	21221	5.1	1.53%
19	22635	DNC	20896	21238.8	35277	3600	7.68%
		SDNC-	20896	21972.7	22635	3600	7.68%
		SDNC	20900.6	21784.2	22635	3600	7.66%
		DNC(SB)	20896	21299.1	22635	3600	7.68%
		SDNC-(SB)	20896.7	21927.2	22635	3600	7.68%
		SDNC(SB)	20909.1	21818.1	22635	3600	7.62%
		DND	20896	22217.9	22635	3600	7.68%
		SDND	20924.6	22243.3	22635	3600	7.56%

Table A.10: Test results for complete models (instance: brazil58)

p	OPT	Model	LR	LB	UB	Time (s)	GAP (%)
		st70 ($p' = 12$)					
7	638.221	DNC	628.5	638.22	638.22	1151.5	1.52%
		SDNC-	628.5	638.22	638.22	1172.1	1.52%
		SDNC	628.5	638.22	638.22	1069.4	1.52%
		DNC(SB)	628.5	638.22	638.22	563.7	1.52%
		SDNC-(SB)	628.5	638.22	638.22	686.6	1.52%
		SDNC(SB)	628.5	638.22	638.22	786.8	1.52%
		DND	628.5	638.22	638.22	447.6	1.52%
		SDND	628.5	638.22	638.22	346.1	1.52%
10	632.54	DNC	628.5	632.54	632.54	87.4	0.64%
		SDNC-	628.5	632.54	632.54	46.2	0.64%
		SDNC	628.5	632.54	632.54	56.5	0.64%
		DNC(SB)	628.5	632.54	632.54	226	0.64%
		SDNC-(SB)	628.5	632.54	632.54	83.9	0.64%
		SDNC(SB)	628.5	632.54	632.54	53	0.64%
		DND	628.5	632.54	632.54	58	0.64%
		SDND	628.5	632.54	632.54	52.8	0.64%
14	630.902	DNC	628.5	630.9	630.9	23.2	0.38%
		SDNC-	628.5	630.9	630.9	45.7	0.38%
		SDNC	628.5	630.9	630.9	7.8	0.38%
		DNC(SB)	628.5	630.9	630.9	79	0.38%
		SDNC-(SB)	628.5	630.9	630.9	6.3	0.38%
		SDNC(SB)	628.5	630.9	630.9	6.5	0.38%
		DND	628.5	630.9	630.9	4	0.38%
		SDND	628.5	630.9	630.9	3.2	0.38%
17	636.194	DNC	628.5	636.19	636.19	709.6	1.21%
		SDNC-	628.5	636.19	636.19	182.1	1.21%
		SDNC	628.5	636.19	636.19	64.9	1.21%
		DNC(SB)	628.5	636.19	636.19	1299.7	1.21%
		SDNC-(SB)	628.5	636.19	636.19	76.3	1.21%
		SDNC(SB)	628.5	636.19	636.19	34	1.21%
		DND	628.5	636.19	636.19	18.5	1.21%
		SDND	628.52	636.19	636.19	14.3	1.21%
23	694.495	DNC	628.5	641.54	1069.9	3600	9.50%
		SDNC-	628.5	646.28	696.42	3600	9.50%
		SDNC	628.67	647.81	694.49	3600	9.48%
		DNC(SB)	628.5	639.01	1343.1	3600	9.50%
		SDNC-(SB)	628.63	643.39	1192.82	3600	9.48%
		SDNC(SB)	628.72	647.8	695.8	3600	9.47%
		DND	628.5	652.26	694.49	3600	9.50%
		SDND	629.6	654.32	694.49	3600	9.34%

Table A.11: Test results for complete models (instance: st70)

p	OPT	Model	LR	LB	UB	Time (s)	GAP (%)
eil76 ($p' = 4$)							
7	542.954	DNC	540.72	542.95	542.95	304.3	0.41%
		SDNC-	540.72	542.95	542.95	109.6	0.41%
		SDNC	540.72	542.95	542.95	39.1	0.41%
		DNC(SB)	540.72	542.95	542.95	169.7	0.41%
		SDNC-(SB)	540.72	542.95	542.95	188.8	0.41%
		SDNC(SB)	540.72	542.95	542.95	86.8	0.41%
		DND	540.72	542.95	542.95	36.3	0.41%
		SDND	540.72	542.95	542.95	31.9	0.41%
10	545.021	DNC	540.72	543.89	545.02	3600	0.79%
		SDNC-	540.72	545.02	545.02	1118.2	0.79%
		SDNC	540.72	545.02	545.02	635.5	0.79%
		DNC(SB)	540.72	545.02	545.02	2962.5	0.79%
		SDNC-(SB)	540.72	545.02	545.02	2568.6	0.79%
		SDNC(SB)	540.72	545.02	545.02	1071.4	0.79%
		DND	540.72	545.02	545.02	129.9	0.79%
		SDND	540.72	545.02	545.02	85.3	0.79%
15	552.149	DNC	540.72	542.08	560.18	3600	2.07%
		SDNC-	540.72	546.22	552.15	3600	2.07%
		SDNC	540.72	543.53	552.33	3600	2.07%
		DNC(SB)	540.72	542.15	552.33	3600	2.07%
		SDNC-(SB)	540.72	544.8	552.15	3600	2.07%
		SDNC(SB)	540.72	544.98	552.15	3600	2.07%
		DND	540.72	545.68	552.15	3600	2.07%
		SDND	540.72	546.46	552.15	3600	2.07%
19	563.955	DNC	540.72	542.43	689.72	3600	4.12%
		SDNC-	540.72	547.77	564.24	3600	4.12%
		SDNC	540.72	545.78	563.95	3600	4.12%
		DNC(SB)	540.72	542.31	566.24	3600	4.12%
		SDNC-(SB)	540.72	545.2	563.95	3600	4.12%
		SDNC(SB)	540.74	544.75	563.95	3600	4.12%
		DND	540.72	547.85	564.24	3600	4.12%
		SDND	540.89	547.52	563.95	3600	4.09%
25	601.71	DNC	540.72	544.4	2.00E+12	3600	10.14%
		SDNC-	540.72	552.57	734.06	3600	10.14%
		SDNC	541.09	551.29	792.27	3600	10.07%
		DNC(SB)	540.72	543.43	1.00E+12	3600	10.14%
		SDNC-(SB)	540.74	548.21	1.00E+12	3600	10.13%
		SDNC(SB)	541.27	550.9	727.75	3600	10.04%
		DND	540.8	552.91	629.28	3600	10.12%
		SDND	541.59	555.94	816.26	3600	9.99%

Table A.12: Test results for complete models (instance: eil76)

p	OPT	Model	LR	LB	UB	Time (s)	GAP (%)
		pr76 ($p' = 8$)					
7	101401	DNC	98993.9	101401	101401	43.8	2.37%
		SDNC-	98993.9	101401	101401	12.9	2.37%
		SDNC	98993.9	101401	101401	31.1	2.37%
		DNC(SB)	98993.9	101401	101401	86.3	2.37%
		SDNC-(SB)	98993.9	101401	101401	15.8	2.37%
		SDNC(SB)	98993.9	101401	101401	24.9	2.37%
		DND	98993.9	101401	101401	27.2	2.37%
		SDND	99001.5	101401	101401	17.1	2.37%
10	101779	DNC	98993.9	101779	101779	727.1	2.74%
		SDNC-	98993.9	101779	101779	129.6	2.74%
		SDNC	98993.9	101779	101779	66.6	2.74%
		DNC(SB)	98993.9	101779	101779	1192.1	2.74%
		SDNC-(SB)	98993.9	101779	101779	167.6	2.74%
		SDNC(SB)	98993.9	101779	101779	132.2	2.74%
		DND	98993.9	101779	101779	30.9	2.74%
		SDND	99046	101779	101779	28.2	2.69%
15	103663	DNC	98993.9	101450	103868	3600	4.50%
		SDNC-	98993.9	102539	103724	3600	4.50%
		SDNC	99043.8	102332	103663	3600	4.46%
		DNC(SB)	98993.9	101269	103663	3600	4.50%
		SDNC-(SB)	99009.2	103051	103724	3600	4.49%
		SDNC(SB)	99062.9	102316	103663	3600	4.44%
		DND	98993.9	102744	103663	3600	4.50%
		SDND	99130.4	102783	103663	3600	4.37%
19	104482	DNC	98993.9	101753	104482	3600	5.25%
		SDNC-	99044.7	103793	104482	3600	5.20%
		SDNC	99143.5	103080	104482	3600	5.11%
		DNC(SB)	98993.9	102007	104482	3600	5.25%
		SDNC-(SB)	99072.2	103167	104482	3600	5.18%
		SDNC(SB)	99163.4	102746	104482	3600	5.09%
		DND	98998.5	103106	104482	3600	5.25%
		SDND	99241.4	104452	104482	3600	5.02%
25	110074	DNC	98993.9	101581	1.00E+12	3600	10.07%
		SDNC-	99292.6	105447	237642	3600	9.79%
		SDNC	99463.1	104937	113866	3600	9.64%
		DNC(SB)	98993.9	102209	1.00E+12	3600	10.07%
		SDNC-(SB)	99361.9	104374	209562	3600	9.73%
		SDNC(SB)	99477.8	105101	127427	3600	9.63%
		DND	99030.3	106351	110074	3600	10.03%
		SDND	99742.8	106973	110300	3600	9.39%

