



HAL
open science

Efficient Algorithms for Routing Problems with Specific Constraints

Daniil Khachai

► **To cite this version:**

Daniil Khachai. Efficient Algorithms for Routing Problems with Specific Constraints. Data Structures and Algorithms [cs.DS]. Université de Bordeaux, 2023. English. NNT : 2023BORD0346 . tel-04372176

HAL Id: tel-04372176

<https://theses.hal.science/tel-04372176v1>

Submitted on 4 Jan 2024

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

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

THÈSE PRÉSENTÉE

POUR OBTENIR LE GRADE DE

DOCTEUR DE

L'UNIVERSITÉ DE BORDEAUX

ÉCOLE DOCTORALE MATHÉMATIQUES ET INFORMATIQUE

MATHÉMATIQUES APPLIQUÉES ET CALCUL SCIENTIFIQUE

Par Daniil KHACHAI

**ALGORITHMES EFFICACES POUR LES PROBLÈMES DE
ROUTAGE AVEC DES CONTRAINTES SPÉCIFIQUES**

Efficient Algorithms for Routing Problems with Specific Constraints

Sous la direction de Boris DETIENNE et Olga BATTIAIA

Soutenue le 30 Novembre 2023

Membres du jury :

Mme BATTIAIA Olga	Professeur, Kedge Business School	Directrice de thèse
M. DETIENNE Boris	Maitre de conférences, Université de Bordeaux	Directeur de thèse
M. FOUILHOUX Pierre	Professeur, Université de Sorbonne Paris Nord	Rapporteur
M. SEMET Frédéric	Professeur, Université de Lille	Rapporteur
M. CLAUTIAUX François	Professeur, Université de Bordeaux	Président
M. GOUVEIA Luis Eduardo Neves	Professeur, Université de Lisbonne	Examineur
Mme WAGLER Annegret	Professeur, Université Clermont Auvergne	Examineur

Membres invités

M. SADYKOV Ruslan	Responsable de la recherche et du développement, Atoptima	invité
-------------------	---	--------

Abstract

Cette thèse se concentre sur la conception algorithmique de trois problèmes d'optimisation combinatoire liés à la recherche en transport, logistique et production avec des types spécifiques de contraintes industrielles. Tout d'abord, nous considérons le problème du voyageur de commerce généralisé à contraintes de priorité (PCGTSP). Ce problème est une extension de deux problèmes d'optimisation combinatoire bien connus : le problème généralisé du voyageur de commerce (GTSP) et le problème du voyageur de commerce asymétrique à contrainte de préséance (PCATSP), dont la version de chemin est connue sous le nom de problème de commande séquentielle (SOP).

Semblable au GTSP classique, le but du PCGTSP est de trouver pour un digraphe d'entrée donné et une partition de son ensemble de nœuds en clusters un itinéraire cyclique (tour) à coût minimum visitant chaque cluster dans un seul nœud. De plus, comme dans le PCATSP, les visites réalisables sont limitées à la visite des clusters dans le respect de l'ordre partiel donné. Contrairement au GTSP et au SOP, à notre connaissance, le PCGTSP reste encore peu étudié tant en termes de théorie polyédrique que d'algorithmes. Dans cette thèse, pour la première fois pour le PCGTSP, nous proposons plusieurs familles d'inégalités valides, établissons la dimension du polytope PCGTS et prouvons des conditions suffisantes garantissant que les inégalités π et σ étendues de Balas induire des facettes. En nous appuyant sur ces résultats théoriques et les approches algorithmiques existantes pour le PCATSP et le SOP, nous introduisons une famille de modèles MILP et plusieurs variantes de l'algorithme branch-and-cut pour le PCGTSP. Nous étudions leurs performances sur les instances de la bibliothèque publique de benchmark PCGTSP LIB, une adaptation connue du SOPLIB classique au problème en question. Les résultats obtenus montrent l'efficacité de l'algorithme. L'article a été publié dans le *European Journal of Operational Research*.

Notre deuxième sujet de recherche est lié à une application industrielle spécifique du PCGTSP - le problème du chemin de coupe discret (CPP). Dans ce problème, nous avons cherché à trouver une trajectoire optimale pour un outil de coupe, afin de minimiser le coût total de traitement, y compris la découpe, le mouvement de l'air, le perçage et autres dépenses, soumis aux contraintes induites par les restrictions de découpe industrielle. Il

est pratique de considérer ces restrictions en termes de contraintes de préséance. Nous introduisons un cadre de solution général pour le RPC qui comprend : (i) l'approche de réduction universelle pour de nombreuses variantes de ce problème au problème du voyageur de commerce généralisé avec contraintes de préséance ; (ii) un support méthodologique pour trouver des solutions (sous-) optimales à ce problème sur la base de l'algorithme de branchement et de coupe et de la méta-heuristique PCGLNS. Les résultats des expériences informatiques montrent l'efficacité du cadre proposé pour résoudre les instances industrielles du problème. L'article a été soumis à l'International Journal of Production Research.

Enfin, nous abordons le problème de routage de véhicules capacitaires (CVRP). CVRP est fortement NP-difficile (même sur le plan euclidien), difficile à approximer dans le cas général et APX-complet pour une métrique arbitraire. Cependant, pour les paramètres géométriques du problème, il existe un certain nombre de schémas d'approximation temporelle quasi-polynomiale et même polynomiale connus. Parmi ces résultats, le célèbre système d'approximation temporelle quasi-polynomiale (QPTAS) proposé par A. Das et C. Mathieu semble être le plus général. Dans cette thèse, nous proposons la première extension de ce schéma à une classe plus large d'espaces métriques. En fait, nous montrons que la métrique CVRP a un QPTAS à tout moment lorsque le problème est installé dans l'espace métrique de toute dimension de doublement $d > 1$ et que la capacité ne dépasse pas $\text{polylog}(n)$. L'article a été publié dans le Journal of Global Optimization.

Mot-clefs : Problème du voyageur de commerce généralisé avec contraintes de précédence; Programmation en nombres entiers; Algorithme Branch-and-Cut; Inégalités valides et facettes ; Structure polyédrique; Problème de chemin de coupe; Problème de coupe du point final; ALNS; Problème de tournées de véhicules avec capacité; un schéma d'approximation en temps quasi-polynomial

Abstract

This thesis focuses on algorithmic design for three combinatorial optimization problems related to transportation, logistics and production research with specific types of industrial constraints. First, we consider the Precedence Constrained Generalized Traveling Salesman Problem (PCGTSP). This problem is an extension of two well-known combinatorial optimization problems — the Generalized Traveling Salesman Problem (GTSP) and the Precedence Constrained Asymmetric Traveling Salesman Problem (PCATSP), whose path version is known as the Sequential Ordering Problem (SOP).

Similarly to the classic GTSP, the goal of the PCGTSP is to find for a given input digraph and partition of its node set into clusters a minimum cost cyclic route (tour) visiting each cluster in a single node. In addition, as in the PCATSP, feasible tours are restricted to visit the clusters with respect to the given partial order. Unlike the GTSP and SOP, to the best of our knowledge, the PCGTSP still remain to be weakly studied both in terms of polyhedral theory and algorithms. In this thesis, for the first time for the PCGTSP, we propose several families of valid inequalities, establish dimension of the PCGTSP polytope and prove sufficient conditions ensuring that the extended Balas' π - and σ -inequalities become facet-inducing. Relying on these theoretical results and existing algorithmic approaches for the PCATSP and SOP, we introduce a family of MILP-models and several variants of the branch-and-cut algorithm for the PCGTSP. We study their performance on the instances of the public benchmark library PCGTSP LIB, a known adaptation of the classic SOPLIB to the problem in question. The obtained results show the efficiency of the algorithm. The paper was published in European Journal of Operational Research.

Our second research topic is related to a specific industrial application of the PCGTSP - the discrete Cutting Path Problem (CPP). In this problem, we aimed to find an optimal path for a cutting tool, in order to minimize the total processing cost including cutting, air-motion, piercing, and other expenses, subject to constraints induced by industrial cutting restrictions. It is convenient to consider such restrictions in terms of precedence constraints. We introduce a general solution framework for CPP that includes: (i) the universal reduction approach for numerous variants of this problem to the the

Precedence Constrained Generalized Traveling Salesman Problem; (ii) methodological support for finding (sub-) optimal solutions of this problem on the basis of branch-and-cut algorithm and PCGLNS meta-heuristic. The results of computational experiments show the efficiency of the proposed framework for solving industrial instances of the problem. The paper was submitted to International Journal of Production Research.

Finally, we tackle the Capacitated Vehicle Routing Problem (CVRP). CVRP is strongly NP-hard (even on the Euclidean plane), hard to approximate in general case and APX-complete for an arbitrary metric. However, for the geometric settings of the problem, there is a number of known quasi-polynomial and even polynomial time approximation schemes. Among these results, the well-known Quasi-Polynomial Time Approximation Scheme (QPTAS) proposed by A. Das and C. Mathieu appears to be the most general. In this thesis, we propose the first extension of this scheme to a more wide class of metric spaces. Actually, we show that the metric CVRP has a QPTAS any time when the problem is set up in the metric space of any fixed doubling dimension $d > 1$ and the capacity does not exceed $\text{polylog}(n)$. The paper was published in Journal of Global Optimization.