Table A.13: Test results for complete models (instance: pr76)

p	OPT	Model	LR	LB	UB	Time (s)	GAP (%)
		rat99 ($p' = 8$)					
9	1209.09	DNC	1203.53	1209.09	1209.09	258.6	0.46%
		SDNC-	1203.53	1209.09	1209.09	72.4	0.46%
		SDNC	1203.53	1209.09	1209.09	33.4	0.46%
		DNC(SB)	1203.53	1209.09	1209.09	177.2	0.46%
		SDNC-(SB)	1203.53	1209.09	1209.09	62.4	0.46%
		SDNC(SB)	1203.53	1209.09	1209.09	23.5	0.46%
		DND	1203.53	1209.09	1209.09	42.8	0.46%
		SDND	1203.53	1209.09	1209.09	21.8	0.46%
14	1224.1	DNC	1203.53	1209.54	1224.74	3600	1.68%
		SDNC-	1203.53	1213.95	1224.09	3600	1.68%
		SDNC	1203.53	1211.41	1224.09	3600	1.68%
		DNC(SB)	1203.53	1208.78	1230.39	3600	1.68%
		SDNC-(SB)	1203.53	1213.81	1224.09	3600	1.68%
		SDNC(SB)	1203.53	1211.19	1224.09	3600	1.68%
		DND	1203.53	1214.63	1224.09	3600	1.68%
		SDND	1203.53	1214.48	1224.09	3600	1.68%
19	1245.16	DNC	1203.53	1208.31	1245.16	3600	3.34%
		SDNC-	1203.53	1218.04	1252.59	3600	3.34%
		SDNC	1203.53	1214.85	1253.33	3600	3.34%
		DNC(SB)	1203.53	1208.32	1250.31	3600	3.34%
		SDNC-(SB)	1203.53	1218.49	1250.31	3600	3.34%
		SDNC(SB)	1203.53	1216.22	1251.82	3600	3.34%
		DND	1203.53	1223.71	1251.82	3600	3.34%
		SDND	1203.94	1221.79	1245.16	3600	3.31%
24	1273.23	DNC	1203.53	1208.34	1276.04	3600	5.47%
		SDNC-	1204.43	1236.66	1273.23	3600	5.40%
		SDNC	1204.74	1229.53	1275.6	3600	5.38%
		DNC(SB)	1203.53	1208.44	1.00E+12	3600	5.47%
		SDNC-(SB)	1204.38	1233.17	1273.23	3600	5.41%
		SDNC(SB)	1204.75	1228.66	1273.23	3600	5.38%
		DND	1203.53	1242.56	1273.23	3600	5.47%
		SDND	1205.3	1237.8	1274.17	3600	5.34%
33	1373.37	DNC	1203.53	1218.81	3.00E+12	3600	12.37%
		SDNC-	1213.93	1285.93	1904.8	3600	11.61%
		SDNC	1215.37	1287.42	3.00E+12	3600	11.50%
		DNC(SB)	1203.53	1223.66	3.00E+12	3600	12.37%
		SDNC-(SB)	1213.01	1267.16	3.00E+12	3600	11.68%
		SDNC(SB)	1214.65	1281.15	3.00E+12	3600	11.56%
		DND	1203.53	1291.43	1798.62	3600	12.37%
		SDND	1217.58	1307.1	1726.59	3600	11.34%

Table A.14: Test results for complete models (instance: rat99)

p	OPT	Model	LR	LB	UB	Time (s)	GAP (%)
		kroA100 ($p' = 13$)					
10	19900.9	DNC	19380.7	19603.3	19900.9	3600	2.61%
		SDNC-	19380.7	19635.2	19900.9	3600	2.61%
		SDNC	19380.7	19650.5	19900.9	3600	2.61%
		DNC(SB)	19380.7	19585.4	19900.9	3600	2.61%
		SDNC-(SB)	19380.7	19657.6	19900.9	3600	2.61%
		SDNC(SB)	19380.7	19633.3	19900.9	3600	2.61%
		DND	19380.7	19623.2	19900.9	3600	2.61%
		SDND	19380.7	19629.7	19900.9	3600	2.61%
14	19637.5	DNC	19380.7	19637.5	19637.5	953	1.31%
		SDNC-	19380.7	19637.5	19637.5	204	1.31%
		SDNC	19380.7	19637.5	19637.5	169.7	1.31%
		DNC(SB)	19380.7	19637.5	19637.5	869.5	1.31%
		SDNC-(SB)	19380.7	19637.5	19637.5	480.3	1.31%
		SDNC(SB)	19380.7	19637.5	19637.5	25.7	1.31%
		DND	19380.7	19637.5	19637.5	13.9	1.31%
		SDND	19380.7	19637.5	19637.5	18.2	1.31%
20	19868.6	DNC	19380.7	19566.9	19868.6	3600	2.46%
		SDNC-	19380.7	19675.5	19868.6	3600	2.46%
		SDNC	19380.7	19793	19868.6	3600	2.46%
		DNC(SB)	19380.7	19567.4	20062.9	3600	2.46%
		SDNC-(SB)	19380.7	19767.8	19868.6	3600	2.46%
		SDNC(SB)	19380.7	19711.2	19868.6	3600	2.46%
		DND	19380.7	19868.6	19868.6	870.3	2.46%
		SDND	19380.7	19868.6	19868.6	971.6	2.46%
25	20279.5	DNC	19380.7	19569.6	20429.1	3600	4.43%
		SDNC-	19380.7	19763.7	20321.2	3600	4.43%
		SDNC	19380.7	19870.6	20279.5	3600	4.43%
		DNC(SB)	19380.7	19568.3	20483.5	3600	4.43%
		SDNC-(SB)	19380.7	19719.6	20279.5	3600	4.43%
		SDNC(SB)	19380.7	19798.9	20279.5	3600	4.43%
		DND	19380.7	19934.3	20279.5	3600	4.43%
		SDND	19383.9	20064.9	20279.5	3600	4.42%
33	22303.23*	DNC	19380.7	19670.1	2.00E+12	3600	13.10%
		SDNC-	19380.7	19969	34793.3	3600	13.10%
		SDNC	19411.7	20266.4	29329.6	3600	12.96%
		DNC(SB)	19380.7	19697.5	1.00E+12	3600	13.10%
		SDNC-(SB)	19383.7	19829.2	28585.1	3600	13.09%
		SDNC(SB)	19418.2	20087	66499.7	3600	12.94%
		DND	19380.7	20573.3	24007.6	3600	13.10%
		SDND	19478.9	20462.4	28846.8	3600	12.66%

Table A.15: Test results for complete models (instance: kroA100)

p	OPT	Model	LR	LB	UB	Time (s)	GAP (%)
		kroB100 ($p' = 20$)					
10	20823.1	DNC	20336	20753.9	20921.3	3600	2.34%
		SDNC-	20336	20770.9	20823.1	3600	2.34%
		SDNC	20336	20780.7	20823.1	3600	2.34%
		DNC(SB)	20336	20786.2	20823.1	3600	2.34%
		SDNC-(SB)	20336	20780.7	20823.1	3600	2.34%
		SDNC(SB)	20336	20807.1	20823.1	3600	2.34%
		DND	20336	20793.2	20823.1	3600	2.34%
		SDND	20336	20823.1	20823.1	2327.1	2.34%
14	20762.9	DNC	20336	20675.9	20762.9	3600	2.06%
		SDNC-	20336	20715.4	20762.9	3600	2.06%
		SDNC	20336	20762.9	20762.9	1834.3	2.06%
		DNC(SB)	20336	20683	20762.9	3600	2.06%
		SDNC-(SB)	20336	20723.2	20762.9	3600	2.06%
		SDNC(SB)	20336	20762.9	20762.9	2750.9	2.06%
		DND	20336	20724.3	20762.9	3600	2.06%
		SDND	20336	20762.9	20762.9	2962.5	2.06%
20	20660	DNC	20336	20660	20660	483.1	1.57%
		SDNC-	20336	20660	20660	4.1	1.57%
		SDNC	20336	20660	20660	9.8	1.57%
		DNC(SB)	20336	20660	20660	292.4	1.57%
		SDNC-(SB)	20336	20660	20660	23.6	1.57%
		SDNC(SB)	20336	20660	20660	42.7	1.57%
		DND	20336	20660	20660	16	1.57%
		SDND	20336	20660	20660	16.2	1.57%
25	20786.9	DNC	20336	20682.2	20786.9	3600	2.17%
		SDNC-	20336	20786.9	20786.9	1146.7	2.17%
		SDNC	20336	20786.9	20786.9	464.2	2.17%
		DNC(SB)	20336	20662.7	20786.9	3600	2.17%
		SDNC-(SB)	20336.1	20786.9	20786.9	953.8	2.17%
		SDNC(SB)	20336.1	20786.9	20786.9	127.2	2.17%
		DND	20336	20786.9	20786.9	146.7	2.17%
		SDND	20338	20786.9	20786.9	53.9	2.16%
33	22923.4*	DNC	20336	20685.5	2.00E+12	3600	11.29%
		SDNC-	20346.6	21026.4	28989.3	3600	11.24%
		SDNC	20372.3	21152.3	31443.6	3600	11.13%
		DNC(SB)	20336	20703.3	1.00E+12	3600	11.29%
		SDNC-(SB)	20356.8	20933.7	1.00E+12	3600	11.20%
		SDNC(SB)	20376.3	21006.7	1.00E+12	3600	11.11%
		DND	20336	21309	23365.2	3600	11.29%
		SDND	20418.3	21301.9	22923.4	3600	10.93%

Table A.16: Test results for complete models (instance: kroB100)

p	OPT	Model	LR	LB	UB	Time (s)	GAP (%)
kroC100 ($p' = 13$)							
10	19923.3	DNC	19703.3	19892.2	19923.3	3600	1.10%
		SDNC-	19703.3	19923.3	19923.3	2976.9	1.10%
		SDNC	19703.3	19923.3	19923.3	1663	1.10%
		DNC(SB)	19703.3	19895.5	19923.3	3600	1.10%
		SDNC-(SB)	19703.3	19923.3	19923.3	3343.6	1.10%
		SDNC(SB)	19703.3	19902.3	19923.3	3600	1.10%
		DND	19703.3	19891.9	19923.3	3600	1.10%
		SDND	19703.3	19923.3	19923.3	1802.1	1.10%
14	19938.8	DNC	19703.3	19891.8	19938.8	3600	1.18%
		SDNC-	19703.3	19938.8	19938.8	562.2	1.18%
		SDNC	19703.3	19938.8	19938.8	1222.5	1.18%
		DNC(SB)	19703.3	19938.8	19938.8	2487.7	1.18%
		SDNC-(SB)	19703.3	19938.8	19938.8	2073.4	1.18%
		SDNC(SB)	19703.3	19938.8	19938.8	1006.1	1.18%
		DND	19703.3	19938.8	19938.8	304.7	1.18%
		SDND	19703.3	19938.8	19938.8	602.7	1.18%
20	20135	DNC	19703.3	19860.5	20841.3	3600	2.14%
		SDNC-	19703.3	19976.9	20135	3600	2.14%
		SDNC	19703.3	19885.6	20135	3600	2.14%
		DNC(SB)	19703.3	19860.5	20292.3	3600	2.14%
		SDNC-(SB)	19703.3	19946.1	20135	3600	2.14%
		SDNC(SB)	19703.3	19903.6	20135	3600	2.14%
		DND	19703.3	19991	20135	3600	2.14%
		SDND	19703.5	20021.6	20135	3600	2.14%
25	20428	DNC	19703.3	19860.7	20466.6	3600	3.55%
		SDNC-	19703.3	19949.8	20428	3600	3.55%
		SDNC	19703.3	19941.4	20428	3600	3.55%
		DNC(SB)	19703.3	19860.5	20450.9	3600	3.55%
		SDNC-(SB)	19703.3	19967.9	20450.9	3600	3.55%
		SDNC(SB)	19703.9	19947.9	20469.3	3600	3.54%
		DND	19703.3	20068.5	20428	3600	3.55%
		SDND	19710.6	20123.1	20428	3600	3.51%
33	22465.73*	DNC	19703.3	19910.9	3.00E+12	3600	12.30%
		SDNC-	19704.1	20162.8	25151.2	3600	12.29%
		SDNC	19720.9	20481.6	43844	3600	12.22%
		DNC(SB)	19703.3	19894.9	2.00E+12	3600	12.30%
		SDNC-(SB)	19711.2	20219.5	1.00E+12	3600	12.26%
		SDNC(SB)	19726	20266.5	37156.3	3600	12.20%
		DND	19703.7	20594.8	25947	3600	12.29%
		SDND	19753.7	20537.2	28482.4	3600	12.07%

Table A.17: Test results for complete models (instance: kroC100)

p	OPT	Model	LR	LB	UB	Time (s)	GAP (%)
		kroD100 ($p' = 14$)					
10	20270.6	DNC	19951.3	20270.6	20270.6	780.2	1.58%
		SDNC-	19951.3	20270.6	20270.6	134.9	1.58%
		SDNC	19951.3	20270.6	20270.6	104.3	1.58%
		DNC(SB)	19951.3	20270.6	20270.6	1016.6	1.58%
		SDNC-(SB)	19951.3	20270.6	20270.6	55.5	1.58%
		SDNC(SB)	19951.3	20270.6	20270.6	75.4	1.58%
		DND	19951.3	20270.6	20270.6	17.1	1.58%
		SDND	19951.3	20270.6	20270.6	39.4	1.58%
14	20267.2	DNC	19951.3	20267.2	20267.2	928.4	1.56%
		SDNC-	19951.3	20267.2	20267.2	24.1	1.56%
		SDNC	19951.3	20267.2	20267.2	42.8	1.56%
		DNC(SB)	19951.3	20267.2	20267.2	396.8	1.56%
		SDNC-(SB)	19951.3	20267.2	20267.2	53.9	1.56%
		SDNC(SB)	19951.3	20267.2	20267.2	58.1	1.56%
		DND	19951.3	20267.2	20267.2	3.4	1.56%
		SDND	19951.3	20267.2	20267.2	18	1.56%
20	20457	DNC	19951.3	20267.2	20469	3600	2.47%
		SDNC-	19951.3	20327.7	20469	3600	2.47%
		SDNC	19951.3	20323.3	20457	3600	2.47%
		DNC(SB)	19951.3	20267.2	20469	3600	2.47%
		SDNC-(SB)	19951.3	20310.8	20457	3600	2.47%
		SDNC(SB)	19951.4	20321.7	20469	3600	2.47%
		DND	19951.3	20361.5	20457	3600	2.47%
		SDND	19952.2	20366.9	20457	3600	2.47%
25	20671.2	DNC	19951.3	20267.2	20761.9	3600	3.48%
		SDNC-	19951.3	20294.3	59378.8	3600	3.48%
		SDNC	19952.3	20325.6	20671.2	3600	3.48%
		DNC(SB)	19951.3	20267.2	20671.2	3600	3.48%
		SDNC-(SB)	19951.3	20350	20671.2	3600	3.48%
		SDNC(SB)	19953.1	20338.8	20671.2	3600	3.47%
		DND	19951.6	20470.7	20671.2	3600	3.48%
		SDND	19955.6	20436.2	20671.2	3600	3.46%
33	22238.56*	DNC	19951.3	20287.5	2.00E+12	3600	10.29%
		SDNC-	19951.3	20420.1	1.00E+12	3600	10.29%
		SDNC	19965.7	20677.8	54850.1	3600	10.22%
		DNC(SB)	19951.3	20288.9	1.00E+12	3600	10.29%
		SDNC-(SB)	19953.1	20479.8	1.00E+12	3600	10.28%
		SDNC(SB)	19970.6	20604.5	29562.4	3600	10.20%
		DND	19952.9	20873.6	23427.3	3600	10.28%
		SDND	19997.8	20753.3	26002.5	3600	10.08%