Keywords: Precedence Constrained Generalized Traveling Salesman Problem; Integer programming; Branch-and-cut algorithm; Facet-inducing inequalities; Polyhedral structure; Cutting Path Problem; Endpoint Cutting Problem; Adaptive Large Neighborhood Search; Capacitated Vehicle Routing Problem; Quasi-Polynomial Time Approximation Scheme

Acknowledgements

When I've started work on my PhD thesis three years ago, I was surrounded by uncertainty. Like a pathfinder, who is about to discover a new continent, I had to step into the great unknown, and hope for the best. And now, after all these years, I have no regrets. Sometimes, the most hard part is to make this one step.

My accomplishment of this degree was made into reality thanks to the help of many others. First of all, I would like to express my gratitude to my supervisors, Olga Battaia and Ruslan Sadykov, who motivated me and gave me an opportunity to become a better researcher. I will always appreciate their kindness, hospitality, experience and trust. In addition, I would like to thank Boris Detienne, who has agreed to help me to finish my PhD at the very last stage. I wish to extend my appreciation to the director of my research unit, Francois Clautiaux, and his colleagues at the EDGE team, whom I've enjoyed working with during the research seminars. Speaking of seminars, I'm eternally grateful to Pierre Pesneau. One of the chapters of my PhD was significantly extended thanks to his dedication and knowledge of algorithmic design and polyhedral theory. I also want to thank researchers who has agreed to be a part of the jury and additional thanks to my reviewers, Frédéric Semet and Pierre Fouilhoux, whose invaluable comments have helped me tremendously to improve my thesis.

My greatest thanks go to my parents, for their endless support, being my sanctuary in the great unknown. Thanks to them, I am who I am today and no words can express the sufficient gratitude. I just hope to do my best to make them proud. In addition, I would like to thank my friends, who were with me during the whole journey. When I think about my home, I cannot imagine it without my first research workplace - Krasovskii Institute of Mathematics and Mechanics. I'm eternally grateful to all the researchers and professors for their advice and suggestions, and for giving me access to the Supercomputer Uran, where I've performed the majority of my computational experiments.

Finally, I wish to thank everyone at University of Bordeaux and Kedge Business School - the professors, faculty and my fellow PhD associates - for their warm welcome and tireless support. In particular, I would like to thank Yann Bouchery and Amir Pirayesh, who allowed me to teach their courses and gain a real experience, as a lecturer.

Table of contents

List of figures	xi
List of tables	xv
1 Introduction	1
1.1 Optimization for the global economy	1
1.2 Considered problems	4
1.3 Related work	6
1.3.1 Precedence Constrained Generalized Traveling Salesman Problem	6
1.3.2 Cutting Path Problem	10
1.3.3 Capacitated Vehicle Routing Problem	14
1.4 Research motivation	18
1.5 Contributions	21
1.5.1 Precedence Constrained Generalized Traveling Salesman Problem	21
1.5.2 Discrete Cutting Path Problems	23
1.5.3 Capacitated Vehicle Routing Problem	24
2 Precedence Constrained Generalized Traveling Salesman Problem: Poly-	
hedral Study, Formulations, and Branch-and-Cut Algorithm	27
2.1 Problem statement	27
2.1.1 Preliminaries	28
2.1.2 Compact MILP model	29
2.2 Valid inequalities	31
2.2.1 Predecessor and successor inequalities	31
2.2.2 Precedence cycle breaking inequalities	32
2.2.3 Single-option inequalities	33
2.2.4 Strengthened precedence variables and network flow based inequalities	34
2.2.5 Separation procedures	36

2.3	Facets of the PCGTS polytope	37
2.3.1	Dimension of the PCGTS polytope	38
2.3.2	Facet-inducing inequalities	47
2.4	Formulations	52
2.5	Branch-and-Cut Algorithm	55
2.5.1	PCGLNS Primal Heuristic	55
2.5.2	Implementation	56
2.6	Numerical evaluation	56
2.6.1	PCGTSP LIB Benchmark library	57
2.6.2	Experiment I: Comparison of the LP-relaxations	58
2.6.3	Experiment II: Comparison of Branch-and-Cut Algorithms	60
3	Discrete Cutting Path Problems: General Solution Framework with Accuracy Guarantees	67
3.1	General modelling approach: from the discrete CPP to the PCGTSP	67
3.1.1	Discrete CPP without cutting preemption	68
3.1.2	ECP: several sources of industrial constraints	72
3.2	Generic solution framework for the discrete CPP	81
3.2.1	Branch-and-cut algorithm	82
3.2.2	An illustrative example	88
3.3	Computational Experiment	92
3.3.1	Experimental setup	92
3.3.2	Results	92
4	CVRP: Efficient approximation in metric spaces of a fixed doubling dimension	97
4.1	Problem statement	97
4.2	Metric spaces of a fixed doubling dimension	99
4.3	Extended Das and Mathieu approximation scheme	102
4.3.1	Accuracy driven rounding	104
4.3.2	Randomized hierarchical clustering	109
4.3.3	Demand ranking and relaxed solutions	114
4.3.4	Baseline Dynamic Programming	126
4.3.5	Modification based on internal DP	129
4.4	Complexity bounds	140

Conclusion and Perspectives	143
4.5 Summary	143
4.6 Open questions and Perspectives	145
Bibliography	147

List of figures

- 1.1 Examples of known industrial cutting features: inner/outer contours, common cuts, islands, and bridges. Red dot marks the current position of a cutting tool. 12
- 2.1 Example of a directed graph and its bipartite representation 40
- 2.2 Matrix of extreme points of $\mathcal{H}(\alpha, \beta, \gamma)$ 40
- 2.3 Block M_{22} and the incidence matrix of B_v^* 40
- 2.4 Cut $\delta(v)$ for Case 1 42
- 2.5 Bipartite graph \bar{B}_v^* for Case 1 42
- 2.6 Cut $\delta(v)$ for Case 2 42
- 2.7 Bipartite graph \bar{B}_v^* for Case 2 42
- 2.8 Cut $\delta(v)$ for Case 3 43
- 2.9 Bipartite graph \bar{B}_v^* for Case 3 43
- 2.10 Representation of $\delta(v)$ and the connected components of \bar{B}_v^* for Case 4. Bold line provides connectivity of the \bar{B}_v^* 44
- 2.11 Representation of $\delta(v)$ and the connected components of the \bar{B}_v^* for Case 5 44
- 2.12 Structure of matrix A 45
- 2.13 Connected components of B_v^* for Case 3. $C(v) \in \bar{\mathfrak{C}}$ (left) and $C(v) \in \bar{\mathfrak{C}}$ (right) 49
- 2.14 Connected components of B_v^* for Case 4. $C(v) \in \bar{\mathfrak{C}}'$ (left) and $C(v) \in \bar{\mathfrak{C}}$ (right) 49
- 2.15 Example of the contradicting solution 52
- 3.1 An example of the discrete CPP without cutting preemption 69

-
- 3.2 Example of the reduction: initial instance with a feasible cutting path(left) and its auxiliary PCGTSP instance (right). In this example, we have the only precedence constraint induced by the embedding of the contour K_2 into K_3 . Therefore, K_2 should be cut out before K_3 , and its PCGTSP counterpart C_3 should be visited by an arbitrary feasible route before C_4 . Thus, the DAG H has the only arc (C_3, C_4) 71
- 3.3 Encoding of a cutting element 72
- 3.4 Contour gadget: twin nodes, arc doubling and induced precedence constraints. In this case, the initial contour consists of elements I and II. To this end, the corresponding contour gadget is represented by two-node clusters C_I and C_{II} and their successor – artificial twin cluster T . Any arc going from two-node clusters is a bold line, while its corresponding artificial twin arcs are labeled by a dash line 73
- 3.5 Instance of the ECP with inner/outer contours, a feasible cutting path is colored orange. In this example, we have two contours. The first one consists of the elements I, II, and III, while the second one of the elements IV and V. Therefore, the corresponding graph G contains two contour gadgets, specified by clusters $C_I, C_{II}, C_{III}, T_1$, and C_{IV}, C_V, T_2 respectively. Since, the second contour is embedded into the first one, cluster T_2 precedes T_1 , which is encoded by the arc (T_2, T_1) in the graph H . Feasible cutting path and the appropriate same cost PCGTSP solution are presented by orange 75
- 3.6 Common cuts can induce several shapes. In this example, part P_2 has two cuts, J_1 and J_2 common with parts P_1 and P_3 , respectively. These common cuts induce the shapes S_1, S_2 , and S_3 , such that J_1 intersects interiors of S_1 and S_3 , while J_2 — interiors of S_2 and S_3 . Therefore, in graph G , each node of the cluster J_1 (J_2) has a twin in the clusters S_1 and S_3 (S_2 and S_3) inheriting all the outgoing arcs. Each twins of the same node are induced cliques. Furthermore, precedence of the cluster J_1 with respect to the clusters S_1 and S_3 (J_2 with respect to S_2 and S_3) is encoded by corresponding arcs of the graph H 76