Table A.18: Test results for complete models (instance: kroD100)

p	OPT	Model	LR	LB	UB	Time (s)	GAP (%)
kroE100 ($p' = 12$)							
10	20766.4	DNC	20618.9	20766.4	20766.4	850.2	0.71%
		SDNC-	20618.9	20766.4	20766.4	345.7	0.71%
		SDNC	20618.9	20766.4	20766.4	510.3	0.71%
		DNC(SB)	20618.9	20766.4	20766.4	164.5	0.71%
		SDNC-(SB)	20618.9	20766.4	20766.4	347.3	0.71%
		SDNC(SB)	20618.9	20766.4	20766.4	299.7	0.71%
		DND	20618.9	20766.4	20766.4	316.4	0.71%
		SDND	20618.9	20766.4	20766.4	248.9	0.71%
14	20777.7	DNC	20618.9	20777.7	20777.7	1581.4	0.76%
		SDNC-	20618.9	20777.7	20777.7	108.2	0.76%
		SDNC	20618.9	20777.7	20777.7	32.4	0.76%
		DNC(SB)	20618.9	20777.7	20777.7	934.3	0.76%
		SDNC-(SB)	20618.9	20777.7	20777.7	152.6	0.76%
		SDNC(SB)	20618.9	20777.7	20777.7	43.5	0.76%
		DND	20618.9	20777.7	20777.7	15.7	0.76%
		SDND	20618.9	20777.7	20777.7	12.6	0.76%
20	20937.4	DNC	20618.9	20750.9	20937.4	3600	1.52%
		SDNC-	20618.9	20937.4	20937.4	2455.1	1.52%
		SDNC	20618.9	20937.4	20937.4	2391.2	1.52%
		DNC(SB)	20618.9	20750.9	20944	3600	1.52%
		SDNC-(SB)	20618.9	20937.4	20937.4	1915.4	1.52%
		SDNC(SB)	20618.9	20885.2	20937.4	3600	1.52%
		DND	20618.9	20937.4	20937.4	663.9	1.52%
		SDND	20618.9	20937.4	20937.4	759.9	1.52%
25	21174.9	DNC	20618.9	20754.8	21181.7	3600	2.63%
		SDNC-	20618.9	20870.9	21174.9	3600	2.63%
		SDNC	20619	20979.2	21174.9	3600	2.63%
		DNC(SB)	20618.9	20751.4	24417.6	3600	2.63%
		SDNC-(SB)	20619	20938.9	21174.9	3600	2.63%
		SDNC(SB)	20620.8	20901.2	21174.9	3600	2.62%
		DND	20618.9	21083.9	21174.9	3600	2.63%
		SDND	20622.4	21109	21174.9	3600	2.61%
33	22782.98	DNC	20618.9	20831.1	2.00E+12	3600	9.50%
		SDNC-	20627.6	21169.3	1.00E+12	3600	9.46%
		SDNC	20650.7	21276.2	24897	3600	9.36%
		DNC(SB)	20620.8	20848.5	2.00E+12	3600	9.50%
		SDNC-(SB)	20632.5	21053.7	1.00E+12	3600	9.44%
		SDNC(SB)	20652.6	21213.2	24170	3600	9.35%
		DND	20618.9	21508.6	35955.1	3600	9.50%
		SDND	20681.9	21528.6	26286.1	3600	9.22%

Table A.19: Test results for complete models (instance: kroE100)

p	OPT	Model	LR	LB	UB	Time (s)	GAP (%)
		rd100 ($p' = 13$)					
10	7524.07	DNC	7336.96	7524.07	7524.07	905.9	2.49%
		SDNC-	7336.96	7524.07	7524.07	1399.4	2.49%
		SDNC	7336.96	7524.07	7524.07	561.7	2.49%
		DNC(SB)	7336.96	7524.07	7524.07	1070.9	2.49%
		SDNC-(SB)	7336.96	7524.07	7524.07	970.1	2.49%
		SDNC(SB)	7336.96	7524.07	7524.07	647.1	2.49%
		DND	7336.96	7524.07	7524.07	719.1	2.49%
		SDND	7336.96	7524.07	7524.07	699.9	2.49%
14	7500.44	DNC	7336.96	7500.44	7500.44	592.2	2.18%
		SDNC-	7336.96	7500.44	7500.44	292.5	2.18%
		SDNC	7336.96	7500.44	7500.44	30.3	2.18%
		DNC(SB)	7336.96	7500.44	7500.44	397.2	2.18%
		SDNC-(SB)	7336.96	7500.44	7500.44	21.4	2.18%
		SDNC(SB)	7336.96	7500.44	7500.44	29.3	2.18%
		DND	7336.96	7500.44	7500.44	16.6	2.18%
		SDND	7336.96	7500.44	7500.44	12.4	2.18%
20	7537.98	DNC	7336.96	7500.63	7537.98	3600	2.67%
		SDNC-	7336.96	7510.55	7556.02	3600	2.67%
		SDNC	7336.96	7509.35	7556.02	3600	2.67%
		DNC(SB)	7336.96	7501.13	7571.43	3600	2.67%
		SDNC-(SB)	7336.96	7508.21	7537.98	3600	2.67%
		SDNC(SB)	7336.96	7509.41	7537.98	3600	2.67%
		DND	7336.96	7537.98	7537.98	1737.8	2.67%
		SDND	7336.96	7537.98	7537.98	2874.6	2.67%
25	7555.83	DNC	7336.96	7501.25	7555.83	3600	2.90%
		SDNC-	7336.96	7521.18	7555.83	3600	2.90%
		SDNC	7336.96	7513.84	7555.83	3600	2.90%
		DNC(SB)	7336.96	7500.58	7555.83	3600	2.90%
		SDNC-(SB)	7336.96	7525.78	7555.83	3600	2.90%
		SDNC(SB)	7336.96	7525.26	7555.83	3600	2.90%
		DND	7336.96	7555.83	7555.83	1281.5	2.90%
		SDND	7336.96	7555.83	7555.83	1636.5	2.90%
33	8131.25*	DNC	7336.96	7506.08	1.00E+12	3600	9.77%
		SDNC-	7336.96	7569.15	1.00E+12	3600	9.77%
		SDNC	7337.79	7587.86	1.00E+12	3600	9.76%
		DNC(SB)	7336.96	7508.6	3.00E+12	3600	9.77%
		SDNC-(SB)	7336.98	7550.91	1.00E+12	3600	9.77%
		SDNC(SB)	7339.19	7571.04	9499.03	3600	9.74%
		DND	7336.96	7623.47	8131.25	3600	9.77%
		SDND	7344.99	7640.96	8682.87	3600	9.67%

Table A.20: Test results for complete models (instance: rd100)

p	OPT	Model	LR	LB	UB	Time (s)	GAP (%)
		gr24 ($p' = 4$)					
6	1266	NC	1224.5	1266	1266	0.4	3.28%
		SNC-	1224.5	1266	1266	0.5	3.28%
		SNC	1224.5	1266	1266	0.4	3.28%
		NC(SB)	1224.5	1266	1266	0.6	3.28%
		SNC-(SB)	1224.5	1266	1266	0.3	3.28%
		SNC(SB)	1224.5	1266	1266	0.3	3.28%
		ND	1224.5	1266	1266	0.2	3.28%
		SND	1224.61	1266	1266	0.3	3.27%
7	1288	NC	1224.5	1288	1288	1	4.93%
		SNC-	1225.19	1288	1288	0.6	4.88%
		SNC	1225.36	1288	1288	0.6	4.86%
		NC(SB)	1224.5	1288	1288	0.9	4.93%
		SNC-(SB)	1225.12	1288	1288	0.4	4.88%
		SNC(SB)	1225.31	1288	1288	0.7	4.87%
		ND	1224.5	1288	1288	0.4	4.93%
		SND	1225.9	1288	1288	0.4	4.82%
8	1317	NC	1224.5	1317	1317	1.3	7.02%
		SNC-	1226.5	1317	1317	1.2	6.87%
		SNC	1228.06	1317	1317	0.8	6.75%
		NC(SB)	1224.5	1317	1317	3.4	7.02%
		SNC-(SB)	1227.33	1317	1317	1.3	6.81%
		SNC(SB)	1228.38	1317	1317	2.4	6.73%
		ND	1224.5	1317	1317	0.5	7.02%
		SND	1231.34	1317	1317	1.1	6.50%
		fri26 ($p' = 7$)					
6	886	NC	880	883	883	0.1	0.68%
		SNC-	880	883	883	0.1	0.68%
		SNC	880	883	883	0.1	0.68%
		NC(SB)	880	883	883	0.1	0.68%
		SNC-(SB)	880	883	883	0	0.68%
		SNC(SB)	880	883	883	0	0.68%
		ND	880	883	883	0	0.68%
		SND	880	883	883	0	0.68%
7	883	NC	880	883	883	0.1	0.34%
		SNC-	880.083	883	883	0.1	0.33%
		SNC	880.092	883	883	0.1	0.33%
		NC(SB)	880	883	883	0.1	0.34%
		SNC-(SB)	880.083	883	883	0	0.33%
		SNC(SB)	880.141	883	883	0	0.32%
		ND	880	883	883	0	0.34%
		SND	880.079	883	883	0	0.33%
8	885	NC	880	885	885	0.5	0.56%
		SNC-	881.286	885	885	0.1	0.42%
		SNC	881.474	885	885	0.1	0.40%
		NC(SB)	880	885	885	0.4	0.56%
		SNC-(SB)	881.4	885	885	0.1	0.41%
		SNC(SB)	881.594	885	885	0.1	0.38%
		ND	880	885	885	0	0.56%
		SND	881.019	885	885	0.1	0.45%

Table A.21: Test results for incomplete models (instances: gr24, fri26)

p	OPT	Model	LR	LB	UB	Time (s)	GAP (%)
bayg29 ($p' = 3$)							
7	1618	NC	1546	1618	1618	2.6	4.45%
		SNC-	1546	1618	1618	2.1	4.45%
		SNC	1546	1618	1618	2.5	4.45%
		NC(SB)	1546	1618	1618	3.3	4.45%
		SNC-(SB)	1546	1618	1618	2.2	4.45%
		SNC(SB)	1546.03	1618	1618	2.1	4.45%
		ND	1546	1618	1618	1.8	4.45%
		SND	1546.26	1618	1618	2	4.43%
8	1642	NC	1546	1642	1642	5.2	5.85%
		SNC-	1546	1642	1642	3.2	5.85%
		SNC	1546	1642	1642	2.6	5.85%
		NC(SB)	1546	1642	1642	6.5	5.85%
		SNC-(SB)	1546.17	1642	1642	5	5.84%
		SNC(SB)	1546.41	1642	1642	2.3	5.82%
		ND	1546	1642	1642	2.8	5.85%
		SND	1546.64	1642	1642	1.5	5.81%
9	1676	NC	1546	1676	1676	7.5	7.76%
		SNC-	1546	1676	1676	5.2	7.76%
		SNC	1546	1676	1676	5.4	7.76%
		NC(SB)	1546	1676	1676	13.3	7.76%
		SNC-(SB)	1546.56	1676	1676	4	7.72%
		SNC(SB)	1546.97	1676	1676	7.3	7.70%
		ND	1546.09	1676	1676	3.9	7.75%
		SND	1547.61	1676	1676	3.5	7.66%
swiss42 ($p' = 7$)							
10	1238	NC	1214.5	1238	1238	1.5	1.90%
		SNC-	1214.5	1238	1238	1	1.90%
		SNC	1214.5	1238	1238	0.9	1.90%
		NC(SB)	1214.5	1238	1238	1	1.90%
		SNC-(SB)	1214.5	1238	1238	0.7	1.90%
		SNC(SB)	1214.5	1238	1238	0.9	1.90%
		ND	1214.5	1238	1238	0.8	1.90%
		SND	1214.5	1238	1238	0.3	1.90%
12	1256	NC	1214.5	1256	1256	8.1	3.30%
		SNC-	1214.5	1256	1256	2.7	3.30%
		SNC	1214.5	1256	1256	2.9	3.30%
		NC(SB)	1214.5	1256	1256	3.9	3.30%
		SNC-(SB)	1214.5	1256	1256	4.2	3.30%
		SNC(SB)	1214.5	1256	1256	1.5	3.30%
		ND	1214.5	1256	1256	0.7	3.30%
		SND	1215.28	1256	1256	1.4	3.24%
14	1292	NC	1214.5	1292	1292	65.8	6.00%
		SNC-	1214.5	1292	1292	19.9	6.00%
		SNC	1215.56	1292	1292	31.5	5.92%
		NC(SB)	1214.5	1292	1292	17.6	6.00%
		SNC-(SB)	1214.67	1292	1292	26.5	5.99%
		SNC(SB)	1215.89	1292	1292	18.9	5.89%
		ND	1214.5	1292	1292	35.6	6.00%
		SND	1219.87	1292	1292	9.6	5.58%

Table A.22: Test results for incomplete models (instances: bayg29, swiss42)

p	OPT	Model	LR	LB	UB	Time (s)	GAP (%)
		att48 ($p' = 5$)					
12	32742.9	NC	31669.3	32742.9	32742.9	7.9	3.28%
		SNC-	31669.3	32742.9	32742.9	8	3.28%
		SNC	31669.5	32742.9	32742.9	10.7	3.28%
		NC(SB)	31669.3	32742.9	32742.9	11	3.28%
		SNC-(SB)	31669.3	32742.9	32742.9	8	3.28%
		SNC(SB)	31669.8	32742.9	32742.9	11	3.28%
		ND	31669.3	32742.9	32742.9	2.4	3.28%
		SND	31671.1	32742.9	32742.9	2.1	3.27%
14	33679.2	NC	31669.3	33679.2	33679.2	132.6	5.97%
		SNC-	31669.3	33679.2	33679.2	86.4	5.97%
		SNC	31675.8	33679.2	33679.2	65.8	5.95%
		NC(SB)	31669.3	33679.2	33679.2	79.3	5.97%
		SNC-(SB)	31669.5	33679.2	33679.2	94.5	5.97%
		SNC(SB)	31679.6	33679.2	33679.2	98.4	5.94%
		ND	31669.4	33679.2	33679.2	13.6	5.97%
		SND	31696.5	33679.2	33679.2	22.8	5.89%
16	37068.82	NC	31669.3	34664	37329.1	3600	14.57%
		SNC-	31669.3	34941.8	37068.8	3600	14.57%
		SNC	31693.9	35092.5	37329.1	3600	14.50%
		NC(SB)	31669.3	OUT OF MEMORY			14.57%
		SNC-(SB)	31673.7	34532.2	37878.2	3600	14.55%
		SNC(SB)	31710.8	34680.6	37329.1	3600	14.45%
		ND	31669.7	35314.1	37561.9	3600	14.57%
		SND	31780.7	35263.2	37561.9	3600	14.27%
		gr48 ($p' = 6$)					
12	5011	NC	4769	5011	5011	48.8	4.83%
		SNC-	4769	5011	5011	28.2	4.83%
		SNC	4769	5011	5011	20.8	4.83%
		NC(SB)	4769	5011	5011	44.6	4.83%
		SNC-(SB)	4769	5011	5011	59.4	4.83%
		SNC(SB)	4769	5011	5011	35.5	4.83%
		ND	4769	5011	5011	8.2	4.83%
		SND	4769	5011	5011	8.5	4.83%
14	5120	NC	4769	5120	5120	109.9	6.86%
		SNC-	4769	5120	5120	66.4	6.86%
		SNC	4769	5120	5120	80.8	6.86%
		NC(SB)	4769	5120	5120	129.9	6.86%
		SNC-(SB)	4769	5120	5120	94.8	6.86%
		SNC(SB)	4769	5120	5120	73.5	6.86%
		ND	4769	5120	5120	30	6.86%
		SND	4769.31	5120	5120	22.4	6.85%
16	5445	NC	4769	5445	5445	2885.2	12.42%
		SNC-	4769	5445	5445	1250.1	12.42%
		SNC	4769	5445	5445	1219	12.42%
		NC(SB)	4769	5445	5445	2030.9	12.42%
		SNC-(SB)	4769	5340.56	5445	3600	12.42%
		SNC(SB)	4769.49	5445	5445	1156.7	12.41%
		ND	4769	5445	5445	1191.5	12.42%
		SND	4776.57	5445	5445	989.6	12.28%