3.7	An example of the ECP instance with common cuts, and the corresponding auxiliary instance of the PCGTS. Regular cutting elements are denoted by R_1, \dots, R_4 , common cuts – by J_1, J_2, J_3 , and the induced shapes – by S_1, \dots, S_6 . All of them is represented by the same-name cluster in the graph G . Observe that the cluster J_1 precedes clusters S_1, S_4 , and S_5 , since the respective common cut intersects interior of these shapes, which is encoded by the arcs $(J_1, S_1), (J_1, S_4), (J_1, S_5)$ in the graph H . The rest of the precedence constraints is obtained in the similar way. A feasible cutting path and the appropriate same cost PCGTSP route is colored orange	77
3.8	Instance of the ECP with an island	78
3.9	Crossroads, cutting elements and bridges	78
3.10	Reduction of the ECP with bridges to the equivalent instance of the PCGTSP: P_i are crossroads, W_j are gateways, and C_q are remaining clusters representing contour elements. Observe that in this case, the nature of precedence constraints is specified by the fact that each crossroad cluster precedes all the gateways assigned to it. For instance, cluster P_1 precedes the clusters W_1 and W_2 , which is encoded by the arcs (P_1, W_1) and (P_1, W_2) in the graph H . A feasible cutting path and the same cost appropriate feasible solution of the PCGTSP are presented by orange	79
3.11	Example of the ECP with bridges	89
3.12	Specific arcs of the graph G	90
3.13	The obtained optimal PCGTSP solution (a) and the appropriate optimal cutting path of the same cost (b)	91
4.1	Proof of Lemma 6	107
4.2	An appropriate net-respecting route, Portals of two levels are denoted by \square and \circ	112
4.3	To obtain an upper bound for the weight of $\text{MST}(X_{red}^l \cup \{y\})$, we combine the spanning trees for the red points C_{red} (bold solid lines) and the portals (dashed) in any cluster at level l with the minimum spanning tree in the auxiliary graph of clusters H	121

List of tables

- 2.1 Options for cluster $C(v)$ 48
- 2.2 PCGTSP LIB library. 57
- 2.3 Comparison of formulations M_1 – M_5 and M_1^* 58
- 2.4 Performance of the combined and sampled formulations. 59
- 2.5 Comparison of the branch-and-cut algorithms (a) 61
- 2.6 Comparison of the branch-and-cut algorithms (b) 62
- 2.7 Comparison of the branch-and-cut algorithms (c) 63
- 2.8 Comparison of the branch-and-cut algorithms (d) 65

- 3.1 PCGLNS performance results 94
- 3.2 PCGLNS gap percentiles for test instances 95

- 4.1 An example of a resource row A_i and related sequence of the concatenation profiles for $\mathcal{K} = 3$, $m_i = 5$, and $\bar{S} = \{\sigma_0, \dots, \sigma_3\}$ 133

Chapter 1

Introduction

In this preliminary chapter, we illustrate how the optimization methods and the operational research can help with the challenges and opportunities of the modern world. To explain the motivation of this research, we provide literature review for each topic considered in the thesis. We conclude the chapter by describing the main contribution the thesis and show the road-map of its structure.

1.1 Optimization for the global economy

The concept of the globalization comes into a reality. Nowadays, it is impossible to imagine a modern world without an international trade [Xu et al., 2020, Hong et al., 2023]. Being a remarkable achievement, global economy presents many challenges and opportunities [Zhang, 2018, Kurniawan et al., 2023].

The first challenge deals with the climate changes and global warming [Peng et al., 2020, Cho et al., 2023]. Due to the rising CO₂ emissions, the industry is trying to find a replacement of fossil fuels by employing a different, more sustainable sources of power [Güney, 2022]. The following study reviews the global impact of solar power plants, using different solar collectors and makes a comprehensive comparison [Gobio-Thomas

[et al., 2023](#)]. In addition, the authors indicate the need of novel algorithms and software to operate those sophisticated machines. Another study highlights the importance of optimization in an attempt to replace some of the fossil fuel with an advanced hydrogen production [[Riera et al., 2023](#)]. Authors mention the potential applications of hydrogen in transmission, distribution and integration with different energy systems, however the necessary infrastructure is insufficient. Furthermore, the paper indicates the previously proposed optimization models, such as Mixed Integer Linear Programming (MILP) in order to improve the production process. The authors conclude their review by mentioning the need of those models in practice. Finally, due to the increased temperatures and ever-rising transportation cost, there is a major risk of disruption of the existing supply chains [[Clavijo-Buritica et al., 2023](#)]. This is especially important for the agricultural production and the question of resilience of the supply chains is considered in this study. The authors propose a framework to analyze and mitigate the potential disruptions of one specific food supply chain, using the methods of optimization. Once again, authors highlight the necessity of novel algorithms.

The second challenge is one of the most recent ones concerned with COVID-19 pandemic [[Adedoyin and Soykan, 2023](#)]. As it is shown in the paper [[Montoya-Torres et al., 2023](#)], pre-pandemic supply chains were not ready for the outbreak. Even after the pandemic was resolved, many companies still suffer the consequences. However, authors provide a method, that involves the operations research techniques in order to compensate the crisis. Speaking of the repercussions of the pandemic, we should mention the variety of disruptions that occur during COVID-19, such as food, medicine and eventually the lack of professional assistance at the right time, in the right quantity, of the right quality [[Shen and Sun, 2023](#), [Chervenkova and Ivanov, 2023](#)]. The authors point out the fact that

conventional methods of the disruption mitigation are no longer viable for a global crisis and propose a novel frameworks in order to stabilize the situation with the assistance of optimization methods.

The third challenge consists of geopolitical issues [Iqbal et al., 2019, Zhang et al., 2023]. A recent work studies changes induced by US-China trade conflict after 2017 [Yuan et al., 2023]. Authors consider possible supply chain reallocations and their complete rebuilding process in the new countries, thus causing a potential disruptions and cost increase. They also propose an approach to minimize the emissions that can occur during the relocation. Another study around this trade conflict propose an optimization model that can reduce the potential costs involved with the new tariffs and development of novel networks [Huang et al., 2023]. Finally, the decision-making aspect is tackled for the remanufacturing of global supply chains [Li et al., 2023b]. In particular, the authors propose practical schemes for decision makers that involve optimization methods in order to resolve the disruptions.

Speaking of opportunities of the global economy, we should definitely mention novel technological advances, such as Artificial Intelligence, and their integration into production process [Kar et al., 2022]. As an approach to deal with aforementioned challenges, Industry 5.0 has attracted many researchers around the globe [Leng et al., 2022]. Its applications include an Internet of Things for the medical devices industry [Nayeri et al., 2023], Deep Learning approach for airport detection to improve the aircraft routes [Li et al., 2023a], and additional resilience towards the industry against the unexpected disruptions, such as geopolitical conflicts or COVID-19 pandemic [Leng et al., 2023]. However, all those results would be impossible without the optimization methods.

1.2 Considered problems

This doctoral thesis is devoted to several deterministic extensions of two classic combinatorial optimization problems — Traveling Salesman Problem (TSP) [Lawler et al., 1985] and Vehicle Routing Problem (VRP) [Toth and Vigo, 2014], enhanced with additional *constraints*.

The TSP was initially introduced in 1932 in [Menger, 1932]. Its statement may be defined as follows. Let $G = (V, E)$ be a graph where V is a set of n nodes. E is a set of edges (or arcs, if G is a directed graph), and let $C = (C_{ij})$ be a transportation cost matrix associated with E . The problem consists of determining a minimum cost Hamiltonian cycle (one that visits each node $v \in V$ exactly once). The algorithmic analysis of the TSP is dated back to the seminal paper [Dantzig et al., 1954]. The TSP is often used as a proving ground for numerous optimization techniques [see ex. Gutin and Punnen, 2007]. In addition, its popularity is explained by many real-life applications, including urban distribution [Macrina et al., 2020], drone trajectory optimization [Otto et al., 2018], logistics [Yu et al., 2021], machine scheduling problems [Mosayebi et al., 2021], and DNA sequencing [Caserta and Voß, 2014].

As for the VRP, its statement can be defined as follows [Laporte, 2009]. Let $G = (V, E)$ be a graph whose node set $V = X \cup \{y\}$, where X is a set of customers, each of them having a unit demand on some homogeneous commodity, and y is a depot. All the customer's demand should be serviced by identical vehicles located initially at the depot y . To each edge $\{i, j\}$, we assign its transportation cost c_{ij} . The problem is to construct a minimum cost family of cyclic routes servicing the total customer demand, each of them departs from and arrives at the depot y .

There are known many versions of the VRP describing various practical applications of this problem. Among them are Multi-Objective VRP for

creation of alternative routes [Rossit et al., 2019], Generalized VRP and TSP with profits [Baldacci et al., 2010], truck-drone routing problem [Liang and Luo, 2022], stochastic dynamic routing problem [Soeffker et al., 2022], sustainable urban vehicle routing [Dündar et al., 2021], two-echelon routing problems [Cuda et al., 2015], and so on.

In this thesis, our research is focused on the following problems:

- the Precedence Constrained Generalized Traveling Salesman Problem (PCGTSP), which extends both Asymmetric TSP and Generalized TSP;
- the discrete Cutting Path Problem (CPP);
- the Capacitated Vehicle Routing Problem (CVRP) in metric spaces of an arbitrary fixed doubling dimension.

The Generalized Traveling Salesman Problem (GTSP) [see e.g. Pop et al., 2023] is a modification of the classic TSP, whose instance is still given by a transportation network specified by a edge-weighted directed graph $G = (V, A)$. Unlike the TSP, in GTSP, the set of nodes V is partitioned into mutually disjoint subsets called *clusters*. The goal is to construct a closed tour that visits each cluster in one node exactly and minimizes the accumulated transportation costs. Considered in this thesis the PCGTSP is a generalization of GTSP, which is enhanced with additional *precedence constraints* specifying the traversing order of the clusters [see e.g. Salman et al., 2020].