Table A.23: Test results for incomplete models (instances: att48, gr48)

p	OPT	Model	LR	LB	UB	Time (s)	GAP (%)
		hk48 ($p' = 6$)					
12	11450	NC	11197	11450	11450	13.4	2.21%
		SNC-	11197	11450	11450	10.8	2.21%
		SNC	11197	11450	11450	6.8	2.21%
		NC(SB)	11197	11450	11450	15.4	2.21%
		SNC-(SB)	11197	11450	11450	6.5	2.21%
		SNC(SB)	11197	11450	11450	7.1	2.21%
		ND	11197	11450	11450	1.3	2.21%
		SND	11197.1	11450	11450	1.4	2.21%
14	11600	NC	11197	11600	11600	26.3	3.47%
		SNC-	11197	11600	11600	29	3.47%
		SNC	11197	11600	11600	10.2	3.47%
		NC(SB)	11197	11600	11600	29.6	3.47%
		SNC-(SB)	11197	11600	11600	35	3.47%
		SNC(SB)	11197.1	11600	11600	8.6	3.47%
		ND	11197	11600	11600	3.4	3.47%
		SND	11197.4	11600	11600	8.4	3.47%
16	12215	NC	11197	12215	12215	1289.1	8.33%
		SNC-	11197	12215	12215	417.2	8.33%
		SNC	11197.2	12215	12215	282.6	8.33%
		NC(SB)	11197	12215	12215	1400.2	8.33%
		SNC-(SB)	11198.5	12215	12215	793.5	8.32%
		SNC(SB)	11199.7	12215	12215	953.4	8.31%
		ND	11197	12215	12215	369.8	8.33%
		SND	11202.7	12215	12215	493.6	8.29%
		eil51 ($p' = 3$)					
12	436.587	NC	418.686	436.585	436.585	56.7	4.10%
		SNC-	418.686	436.585	436.585	37	4.10%
		SNC	418.838	436.585	436.585	27.8	4.07%
		NC(SB)	418.686	436.585	436.585	97.2	4.10%
		SNC-(SB)	418.775	436.585	436.585	96.7	4.08%
		SNC(SB)	419.056	436.585	436.585	31.8	4.02%
		ND	418.87	436.585	436.585	10.3	4.06%
		SND	419.278	436.585	436.585	14.1	3.96%
15	445.925	NC	418.686	445.925	445.925	534.2	6.11%
		SNC-	418.686	445.925	445.925	189	6.11%
		SNC	419.212	445.925	445.925	373.3	5.99%
		NC(SB)	418.686	445.925	445.925	648.9	6.11%
		SNC-(SB)	418.999	445.925	445.925	1595	6.04%
		SNC(SB)	419.497	445.925	445.925	337.8	5.93%
		ND	419.108	445.925	445.925	61.2	6.01%
		SND	419.791	445.925	445.925	93.2	5.86%
17	473.977	NC	418.686	451.19	490.556	3600	11.67%
		SNC-	418.755	466.974	478.026	3600	11.65%
		SNC	419.685	468.417	475.886	3600	11.45%
		NC(SB)	418.686	465.885	475.557	3600	11.67%
		SNC-(SB)	419.204	OUT OF MEMORY			11.56%
		SNC(SB)	419.992	460.94	475.886	3600	11.39%
		ND	419.319	462.975	475.557	3600	11.53%
		SND	420.249	473.978	473.978	3275.4	11.34%

Table A.24: Test results for incomplete models (instances: hk48, eil51)

p	OPT	Model	LR	LB	UB	Time (s)	GAP (%)
berlin52 ($p' = 7$)							
13	7298.63	NC	7166.3	7298.63	7298.63	11.5	1.81%
		SNC-	7166.3	7298.63	7298.63	4.6	1.81%
		SNC	7166.4	7298.63	7298.63	6.6	1.81%
		NC(SB)	7166.3	7298.63	7298.63	24.5	1.81%
		SNC-(SB)	7166.32	7298.63	7298.63	7.4	1.81%
		SNC(SB)	7166.43	7298.63	7298.63	7.7	1.81%
		ND	7166.3	7298.63	7298.63	1.4	1.81%
		SND	7166.46	7298.63	7298.63	2.2	1.81%
15	7429.59	NC	7166.3	7429.59	7429.59	76.3	3.54%
		SNC-	7166.3	7429.59	7429.59	51.3	3.54%
		SNC	7166.52	7429.59	7429.59	14	3.54%
		NC(SB)	7166.3	7429.59	7429.59	133.5	3.54%
		SNC-(SB)	7166.38	7429.59	7429.59	17.7	3.54%
		SNC(SB)	7166.55	7429.59	7429.59	14.7	3.54%
		ND	7166.31	7429.59	7429.59	8.4	3.54%
		SND	7167.45	7429.59	7429.59	12	3.53%
17	7800.77	NC	7166.3	7716.85	7804.09	3600	8.13%
		SNC-	7166.48	7800.77	7800.77	527.8	8.13%
		SNC	7168.01	7800.77	7800.77	368.8	8.11%
		NC(SB)	7166.3	7800.77	7800.77	1226.5	8.13%
		SNC-(SB)	7166.65	7800.77	7800.77	818.8	8.13%
		SNC(SB)	7168.34	7800.77	7800.77	1601.2	8.11%
		ND	7166.33	7800.77	7800.77	302.2	8.13%
		SND	7173.26	7800.77	7800.77	532.6	8.04%
brazil58 ($p' = 12$)							
14	21221	NC	20896	21221	21221	2.9	1.53%
		SNC-	20896	21221	21221	1.8	1.53%
		SNC	20896	21221	21221	1.5	1.53%
		NC(SB)	20896	21221	21221	4.1	1.53%
		SNC-(SB)	20896	21221	21221	2.2	1.53%
		SNC(SB)	20896	21221	21221	2.5	1.53%
		ND	20896	21221	21221	0.9	1.53%
		SND	20896	21221	21221	1	1.53%
17	21847	NC	20896	21847	21847	565.4	4.35%
		SNC-	20896	21847	21847	161.1	4.35%
		SNC	20896	21847	21847	287.3	4.35%
		NC(SB)	20896	21847	21847	700.6	4.35%
		SNC-(SB)	20896	21847	21847	1283.8	4.35%
		SNC(SB)	20897.8	21847	21847	337.7	4.34%
		ND	20896	21847	21847	58.4	4.35%
		SND	20904	21847	21847	49.5	4.32%
19	22635	NC	20896	22635	22635	1462	7.68%
		SNC-	20896	22635	22635	960.7	7.68%
		SNC	20900.6	22635	22635	228.2	7.66%
		NC(SB)	20896	22635	22635	1351.6	7.68%
		SNC-(SB)	20896.7	22131.8	109324	3600	7.68%
		SNC(SB)	20909.1	22160.7	98619	3600	7.62%
		ND	20896	22635	22635	458.4	7.68%
		SND	20924.6	22635	22635	633.4	7.56%

Table A.25: Test results for incomplete models (instances: berlin52, brazil58)

p	OPT	Model	LR	LB	UB	Time (s)	GAP (%)
		st70 ($p' = 12$)					
17	636.194	NC	628.5	636.191	636.191	11.8	1.21%
		SNC-	628.5	636.191	636.191	10.6	1.21%
		SNC	628.5	636.191	636.191	5	1.21%
		NC(SB)	628.5	636.191	636.191	16.9	1.21%
		SNC-(SB)	628.5	636.191	636.191	2.1	1.21%
		SNC(SB)	628.5	636.191	636.191	2.5	1.21%
		ND	628.5	636.191	636.191	2.3	1.21%
		SND	628.516	636.191	636.191	1.3	1.21%
20	645.145	NC	628.5	645.145	645.145	161.3	2.58%
		SNC-	628.5	645.145	645.145	60.9	2.58%
		SNC	628.5	645.145	645.145	32.6	2.58%
		NC(SB)	628.5	645.145	645.145	55.7	2.58%
		SNC-(SB)	628.5	645.145	645.145	43.7	2.58%
		SNC(SB)	628.544	645.145	645.145	34.3	2.57%
		ND	628.5	645.145	645.145	12	2.58%
		SND	628.671	645.145	645.145	3.2	2.55%
23	694.495	NC	628.5	663.444	2006.28	3600	9.50%
		SNC-	628.5	668.257	3285.09	3600	9.50%
		SNC	628.669	662.392	3073.84	3600	9.48%
		NC(SB)	628.5	656.746	3292.69	3600	9.50%
		SNC-(SB)	628.626	663.857	3044.02	3600	9.48%
		SNC(SB)	628.718	660.568	3092.89	3600	9.47%
		ND	628.5	673.009	694.493	3600	9.50%
		SND	629.601	664.423	2812.45	3600	9.34%
		eil76 ($p' = 4$)					
19	563.955	NC	540.723	554.461	563.955	3600	4.12%
		SNC-	540.723	563.955	563.955	3098.9	4.12%
		SNC	540.723	563.955	563.955	2807.9	4.12%
		NC(SB)	540.723	563.955	563.955	1973.2	4.12%
		SNC-(SB)	540.723	553.662	894.548	3600	4.12%
		SNC(SB)	540.739	556.987	564.243	3600	4.12%
		ND	540.723	563.955	563.955	612.9	4.12%
		SND	540.89	563.955	563.955	1022	4.09%
22	573.182	NC	540.723	557.346	586.244	3600	5.66%
		SNC-	540.723	562.153	958.698	3600	5.66%
		SNC	540.749	565.879	573.182	3600	5.66%
		NC(SB)	540.723	564.593	575.923	3600	5.66%
		SNC-(SB)	540.727	560.964	1385.38	3600	5.66%
		SNC(SB)	540.871	561.837	1213.99	3600	5.64%
		ND	540.73	570.106	573.182	3600	5.66%
		SND	541.212	569.056	573.182	3600	5.58%
25	601.71	NC	540.723	563.269	2099.68	3600	10.14%
		SNC-	540.723	572.766	1831.77	3600	10.14%
		SNC	541.091	573.6	1932.28	3600	10.07%
		NC(SB)	540.723	561.768	1978.3	3600	10.14%
		SNC-(SB)	540.738	568.388	2006.14	3600	10.13%
		SNC(SB)	541.273	571.345	1790.53	3600	10.04%
		ND	540.803	571.329	1797.7	3600	10.12%
		SND	541.586	571.517	1904.49	3600	9.99%

Table A.26: Test results for incomplete models (instances: st70, eil76)

p	OPT	Model	LR	LB	UB	Time (s)	GAP (%)
pr76 ($p' = 8$)							
19	104482	NC	98993.9	104482	104482	160.1	5.25%
		SNC-	99044.7	104482	104482	37.2	5.20%
		SNC	99143.5	104482	104482	46.1	5.11%
		NC(SB)	98993.9	104482	104482	113	5.25%
		SNC-(SB)	99072.2	104482	104482	39	5.18%
		SNC(SB)	99163.4	104482	104482	41.1	5.09%
		ND	98998.5	104482	104482	18.6	5.25%
		SND	99241.4	104482	104482	16.5	5.02%
22	105996	NC	98993.9	105996	105996	168.3	6.61%
		SNC-	99160.6	105996	105996	143.3	6.45%
		SNC	99277.4	105996	105996	134.3	6.34%
		NC(SB)	98993.9	105996	105996	491.6	6.61%
		SNC-(SB)	99207.3	105996	105996	77.1	6.40%
		SNC(SB)	99291	105996	105996	136.7	6.33%
		ND	99014.4	105996	105996	28.2	6.59%
		SND	99405	105996	105996	20.1	6.22%
25	110074	NC	98993.9	108262	110300	3600	10.07%
		SNC-	99292.6	110074	110074	1674.3	9.79%
		SNC	99463.1	110074	110074	2285.4	9.64%
		NC(SB)	98993.9	108831	110074	3600	10.07%
		SNC-(SB)	99361.9	108030	178261	3600	9.73%
		SNC(SB)	99477.8	110074	110074	1314.3	9.63%
		ND	99030.3	110074	110074	2383.9	10.03%
		SND	99742.8	108783	110074	3600	9.39%
rat99 ($p' = 8$)							
24	1273.23	NC	1203.53	1273.23	1273.23	1560.6	5.47%
		SNC-	1204.43	1273.23	1273.23	398	5.40%
		SNC	1204.74	1273.23	1273.23	363.7	5.38%
		NC(SB)	1203.53	1254.19	4083.85	3600	5.47%
		SNC-(SB)	1204.38	1273.23	1273.23	413.1	5.41%
		SNC(SB)	1204.75	1273.23	1273.23	396.1	5.38%
		ND	1203.53	1273.23	1273.23	70.6	5.47%
		SND	1205.3	1273.23	1273.23	135.9	5.34%
29	1311.42	NC	1203.53	1291.45	1313	3600	8.23%
		SNC-	1207.6	1311.42	1311.42	873.1	7.92%
		SNC	1208.49	1311.42	1311.42	1139.1	7.85%
		NC(SB)	1203.53	1291.25	1315	3600	8.23%
		SNC-(SB)	1207.29	1300.34	1311.42	3600	7.94%
		SNC(SB)	1208.15	1311.42	1311.42	794.7	7.87%
		ND	1203.53	1311.42	1311.42	386.9	8.23%
		SND	1210.05	1311.42	1311.42	196.2	7.73%
33	1373.37	NC	1203.53	1314.25	1522.48	3600	12.37%
		SNC-	1213.93	1329.28	1486.28	3600	11.61%
		SNC	1215.37	1331.79	1522.48	3600	11.50%
		NC(SB)	1203.53	1307.27	1514.99	3600	12.37%
		SNC-(SB)	1213.01	OUT OF MEMORY			11.68%
		SNC(SB)	1214.65	1334.58	1463.57	3600	11.56%
		ND	1203.53	1331.07	1476.96	3600	12.37%
		SND	1217.58	1330.04	1434	3600	11.34%

Table A.27: Test results for incomplete models (instances: pr76, rat99)

p	OPT	Model	LR	LB	UB	Time (s)	GAP (%)
kroA100 ($p' = 13$)							
25	20279.5	NC	19380.7	20279.5	20279.5	2790.4	4.43%
		SNC-	19380.7	20279.5	20279.5	2272.5	4.43%
		SNC	19380.7	20279.5	20279.5	1499.5	4.43%
		NC(SB)	19380.7	20279.5	20279.5	3027.5	4.43%
		SNC-(SB)	19380.7	20279.5	20279.5	2001.8	4.43%
		SNC(SB)	19380.7	20279.5	20279.5	644.1	4.43%
		ND	19380.7	20279.5	20279.5	22.8	4.43%
		SND	19383.9	20279.5	20279.5	21.4	4.42%
29	20773.3	NC	19380.7	20337.6	49438	3600	6.70%
		SNC-	19380.7	20467.9	101552	3600	6.70%
		SNC	19380.7	20773.3	20773.3	1779.6	6.70%
		NC(SB)	19380.7	20312.7	52410.7	3600	6.70%
		SNC-(SB)	19381.4	OUT OF MEMORY			6.70%
		SNC(SB)	19389.8	20426.2	21099.3	3600	6.66%
		ND	19380.7	20773.3	20773.3	394.3	6.70%
		SND	19411	20681.2	20773.3	3600	6.56%
33	22303.23*	NC	19380.7	20685.7	165276	3600	13.10%
		SNC-	19380.7	20890.2	158225	3600	13.10%
		SNC	19411.7	20948.1	159356	3600	12.96%
		NC(SB)	19380.7	20673.4	153253	3600	13.10%
		SNC-(SB)	19383.7	OUT OF MEMORY			13.09%
		SNC(SB)	19418.2	20926.1	132250	3600	12.94%
		ND	19380.7	21015.3	156051	3600	13.10%
		SND	19478.9	21057.4	160746	3600	12.66%
kroB100 ($p' = 20$)							
25	20786.9	NC	20336	20786.9	20786.9	301.7	2.17%
		SNC-	20336	20786.9	20786.9	461.7	2.17%
		SNC	20336	20786.9	20786.9	57.3	2.17%
		NC(SB)	20336	20786.9	20786.9	727.5	2.17%
		SNC-(SB)	20336.1	20786.9	20786.9	228.4	2.17%
		SNC(SB)	20336.1	20786.9	20786.9	33	2.17%
		ND	20336	20786.9	20786.9	16.6	2.17%
		SND	20338	20786.9	20786.9	9.5	2.16%
29	21094.6	NC	20336	20933.1	106271	3600	3.60%
		SNC-	20336	20955	101498	3600	3.60%
		SNC	20338.2	21094.6	21094.6	694.7	3.59%
		NC(SB)	20336	20979.8	122075	3600	3.60%
		SNC-(SB)	20339.4	21094.6	21094.6	689.2	3.58%
		SNC(SB)	20344.1	20957.4	123257	3600	3.56%
		ND	20336	21094.6	21094.6	174	3.60%
		SND	20356.2	21094.6	21094.6	59	3.50%
33	22923.4*	NC	20336	21234.9	136378	3600	11.29%
		SNC-	20346.6	21284.8	142074	3600	11.24%
		SNC	20372.3	21319.8	148805	3600	11.13%
		NC(SB)	20336	21272.8	140267	3600	11.29%
		SNC-(SB)	20356.8	OUT OF MEMORY			11.20%
		SNC(SB)	20376.3	21249.8	146709	3600	11.11%
		ND	20336	21374.7	141884	3600	11.29%
		SND	20418.3	21674.2	138607	3600	10.93%