In turn, the Capacitated Vehicle Routing Problem [Toth and Vigo, 2001] considered in this thesis is a generalization of the classic VRP, where each feasible route has a given *capacity*.

Unlike the first two problems, the discrete CPP [Dewil et al., 2016] comes from the industry and generally speaking, is not a combinatorial optimization problem itself. The problem is related to sheet cutting

processes, whose instance is given by a cutting plan specifying location of flat parts to be cut out from a plate of material (e.g. metal or plastic). It is required to construct a path of the cutting tool, which satisfies given industrial requirements and minimizes the processing cost. We describe those requirements in more detail in Subsection 1.3.2. In this thesis, we show that the discrete CPP benefits from the algorithmic results obtained for the PCGTSP, and can be considered as a practical application of this problem.

1.3 Related work

1.3.1 Precedence Constrained Generalized Traveling Salesman Problem

Introduced in the seminal paper by [Srivastava et al., 1969], the Generalized Traveling Salesman Problem is one of the most well-known generalizations [Gutin and Punnen, 2007, Pop et al., 2023] of the classic Traveling Salesman Problem. It has numerous industrial applications including air time minimization in metal sheet cutting [Dewil et al., 2016, Chentsov et al., 2018b, Makarovskikh et al., 2018] and coordinate measuring machinery [Salman et al., 2016].

Being an extension of the classic TSP, the GTSP is strongly NP-hard even on the Euclidean plane [Papadimitriou, 1977] any time when number of clusters m is a part of the input. On the other hand, an adaptation to this problem of the well-known Held and Karp dynamic programming scheme [Held and Karp, 1962] has running-time bound $O(n^3 m^2 \cdot 2^m)$ where n is the number of nodes, and m is the number of clusters, i.e. the GTSP belongs to the class of Fixed-Parameter Tractable (FPT) problems, being parameterized by the number of clusters. Furthermore, it can be solved to optimality in polynomial time, provided that $m = O(\log n)$.

In the case of PCGTSP, the aforementioned running time bound can be further improved using notion of *ideal* of partially ordered set and *width* of such an order [see e.g. [Davey and Priestley, 2002](#)]. A subset Q of a partially ordered set P is called its ideal if, whenever $x \in Q$, $y \in P$ and $y \leq x$, we have $y \in Q$. In turn, width of an order defined on the set P is equal to a size of the largest its *antichain* (a subset L of P , where any two distinct elements of L are incomparable). As it is shown in [[Khachay et al., 2021b](#)], the running time of the PCGTSP is $O(n^3 m^2 \cdot |\mathcal{J}|)$, where \mathcal{J} is a family of ideals of the given partially ordered set of clusters. In particular, if the order specifying the precedence constraints is of fixed width w , then $|\mathcal{J}| = O(m^w)$ [[Steiner, 1990](#)]. Thus, in this case, the PCGTSP can be solved to optimality in polynomial time.

The algorithmic design for the GTSP has been developed in the literature in several directions. The first approach is based on the reduction of the initial problem to an appropriate instance of the Asymmetric TSP (ATSP) [[Noon and Bean, 1993](#), [Laporte and Semet, 1999](#)], which at first glance gives an opportunity to employ a vast variety of known algorithms designed for the ATSP [see e.g. [Roberti and Toth, 2012](#)]. However, as it is stated [[Karapetyan and Gutin, 2012](#)], despite its mathematical elegance, this approach suffers from several technical shortcomings. First, even a close-to-optimal solutions of such auxiliary ATSP instances may correspond to infeasible solutions of the initial GTSP. Furthermore, such instances may have a quite unusual shape and thus difficult to solve for the existing TSP solvers.

Another approach provides various heuristics and meta-heuristics. Among them are the memetic algorithms [[Gutin and Karapetyan, 2010](#)], an extension of the Lin-Kernighan-Helsgaun heuristic solver [[Helsgaun, 2015](#)], and the GLNS meta-heuristic [[Smith and Imeson, 2017](#)] based on the Adaptive

Large Neighborhood Search (ALNS) framework, which appears to be the most efficient at the moment.

Finally, we should mention two research directions. First of them is related to design of exact algorithms, including problem-specific branch-and-bound and branch-and-cut algorithms [Fischetti et al., 1997, Yuan et al., 2020]. And the second one tackles approximation algorithms with theoretical performance guarantees [see e.g. Khachai and Neznakhina, 2017]).

The Sequential Ordering Problem (SOP), which is also known as Precedence Constrained Asymmetric TSP (PCATSP), was introduced in [Escudero, 1988]. We should mention three groups of important results obtained for the both problems on which the current research for the PCGTSP is based on.

The first of them, in the field of polyhedral study of the PCATSP, was obtained in the seminal paper [Balas et al., 1995], where sufficient conditions for the π - and σ -inequalities to be facet-inducing were proved.

The second group comprises valid inequalities that exploit precedence constraints explicitly and approaches to their strengthening, as well as the design of MILP-models (formulations) in order to obtain better lower bounds while decreasing time complexity of the appropriate LP-relaxations. Among them are compact formulations proposed in [Sarin et al., 2005] as an extension of results of [Gouveia and Pires, 1999, 2001, Sherali and Driscoll, 2002], and formulations whose exponential-size sets of valid inequalities are supplemented with polynomial-time separation techniques [Gouveia and Pesneau, 2006]. To the best of our knowledge, to the date, the models providing the tightest lower bounds were introduced in [Gouveia et al., 2018].

The last group of results relies on design and implementation of problem-specific branch-and-cut algorithms including ones proposed in [Ascheuer

et al., 2000]), [Cire and van Hoeve, 2013] and [Gouveia and Ruthmair, 2015], where the last one is regarded to be state-of-the-art on the topic.

The PCGTSP extends the PCATSP as follows. Any instance of PCATSP is considered to be the instance of PCGTSP, where all clusters are singletons. Unlike both GTSP and PCATSP, the PCGTSP still remains weakly studied. To the best of our knowledge, all the related published results are limited to:

(i) efficient algorithms for several specific precedence constraints including partial orders of Balas-type [see e.g. Balas and Simonetti, 2001, Chentsov et al., 2016] and the orders that lead to quasi- and pseudo-pyramidal optimal tours [Khachay and Neznakhina, 2020];

(ii) the PCGLNS heuristic solver proposed in [Khachay et al., 2020a] that extends the results obtained in [Smith and Imeson, 2017] to the case of PCGTSP;

(iii) branch-and-bound and DP-and-bound algorithms for this problem [Khachay et al., 2021b], based on Balas instance preprocessing [Balas et al., 1995], Held and Karp branching framework [see e.g. Morin and Marsten, 1976], and the combinatorial lower bounds from [Salman et al., 2020],

(iv) the public PCGTSP LIB benchmark library proposed in (Salman et al. [2020]) as an extension of the well-known SOPLIB library. According to the literature [Salman et al., 2020, Khachay et al., 2021b], 12 out of 40 instances of this library were solved to optimality. Meanwhile, their solutions can be found within a competitive time by Gurobi solver supplied with our extension of the L1PCATSPxy compact model, previously introduced in [Sarin et al., 2005] for the PCATSP, built-in cutting planes, and PCGLNS primal heuristic.

In addition, we should mention the branch-and-cut algorithm proposed recently in [Yuan et al., 2020] for the GTSP with time windows. This result seems to be relevant as the time windows defined on clusters induce

natural precedence constraints. Unfortunately, this approach is hardly applicable to the general PCGTSP, since a partial order defined on the set of clusters not necessarily admits encoding in terms of time windows.

1.3.2 Cutting Path Problem

Speaking of industrial applications of the PCGTSP, we focus on air time minimization in metal sheet cutting [Dewil et al., 2016, Chentsov et al., 2018b, Makarovskikh et al., 2018] and coordinate measuring machinery [Salman et al., 2016]. As it is shown in Chapter 3, the nature of cutting features allow us to propose an efficient general solution framework for the CPP relying on our recent results obtained for the PCGTSP.

Sheet cutting processes are widely employed by various industries including aerospace, automotive, or garment. These processes usually employ a cutting technology, e.g. laser, gas plasma torch, or diamond cutter to obtain flat parts from a rectangular plate of metal, glass, fabric, etc.

Typically, design of a cutting procedure pursues two different optimization criteria motivated by cutting waste reduction and decreasing of the process time or cost, respectively. Even if some rare works [Sherif et al., 2014, Oliveira et al., 2020] develop algorithms to optimize both these objectives simultaneously, the dominating research direction is based on the decomposition of the initial problem into two closely related subproblems, known as *the Nesting Problem (NP)* [Herrmann and Delalio, 2001] and *the Cutting Path Problem (CPP)* [see, e.g. Lee and Kwon, 2006, Silva et al., 2019]. The former can be reduced to a two-dimensional packing problem [Gilmore and Gomory, 1961, Dyckhoff, 1990, Wäscher et al., 2007] with the objective to minimize the nesting costs [Bennell and Oliveira, 2008], Cherri et al. [2019]. The latter problem aims to construct an optimal path for the cutting tool while minimizing the *total processing cost* that includes

cutting, air-motion, piercing, and other expenses, subject to constraints induced by industrial cutting restrictions [Dewil et al., 2016].