Table A.28: Test results for incomplete models (instances: kroA100, kroB100)

p	OPT	Model	LR	LB	UB	Time (s)	GAP (%)
kroC100 ($p' = 13$)							
25	20428	NC	19703.3	20077.4	20469.3	3600	3.55%
		SNC-	19703.3	20118.5	20428	3600	3.55%
		SNC	19703.3	20092.4	20428	3600	3.55%
		NC(SB)	19703.3	20088.1	20428	3600	3.55%
		SNC-(SB)	19703.3	20031.5	118761	3600	3.55%
		SNC(SB)	19703.9	20072.9	20428	3600	3.54%
		ND	19703.3	20328.8	20428	3600	3.55%
		SND	19710.6	20428	20428	2902.6	3.51%
29	20923.6	NC	19703.3	20273.9	149110	3600	5.83%
		SNC-	19703.3	20202.4	133042	3600	5.83%
		SNC	19707.7	20295.1	145279	3600	5.81%
		NC(SB)	19703.3	20199.5	140783	3600	5.83%
		SNC-(SB)	19703.4	20644	20923.8	3600	5.83%
		SNC(SB)	19711.8	20361.9	36260	3600	5.79%
		ND	19703.3	20343.8	128569	3600	5.83%
		SND	19725.9	20423.1	20923.8	3600	5.72%
33	22465.73*	NC	19703.3	20569.7	179980	3600	12.30%
		SNC-	19704.1	20532.2	175156	3600	12.29%
		SNC	19720.9	20653.3	176827	3600	12.22%
		NC(SB)	19703.3	20521.1	155538	3600	12.30%
		SNC-(SB)	19711.2	OUT OF MEMORY			12.26%
		SNC(SB)	19726	20589.4	172457	3600	12.20%
		ND	19703.7	20776.4	106155	3600	12.29%
		SND	19753.7	21051.6	165446	3600	12.07%
kroD100 ($p' = 14$)							
25	20671.2	NC	19951.3	20671.2	20671.2	895.9	3.48%
		SNC-	19951.3	20671.2	20671.2	1499.8	3.48%
		SNC	19952.3	20671.2	20671.2	357.5	3.48%
		NC(SB)	19951.3	20671.2	20671.2	571.2	3.48%
		SNC-(SB)	19951.3	20671.2	20671.2	1862.5	3.48%
		SNC(SB)	19953.1	20671.2	20671.2	418.8	3.47%
		ND	19951.6	20671.2	20671.2	42.9	3.48%
		SND	19955.6	20671.2	20671.2	31.7	3.46%
29	21043.2	NC	19951.3	20663.2	21060.8	3600	5.19%
		SNC-	19951.3	20746.8	111253	3600	5.19%
		SNC	19954.5	20776.2	90719.4	3600	5.17%
		NC(SB)	19951.3	20695.6	103147	3600	5.19%
		SNC-(SB)	19951.5	20819.7	21043.2	3600	5.19%
		SNC(SB)	19956.9	20749.3	98707.8	3600	5.16%
		ND	19952.1	21043.2	21043.2	2067	5.19%
		SND	19969.1	21043.2	21043.2	2926.7	5.10%
33	22238.56*	NC	19951.3	20946.8	141789	3600	10.29%
		SNC-	19951.3	21054.5	150113	3600	10.29%
		SNC	19965.7	21125.1	157067	3600	10.22%
		NC(SB)	19951.3	20850.3	156280	3600	10.29%
		SNC-(SB)	19953.1	OUT OF MEMORY			10.28%
		SNC(SB)	19970.6	21081.1	126916	3600	10.20%
		ND	19952.9	21205	136323	3600	10.28%
		SND	19997.8	21126.4	153238	3600	10.08%

Table A.29: Test results for incomplete models (instances: kroC100, kroD100)

p	OPT	Model	LR	LB	UB	Time (s)	GAP (%)
kroE100 ($p' = 12$)							
25	21174.9	NC	20618.9	21016.5	21174.9	3600	2.63%
		SNC-	20618.9	21174.9	21174.9	1908	2.63%
		SNC	20619	20969.9	98689.7	3600	2.63%
		NC(SB)	20618.9	21174.9	21174.9	3293.2	2.63%
		SNC-(SB)	20619	20982.6	117409	3600	2.63%
		SNC(SB)	20620.8	20943.7	106210	3600	2.62%
		ND	20618.9	21174.9	21174.9	216.8	2.63%
		SND	20622.4	21174.9	21174.9	281.6	2.61%
29	21386.1	NC	20618.9	21158.4	125844	3600	3.59%
		SNC-	20618.9	21204.5	148424	3600	3.59%
		SNC	20628.1	21160.9	136671	3600	3.54%
		NC(SB)	20618.9	21202.9	51613.9	3600	3.59%
		SNC-(SB)	20621	21386.1	21386.1	763.7	3.58%
		SNC(SB)	20630.2	21157.1	74431.2	3600	3.53%
		ND	20618.9	21386.1	21386.1	255.2	3.59%
		SND	20641.5	21386.1	21386.1	313.6	3.48%
33	22782.98	NC	20618.9	21383.6	164284	3600	9.50%
		SNC-	20627.6	21414.8	172007	3600	9.46%
		SNC	20650.7	21598.3	176203	3600	9.36%
		NC(SB)	20618.9	21338.9	165824	3600	9.50%
		SNC-(SB)	20632.5	OUT OF MEMORY			9.44%
		SNC(SB)	20652.6	21481	173693	3600	9.35%
		ND	20618.9	21643.5	169903	3600	9.50%
		SND	20681.9	21946.2	159909	3600	9.22%
rd100 ($p' = 13$)							
25	7555.83	NC	7336.96	7555.83	7555.83	954.9	2.90%
		SNC-	7336.96	7555.83	7555.83	215.1	2.90%
		SNC	7336.96	7555.83	7555.83	81.7	2.90%
		NC(SB)	7336.96	7555.83	7555.83	502.9	2.90%
		SNC-(SB)	7336.96	7555.83	7555.83	344.2	2.90%
		SNC(SB)	7336.96	7555.83	7555.83	283.8	2.90%
		ND	7336.96	7555.83	7555.83	23.4	2.90%
		SND	7336.96	7555.83	7555.83	14.3	2.90%
29	7684.52	NC	7336.96	7583.12	36281.9	3600	4.52%
		SNC-	7336.96	7612.57	7696.21	3600	4.52%
		SNC	7336.96	7684.52	7684.52	3316	4.52%
		NC(SB)	7336.96	7649.54	7684.52	3600	4.52%
		SNC-(SB)	7336.96	7621.62	7696.21	3600	4.52%
		SNC(SB)	7336.99	7620.16	7696.21	3600	4.52%
		ND	7336.96	7684.52	7684.52	451.6	4.52%
		SND	7338.66	7684.52	7684.52	146.6	4.50%
33	8131.25*	NC	7336.96	7656.96	45970.8	3600	9.77%
		SNC-	7336.96	7706.64	49154.4	3600	9.77%
		SNC	7337.79	7747.51	41237.3	3600	9.76%
		NC(SB)	7336.96	7682.43	45821.4	3600	9.77%
		SNC-(SB)	7336.98	OUT OF MEMORY			9.77%
		SNC(SB)	7339.19	7706.42	49829.6	3600	9.74%
		ND	7336.96	7760.02	46293.8	3600	9.77%
		SND	7344.99	7748.59	46108.9	3600	9.67%

Table A.30: Test results for incomplete models (instances: kroE100, rd100)