For the discrete CPP, all the cutting operations should be started only within a given finite set of points of each contour. In particular, there are two discrete settings: (i) each contour has a dedicated single starting point, (ii) each contour may have several such points. In this thesis, we consider these discrete versions of the CPP for which we describe a number of frequently encountered in practice cutting features that have an important impact on the problem modeling and the efficiency of the obtained solutions.

One of the most common type of such constraints describes the case where *one contour is enclosed into another one*. In this case, the outer contour should not be cut out completely before the inner one. The similar constraint occurs if a waste area is encompassed by several parts to be cut. This area, called *an island*, should be completely cut out before the last of the neighboring parts.

Another common industrial feature is the existence of so called *common cuts* belonging to several contours simultaneously because of adjacency of two parts.

Finally, in the modeling of CPP, the cases of relatively *thin* and *thick* material plates need to be treated differently. In the former case, each cut can be made *on the fly*, while in the latter one, special *piercing* procedures may be needed inducing significant additional costs. To avoid these costs, sometimes it may be convenient to combine several contours by *bridges*, which make it possible to traverse between these contours without switching off the cutting tool. If the model for the corresponding CPP is appropriately adjusted, it helps also to decide between piercing expenses and bridge-cutting costs (see Fig.1.1).

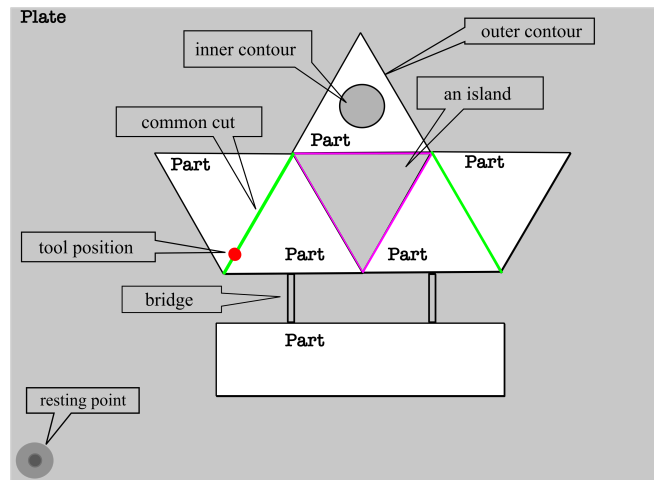


Figure 1.1: Examples of known industrial cutting features: inner/outer contours, common cuts, islands, and bridges. Red dot marks the current position of a cutting tool.

In the literature, several variants of the CPP are considered. The classification of these variants introduced in the seminal paper by [Hoeft and Palekar \[1997\]](#) was notably extended in [[Dewil et al., 2016](#), [Chentsov et al., 2018a](#), [Petunin, 2019](#), [Petunin et al., 2022](#)]. According to this classification, the CPP family is dichotomized according to the rule of choosing the entry cutting point for each contour. Within the first sub-family called *continuous*, a cutting procedure can be started / interrupted / resumed at any point of a contour. This group includes two main problems: the Continuous Cutting Problem (CCP), where each contour must be cut out entirely before moving to the next one, and the Intermittent Cutting Problem (ICP) which allows cutting preemption.

Early work on the discrete versions of the CPP is due to [Han and Na \[1999\]](#) who developed a simulated annealing-based algorithm for torch path optimization for laser cutting. [Castelino et al. \[2003\]](#) provided a Generalized Traveling Salesman Problem (GTSP) formulation for the discrete CPP referred to as machining tool path problem. To solve the obtained problem, the authors exploited an extended Noon and Bean transformation [[Noon and Bean, 1993](#)] to the Asymmetric Traveling Salesman Problem (ATSP),

which was solved subsequently with Lin-Kernighan-Helsgaun heuristic [Helsgaun, 2000], and branch-and-cut algorithm proposed in [Ascheuer et al., 2000]. However, as it shown in [Karapetyan and Gutin, 2012], this approach can hardly be applied for the practical solution of the CPP.

In [Chentsov, 2014], the discrete CPP was modeled in terms of sequential traversing of a given set of megalopolices (clusters) complemented with precedence constraints and internal jobs. The authors proposed an original scheme of Dynamic Programming (DP) to solve the problem to optimality even in the case of uncertain air-time motion and cutting costs. Although the obtained problem is known to be NP-hard, efficient optimal and approximation algorithms were proposed for several known classes of the precedence constraints [see, e.g. Chentsov et al., 2016, Khachay and Neznakhina, 2020]. The approach was further developed for CNC plate cutting machines in [Chentsov et al., 2018a].

The Endpoint Cutting Problem (ECP) which is a discrete cutting path problem that allows preemption for cutting tool has been introduced by Manber and Israni [1984]. In the literature, this problem was often reduced to the well-known Rural Postman Problem (RPP) and several approximate approaches were proposed to solve it: heuristics exploiting the concept of dynamic graphs [Moreira et al., 2007], a heuristic method taking into account the density of a super-hard material [Imahori et al., 2008], memetic algorithms [Rodrigues and Ferreira, 2012].

Other mathematical models and solution methods have been also exploited for ECP. In [Dewil et al., 2015], several special formulations of ECP were reduced to the GTSP consequently solved by known heuristics. Hu et al. [2022] introduced an efficient Delaunay triangulation-based bridging algorithm minimizing the total bridge length. Makarovskikh et al. [2018] developed novel approximation algorithms for ECP and ICP, relying on their recent theoretical results for planar graphs. Finally, Cuellar-Usaquén

et al. [2023] proposed and evaluated a novel GRASP-based algorithm for ECP.

The analysis of the aforementioned literature reveals several important research gaps:

- (i) although there exist attempts to reduce special variants of the CPP to classic combinatorial optimization problem including TSP, GTSP, or RPP, none of the approaches from the literature is capable to model the tool routing problem with all the diversity of cutting features, which makes difficult their use in industrial applications;
- (ii) numerical analysis of the corresponding algorithms appears to be developed rather weakly. Among the related work, only papers [Dewil et al., 2014, 2015, Moreira et al., 2007, Rodrigues and Ferreira, 2012, Cuellar-Usaquén et al., 2023] report some experimental results. However, the experiments carried out in these papers still remain irreproducible. The reported results cannot be taken for the comparison due to either the lack of implementation details, insufficiently clear experimental setup, or inaccessibility of benchmark instances.

On the other hand, the PCGTSP has a variety of well-developed algorithms and solution techniques. The research effort on this problem stems from seminal papers of Balas et al. [1995], Balas and Simonetti [2001]. At this moment, the branch-and-cut algorithm, proposed in [Khachai et al., 2023], and based on strong MILP formulations, extensive polyhedral results, and promising PCGLNS primal heuristic from [Khachay et al., 2020a], is the state-of-the-art for this problem.

1.3.3 Capacitated Vehicle Routing Problem

In order to tackle the third research direction related to the approximate methods, we propose an approximation algorithm with accuracy guaran-

tees for the problem, which is related to the TSP – Capacitated Vehicle Routing Problem (CVRP). Being an extension of the well-known Vehicle Routing Problem, it is actively studied combinatorial problem with numerous important applications in Operations Research ([Demir et al. \[2019\]](#), [Laporte \[2009\]](#), [Toth and Vigo \[2014\]](#)).

To the best of our knowledge, the problem was introduced by G. Dantzig and J. Ramser in their seminal paper [[Dantzig and Ramser, 1959](#)], which provided the first mathematical model of gasoline distribution over the network of gas stations.

Since then, the field of the algorithmic design for the CVRP is developed in several research directions. The first direction is based on a proposition of the appropriate mixed integer program for the problem in question and finding an optimal solution of this program using some of the well-known branch-and-price methods [[Demir et al., 2019](#)]. Recently, a significant success was achieved both in development such algorithms and computational hardware [[Hokama et al., 2016](#), [Pessoa et al., 2018](#)]. It is worth to mention one of the most recent results by [[Pessoa et al., 2020](#)], where the generic solver for many VRP settings, including CVRP was proposed. In particular, authors managed to solve six more instances of the CVRPLIB benchmarking library to optimality. However, an instance with less than 9000 nodes took more than 15 days of computation time. Unfortunately, with the real-life applications such time limits are unacceptable.

A wide range of modern heuristic algorithms and meta-heuristics makes up the basis of the second research direction. To date, the most significant numerical results were obtained for local search algorithms [[Arnold and Sörensen, 2019](#), [Avdoshin and Beresneva, 2019](#)], Tabu search [[Qiu et al., 2018](#)], Variable Neighborhood Search (VNS) [[Frifita and Masmoudi, 2020](#), [Polat, 2017](#)], machine learning [[Nazari et al., 2018](#)], evolutionary [[Vidal et al., 2013b](#)], and bio-inspired algorithms [[Necula et al., 2017](#), [Su-Ping](#)

and Wei-Wei, 2016], as well as their various combinations [Chen et al., 2019, Nalepa and Blocho, 2016]. Often heuristic algorithms demonstrate remarkable performance, yielding close to optimal or even optimal solutions for CVRP instances of extremely large size. Nevertheless, an absence of any theoretical guarantees implies additional computational expenses related to numerical performance evaluation during the transition to any novel class of instances. Furthermore, in the case when the results of such an evaluation appear to be inadmissible, we have to perform an additional tuning of outer parameters of the considered heuristic algorithm [see e.g. Williamson and Shmoys, 2011].

The arguments above confirm the relevance of the third direction related to the design of approximation algorithms with theoretical performance guarantees. It is known that CVRP is NP-hard in the strong sense, enclosing the classic TSP, and remains intractable even on the Euclidean plane [Papadimitriou, 1977]. The problem is hard to approximate in general case (provided $P \neq NP$), APX-complete for an arbitrary metric [Asano et al., 1997, Haimovich and Rinnooy Kan, 1985] even for an arbitrary fixed capacity $q \geq 3$.

Before we proceed with the review of recent results on approximation algorithms, we recall some necessary notation. Let $C: \mathbb{N} \rightarrow \mathbb{R}_+$. We call \mathcal{A} a $C(n)$ *approximation algorithm*, if for an arbitrary instance I of the combinatorial optimization problem \mathcal{J} of length n , the weight of a feasible solution $W_I(\mathcal{A}(I))$ is different from the optimal value OPT at most $C(n)$ times. In turn, we call $C(n)$ an approximation ratio or *accuracy guarantee*.

A *time-complexity* (or running time) of an algorithm \mathcal{A} is the function $T: \mathbb{N} \rightarrow \mathbb{R}_+$, estimated by an upper bound of number of operations necessary for an algorithm \mathcal{A} to obtain a feasible solution $\mathcal{A}(I)$, for an arbitrary instance I of length n . Such an algorithm is called a *polynomial-time* algorithm, if its time complexity $T(n)$ is restricted by some polynomial

$\text{poly}(n)$. In the literature, these algorithms are called *efficient* [Garey and Johnson, 1979].

Sometimes it is possible to construct an approximation scheme for some NP-hard problems. The instance I has a *Polynomial-Time Approximation Scheme* (PTAS), if for an arbitrary $\varepsilon > 0$ there is an $(1 + \varepsilon)$ -approximation algorithm \mathcal{A}_ε , whose time-complexity $T(n) = \text{poly}_\varepsilon(n)$ for some polynomial $\text{poly}_\varepsilon(n)$.

In the case, when $\text{poly}_\varepsilon(n) = f(1/\varepsilon) \cdot n^c$ for some $c \geq 1$, then this PTAS is called *Efficient Polynomial-Time Approximation Scheme* (EPTAS). Furthermore, if $T(n) = \text{poly}(\frac{1}{\varepsilon}, n)$, then we have a *Fully Polynomial-Time Approximation Scheme* (FPTAS). Finally, when $T(n) = n^{\text{poly}_\varepsilon(\log n)}$, such an approximation scheme is called *Quasi-Polynomial-Time Approximation Scheme* (QPTAS).

In the field of approximation algorithms with theoretical bounds, the most significant results were achieved for the settings of CVRP in finite-dimensional Euclidean spaces. All of them date back to the celebrated papers by M. Haimovich and A. Rinnooy Kan [Haimovich and Rinnooy Kan, 1985] and S. Arora [Arora and Safra, 1998]. At the moment, the most general result for the CVRP on the Euclidean plane is the QPTAS proposed by A. Das and C. Mathieu [Das and Mathieu, 2015]. Further, for the planar CVRP with restricted capacity growth, there are known several Polynomial-Time Approximation Schemes (PTAS), among them the algorithm proposed in [Adamaszek et al., 2010] appears to be the state-of-the-art. This PTAS allows to find an $(1 - \varepsilon)$ -approximate solution of the problem in polynomial time provided $q \leq 2^{\log^{\delta(\varepsilon)} n}$. The approach proposed in [Haimovich and Rinnooy Kan, 1985] was extended to many modifications of the problem including the CVRP settings in Euclidean spaces of an arbitrarily fixed dimension [Khachai and Dubinin, 2017, Khachay and Dubinin, 2016, Khachay and Zaytseva, 2015], additional time windows

constraints [Khachai and Ogorodnikov, 2019b, Khachay and Ogorodnikov, 2018], and heterogeneity of demand [Khachay and Ogorodnikov, 2019].

Thus, until now, the class of metric CVRP instances approximable by PTAS or QPTAS was exhausted by the Euclidean settings of the problem except maybe some special cases investigated in [Becker et al., 2019, Khachai and Ogorodnikov, 2019a]. For a long time, the similar theoretic gap remained for the very close Traveling Salesman Problem, until the pioneering papers by K. Talwar [Talwar, 2004], and Y. Bartal et al. [Bartal et al., 2016] providing an opportunity for the extension of famous Arora’s PTAS [Arora, 1998] to the universe of metric spaces of a fixed doubling dimension.

1.4 Research motivation

The straightforward practical solution of both TSP, VRP as well as their generalizations proved to be complicated since those problems are NP-hard in the strong sense, intractable and hard to approximate [Papadimitriou, 1977, Haimovich and Rinnooy Kan, 1985]. Therefore, the researchers have to make a compromise between the running-time and the desired accuracy of the obtained solutions.

According to the literature, the algorithm design and analysis for NP-hard combinatorial optimization problems develops in three main research directions [Pardalos et al., 2013].

The first direction is related to exact (optimal) algorithms, including Branch-and-Bound, Branch-and-Cut, Branch-and-Price and their combinations [see e.g. Ascheuer et al., 2000, Cire and van Hoeve, 2013, Gouveia and Ruthmair, 2015, Clautiaux et al., 2019]. Having a polyhedral theory as their theoretical background [Balas et al., 1995, Chrétienne et al., 2014], these methods can obtain high quality solutions, either optimal or close to optimal. For certain classes of NP-hard problems, sometimes

it is possible to provide a general solver framework [Pessoa et al., 2020]. Another interesting feature of those methods is the possibility to stop the computations after reaching a certain gap value, thus reducing the total time. Unfortunately, due to the nature of NP-hard problems, those algorithms may encounter computational issues with instances of a large size.

In order to resolve this, the second research direction is actively developing, which consists of numerous heuristics, meta-heuristics and matheuristics [Vidal et al., 2013a, Archetti and Speranza, 2014, Glover et al., 2001]. Among them are local search algorithms [Arnold and Sörensen, 2019, Avdoshin and Beresneva, 2019], Tabu search [Qiu et al., 2018], Variable Neighborhood Search (VNS) [Frifita and Masmoudi, 2020, Polat, 2017], machine learning [Nazari et al., 2018], evolutionary [Vidal et al., 2013b] and bio-inspired algorithms [Necula et al., 2017, Su-Ping and Wei-Wei, 2016], memetic algorithms [Gutin and Karapetyan, 2010], and several efficient heuristic solvers, such as Lin-Kernighan-Helsgaun [Helsgaun, 2015] and ALNS-based solver [Smith and Imeson, 2017]. However, while being able to obtain solutions of remarkable quality, an absence of any theoretical guarantees implies additional computational expenses related to numerical performance evaluation and possible tuning of their internal parameters during the transition to any novel class of instances.

The last direction is related to developing approximation algorithms with a priori performance guarantees. Here, we should mention the fundamental results by [Christofides, 1976, Serdyukov, 1978] and [Haimovich and Rinnooy Kan, 1985]. In particular, in these papers it was proven that the metric settings of TSP and VRP are in APX, i.e. both problems can be approximated in polynomial time within constant factors. In addition, the results obtained in those papers are used as a building blocks for the numerous approximation algorithms for the related problems. One of the

most interesting methods in this field are Polynomial Time Approximation Schemes (PTAS), that allow one to obtain the approximate solutions with a predefined accuracy [Arora, 1998, Mitchell, 1999]. Eventually, this PTAS was extended to more general metrics in [Talwar, 2004, Bartal et al., 2016]. For the Capacitated Vehicle Routing Problem, the most important approximation result was obtained in [Das and Mathieu, 2015].

To the best of our knowledge, according to those research directions, the problems considered in this thesis were developed unevenly. In particular, the PCGTSP was still lacking efficient optimal algorithms and heuristics, while those were proposed for close settings - Generalized Traveling Salesman Problem [Fischetti et al., 1997] and Sequential Ordering Problem [Balas et al., 1995]. In addition, there were certain approximation algorithms for another close variant - the Traveling Salesman Problem with Neighborhoods [Chan and Jiang, 2018]. Therefore, we can state the first research question.

RQ 1. For the PCGTSP: propose MILP-formulations of high relaxation ability, introduce families of valid inequalities and their liftings, obtain results describing facet structure of PCGTS polytope, and finally propose and implement high-performance Branch-and-Cut algorithm.

On the other hand, the CVRP is a well studied combinatorial optimization problem with respect to all three aforementioned directions. It has a plenty of efficient exact methods [see ex. Pessoa et al., 2020] and heuristics [Gendreau and Potvin, 2019]. In addition, there are known results in the field of approximation algorithms for this problem [Adamaszek et al., 2010, Das and Mathieu, 2015, Khachay and Ogorodnikov, 2018]. However, the most of them were proposed for the Euclidean spaces of fixed dimension. At the same time, from 2016, we know breakthrough approximation results for TSP and some related problems in much more general metric spaces

[Bartal et al., 2016, Chan and Jiang, 2018]. Thus, we have another research question.

RQ 2. For the CVRP: extending the seminal results of Das and Mathieu, prove the approximability of the problem by a QPTAS in a metric space of an arbitrary fixed doubling dimension.

First two research questions are entirely related to the field of combinatorial optimization. However, it is important to pay attention to applications of Operations Research, to which the obtained results can contribute. To this end, we consider the CPP as an industrial application of the PCGTSP. Although, the CPP appears to be widely studied and well-known to the engineering community, a general solution framework still remains undeveloped for this problem. Therefore, we formulate the last research question.

RQ 3. For the discrete CPP: propose a general solution framework taking into account all the known industrial features of the cutting process.

1.5 Contributions

In this section, we briefly overview our results with respect to the aforementioned research questions.

1.5.1 Precedence Constrained Generalized Traveling Salesman Problem

In this chapter, we introduce polyhedral study, formulations, and branch-and-cut algorithm for the PCGTSP. Contribution of this chapter is three-fold:

(i) by evolving the inductive framework developed in [Fischetti et al., 1995] for the symmetric GTSP, we establish dimension of the PCGTS

polytope and extend the sufficient facet-inducing conditions for π - and σ -inequalities proved initially in [Balas et al., 1995] for the PCATSP, to the more general case of the PCGTSP;

(ii) relying on the known results on formulations for the PCATSP [Sarin et al., 2005, Gouveia and Pesneau, 2006, Gouveia et al., 2018], we propose novel valid inequalities for the PCGTSP and a family of compact and exponential-size MILP-models for this problem aimed to increase tightness of their lower bounds and speed-up the solution procedure for the appropriate LP-relaxations;

(iii) by combining the best formulations (in terms of lower bounds and running times) and the PCGLNS primal heuristic, for the first time, we propose several variants of the branch-and-cut algorithm for the PCGTSP, and compare their performance with aforementioned best known results and our adaptation of the state-of-the-art algorithm proposed in [Gouveia and Ruthmair, 2015] for the SOP.

As a result, the number of PCGTSPLIB instances solved to optimality has increased twice, to 24 out of 40 instances. Furthermore, the carried out numerical evaluation confirm that the considered MILP-models and branch-and-cut algorithm for the PCGTSP benefit well from the incorporation of the predecessor/successor inequalities.

Chapter 2 has a following structure. In Section 2.1, we give a mathematical statement of the considered problem, introduce some necessary definitions and notation, discuss the instance preprocessing, and describe the compact MILP-model used throughout the chapter. In Section 2.2, we propose novel families of valid inequalities for the problem in question and explain the corresponding separation procedures. Section 2.3 deals with the polyhedral study of the PCGTSP. By extending the seminal results of Balas et al. [1995] and Fischetti et al. [1995], we establish dimension of the PCGTS polytope and prove the conditions sufficient for π -

and σ -inequalities to be facet-inducing. Further, Section 2.4 represents the proposed formulations for the PCGTSP, while Section 2.5 gives an overview of our branch-and-cut algorithm. In Section 2.6 we report the results of the numerical evaluation, both for the proposed formulations and suggested variants of the branch-and-cut algorithm.

The results of the Chapter 2 were presented at the IMB Seminaire Optimal Plus at the University of Bordeaux, and published in the European Journal of Operational Research [Khachai et al., 2023].

1.5.2 Discrete Cutting Path Problems

In the following chapter, we propose novel generic solution framework with accuracy guarantees. Contribution of this chapter is two-fold:

- (i) for each variant of the discrete CPP from the literature, we propose a polynomial time cost efficient reduction to auxiliary instances of the PCGTSP;
- (ii) relying on the results of numerical evaluation carried out against industrial instances, we show that the algorithms developed for the PCGTSP can be successfully applied in order to solve the CPP efficiently as well. In particular, the PCGLNS heuristic provides good quality solutions in a few minutes with high probability.

The rest of the chapter is organized as follows. Section 3.1 presents our approach to reduce different settings of the discrete CPP to the same combinatorial optimization problem — the Precedence Constrained Traveling Salesman Problem. In Section 3.2, we describe the generic solution framework for the discrete CPP. Further, in Section 3.3, we report and discuss the results of the computational evaluation of the proposed framework against real-life discrete CPP instances.

The results of Chapter 3 were presented partially at the 10th IFAC Conference MIM 2022 on Manufacturing modelling, Management and Control, and in full at the 22nd International Conference on Mathematical Optimization Theory and Operations Research (MOTOR 2023). The paper was submitted for publication in the International Journal of Production Research.

1.5.3 Capacitated Vehicle Routing Problem

In the final research chapter, we propose an efficient approximation in metric spaces of a fixed doubling dimension for the CVRP. Contributions of this chapter are as follows:

- (i) by extending the approach of [Bartal et al., 2016] for metric TSP, we show that the approach proposed by Das and Mathieu for the efficient approximation of the Euclidean CVRP [Das and Mathieu, 2015] can be extended to the significantly wider class of metric CVRP settings. We prove that this framework combined with recent approximation results obtained for the metric TSP, for any given $\varepsilon > 0$, provides a $(1 + O(\varepsilon))$ -approximate solution for the CVRP formulated in a metric space of an arbitrary fixed doubling dimension $d > 1$.
- (ii) nevertheless, broadly speaking, the approximation scheme obtained by the straightforward application of the Das and Mathieu framework is no longer a QPTAS in general metric space of a fixed doubling dimension, even for a fixed arbitrary capacity $q > 2$. Therefore, in this chapter, we introduce a refinement of their algorithm by replacing the stage of exhaustive search with our internal dynamic program, such that the resulting scheme becomes a QPTAS again, at least for $q = \text{polylog}(n)$.

The remainder of Chapter 4 has a following structure. In Section 4.1, we present the statement of the metric CVRP. Then, Section 4.2 recalls some necessary notation regarding the metrics of a fixed doubling dimension. Main results of the chapter are presented in Section 4.3 and Section 4.4. In particular, Section 4.3 deals with approximation properties of the proposed scheme, whilst, in Section 4.4, we prove an upper bound of its running time.

The results of Chapter 4 were published in the Journal of Global Optimization [Khachay et al., 2021a].

Chapter 2

Precedence Constrained Generalized Traveling Salesman Problem: Polyhedral Study, Formulations, and Branch-and-Cut Algorithm

2.1 Problem statement

An instance of PCGTSP is given by a triple $(G, \mathcal{C}, \mathcal{G})$, where

- the complete loopless arc-weighted digraph $G = (V, E, c)$, $|V| = n$, defines a groundset network supplemented with transportation costs $c(u, v)$ for an arbitrary arc $(u, v) \in E$;

- the partition $\mathcal{C} = \{C_1, \dots, C_m\}$ splits the nodeset V of the graph G into m non-empty pairwise-disjoint *clusters*, where the cluster C_1 is referred to as *depot*;

- the directed acyclic graph $\mathcal{G} = (\mathcal{C}, A)$ defines a partial order (*precedence constraints*) on the set of clusters \mathcal{C} . Further, without loss of generality, we assume \mathcal{G} to be *transitively closed*, i.e. $(C_i, C_j) \in A$ and $(C_j, C_k) \in A$ imply $(C_i, C_k) \in A$, and that $(C_1, C_p) \in A$ for any $p \in \{2, \dots, m\}$.

A closed m -tour T is called a *feasible* solution of the PCGTSP, if

- it departs from and arrives at some node $v_1 \in C_1$;

- it visits each cluster $C_p \in \mathcal{C}$ exactly once;
- the tour T is consistent with the partial order \mathcal{G} , i.e. no cluster C_q can be visited by the tour T before its arbitrary predecessor in the order \mathcal{G} .

The cost of a tour $T = (v_1, v_2, \dots, v_m)$ is the sum of costs of its arcs $cost(T) = c(v_m, v_1) + \sum_{i=1}^{m-1} c(v_i, v_{i+1})$. The objective of the PCGTSP is to find a feasible m -tour of the minimum cost.

2.1.1 Preliminaries

We start with some necessary definitions and notation. For any pair of clusters C_p and C_q except the depot cluster C_1 , for which $(C_p, C_q) \in A$, we refer to C_p as a *predecessor* of C_q (and C_q as a *successor* of C_p) or shortly $C_p \prec C_q$. Further, to any non-depot cluster C , we assign subsets $\pi(C) = \{C_i \neq C_1 : C_i \prec C\}$ and $\sigma(C) = \{C_i \neq C_1 : C \prec C_i\}$ of its predecessors and successors, respectively. This notation can be easily extended to an arbitrary nonempty subset of clusters $\mathcal{C}' \subset \mathcal{C} \setminus \{C_1\}$: $\pi(\mathcal{C}') = \cup_{C \in \mathcal{C}'} \pi(C)$, $\sigma(\mathcal{C}') = \cup_{C \in \mathcal{C}'} \sigma(C)$. In turn, by $\tilde{\pi}(C)$ and $\tilde{\sigma}(C)$ we denote the subsets of $\pi(C)$ and $\sigma(C)$ respectively consisting of the direct predecessors (parents) and successors (children) of the cluster C . Finally, by $\mathfrak{C}^+ = \cup_{i=2}^m \pi(C_i)$ and $\mathfrak{C}^- = \cup_{i=2}^m \sigma(C_i)$ we denote the sets of all aforementioned predecessors and successors, respectively.

If, for $C \neq C_1$, $\pi(C) \cup \sigma(C) = \emptyset$, we call C a *free* cluster. In terms of polyhedral results, we restrict ourselves to the setting of PCGTSP with a singleton free cluster, which we call C_{Balas} .

In the following, by $C(v)$ we denote (the only) cluster that contains an arbitrary node $v \in V$. We call v a *non-individual* node, if $|C(v)| > 1$, otherwise v is called *individual*. To simplify the problem at hand, we use the instance preprocessing technique proposed in [Balas et al., 1995]. We exclude any arc $(i, j) \in E$, for which at least one of the following options

holds:

$$(i \in C_1) \ \& \ (j \in \mathfrak{C}^-) \quad (2.1)$$

$$(i \in \mathfrak{C}^+) \ \& \ (j \in C_1) \quad (2.2)$$

$$C(j) \prec C(i) \quad (2.3)$$

$$\exists \tilde{C} \in \mathfrak{C} = \mathfrak{C}^- \cap \mathfrak{C}^+ : (C(i) \prec \tilde{C}) \ \& \ (\tilde{C} \prec C(j)) \quad (2.4)$$

$$C(i) = C(j). \quad (2.5)$$

For any proper subset $\emptyset \neq S \subset V$, we use the standard notation $\delta^-(S) = \{(i, j) \in E : i \notin S, j \in S\}$, $\delta^+(S) = \{(i, j) \in E : i \in S, j \notin S\}$, and $\delta(S) = \delta^+(S) \cup \delta^-(S)$ for the appropriate incoming and outgoing cuts, and their union, respectively.

In the case of a singleton $S = \{v\}$, we use simple notation $\delta^+(v)$ and $\delta^-(v)$.

Without loss of generality, we assume that graph G has no isolated nodes after preprocessing. Furthermore, we can assume that, for any node $v \in V$, $\delta^+(v) \neq \emptyset$ and $\delta^-(v) \neq \emptyset$. As a simple consequence, we obtain that $\delta^+(C) \neq \emptyset$ and $\delta^-(C) \neq \emptyset$ for any cluster C as well.

2.1.2 Compact MILP model

To obtain a basic compact model for the considered problem, we extend the known L1PCATSPxy formulation, proposed in [Sarin et al., 2005] for the PCATSP, which is the best performer among compact models in terms of LP-relaxation bounds for that problem.

For any $(i, j) \in E$ and node $v \in V$, we introduce the following binary decision variables:

$$x_{ij} = \begin{cases} 1, & \text{if } (i, j) \text{ belongs to the solution} \\ 0, & \text{otherwise,} \end{cases} \quad z_v = \begin{cases} 1, & \text{if } v \text{ is visited by the solution} \\ 0, & \text{otherwise.} \end{cases}$$

In addition, we introduce auxiliary variables y_{pq} and u_{pq} :

$$y_{pq} = \begin{cases} 1, & \text{if cluster } C_p \text{ precedes } C_q \text{ in the solution (not necessarily immediately)} \\ 0, & \text{otherwise,} \end{cases}$$

$$u_{pq} = \begin{cases} 1, & \text{if cluster } C_p \text{ immediately precedes } C_q \text{ in the solution} \\ 0, & \text{otherwise.} \end{cases}$$

The proposed MILP model for the PCGTSP is as follows:

$$\min \sum_{(i,j) \in E} c_{ij} x_{ij}, \quad (2.6)$$

$$\text{s.t.} \quad \sum_{i \in C_k} z_i = 1 \quad (k \in \{1, \dots, m\}) \quad (2.7)$$

$$\sum_{(i,j) \in \delta^+(i)} x_{ij} = z_i \quad (i \in V) \quad (2.8)$$

$$\sum_{(i,j) \in \delta^-(i)} x_{ji} = z_i \quad (i \in V) \quad (2.9)$$

$$\sum_{q=1, q \neq p}^m u_{pq} = 1 \quad (p \in \{1, \dots, m\}), \quad \sum_{p=1, p \neq q}^m u_{pq} = 1 \quad (q \in \{1, \dots, m\}) \quad (2.10)$$

$$\sum_{i \in \delta^+(C_p)} \sum_{j \in \delta^-(C_q)} x_{ij} = u_{pq} \quad (p, q \in \{1, \dots, m\}, p \neq q) \quad (2.11)$$

$$(y_{pq} + u_{qp}) + y_{qr} + y_{rp} \leq 2 \quad (p, q, r \in \{2, \dots, m\}, p \neq q \neq r) \quad (2.12)$$

$$u_{pq} - y_{pq} \leq 0 \quad (p, q \in \{2, \dots, m\}, p \neq q) \quad (2.13)$$

$$y_{pq} + y_{qp} = 1 \quad (\{p, q\} \subset \{2, \dots, m\}) \quad (2.14)$$

$$y_{pq} = 1 \quad (p, q \in \{2, \dots, m\}, C_p \prec C_q) \quad (2.15)$$

$$x_{ij}, z_i \in \{0, 1\}, u_{pq} \geq 0, y_{pq} \geq 0 \quad (2.16)$$

The objective is to minimize the total traveling cost (3.1). Constraints (3.2) ensure that exactly one node from each cluster is visited. Constraints

(3.3) and (3.4) are flow conservation constraints in terms of nodes, while constraints (2.10) are flow conservation constraints in terms of clusters. Technical constraints (3.7) establish the link between the decision and auxiliary variables. Similarly to the initial L1PCATSP_{xy} model, constraints (2.12)-(2.15) ensure subtour elimination and establish the given precedence constraints simultaneously.

By evolving the arguments of [Sarin et al., 2005], it is easy to verify the following observation.

Observation 1. *For any feasible solutions $[x', z', u', y']$ and $[x'', z'', u'', y'']$ of the model (3.1)-(3.12), $(x' = x'') \wedge (z' = z'') \Rightarrow (u' = u'') \wedge (y' = y'')$.*

2.2 Valid inequalities

In this section, we extend to the case of PCGTSP some known families of valid inequalities initially introduced in papers [Balas et al., 1995, Gouveia and Ruthmair, 2015, Gouveia et al., 2018] for the PCATSP. It is convenient to specify these inequalities in terms of the following standard notation. For any non-empty disjoint cluster subsets $\mathcal{U}', \mathcal{U}'' \subset \mathcal{C}$,

$$x(\mathcal{U}', \mathcal{U}'') = \sum_{C_p \subset \mathcal{U}'} \sum_{C_q \subset \mathcal{U}''} \sum_{i \in C_p} \sum_{j \in C_q} x_{ij} \equiv \sum_{C_p \subset \mathcal{U}'} \sum_{C_q \subset \mathcal{U}''} u_{pq}.$$

2.2.1 Predecessor and successor inequalities

Proposition 1. *For an arbitrary non-empty $\mathfrak{S} \subset \mathfrak{C} \setminus \{C_1\}$, $\bar{\mathfrak{S}} = \mathfrak{C} \setminus \mathfrak{S}$, the predecessor-inequality (π -inequality):*

$$x(\mathfrak{S} \setminus \pi(\mathfrak{S}), \bar{\mathfrak{S}} \setminus \pi(\mathfrak{S})) \geq 1 \quad (2.17)$$

is valid for the PCGTSP.

Proof. Let T be an arbitrary tour that satisfies the precedence constraints and C_p be the last cluster in \mathfrak{S} visited by T . Then, $C_p \in \mathfrak{S} \setminus \pi(\mathfrak{S})$

and for the next cluster visited by T , $C_q \in \bar{\mathfrak{S}} \setminus \pi(\mathfrak{S})$. Such a cluster exists, since the tour T should depart from and arrive at C_1 . Therefore, $x(\mathfrak{S} \setminus \pi(\mathfrak{S}), \bar{\mathfrak{S}} \setminus \pi(\mathfrak{S})) \geq u_{pq} = 1$. \square

Since the following two propositions can be treated similarly, we skip their proofs for the sake of brevity.

Proposition 2. *For an arbitrary non-empty $\mathfrak{S} \subset \mathfrak{C} \setminus \{C_1\}$, $\bar{\mathfrak{S}} = \mathfrak{C} \setminus \mathfrak{S}$, the successor-inequality (σ -inequality):*

$$x(\bar{\mathfrak{S}} \setminus \sigma(\mathfrak{S}), \mathfrak{S} \setminus \sigma(\mathfrak{S})) \geq 1 \quad (2.18)$$

is valid for the PCGTSP.

Proposition 3. *Let $\mathcal{X}, \mathcal{Y} \subset \mathfrak{C} \setminus \{C_1\}$ be non-empty subsets such that, for an arbitrary clusters $C' \in \mathcal{X}$ and $C'' \in \mathcal{Y}$, $C' \prec C''$, and let $\mathcal{Q} = \{C_1\} \cup \pi(\mathcal{X}) \cup \sigma(\mathcal{Y})$. Then for any $\mathfrak{S} \subset \mathfrak{C}$, $\bar{\mathfrak{S}} = \mathfrak{C} \setminus \mathfrak{S}$ such that $\mathcal{X} \subseteq \mathfrak{S}$, $\mathcal{Y} \subseteq \bar{\mathfrak{S}}$, the (π, σ) -inequality:*

$$x(\mathfrak{S} \setminus \mathcal{Q}, \bar{\mathfrak{S}} \setminus \mathcal{Q}) \geq 1 \quad (2.19)$$

is valid for the PCGTSP.

2.2.2 Precedence cycle breaking inequalities

For some natural t , consider a subset $\mathfrak{C}' = \{C_{i_1}, \dots, C_{i_{2t+1}}\} \subset \mathfrak{C} \setminus \{C_1\}$, such that $C_{i_1} \prec \dots \prec C_{i_{2t+1}}$. Introduce the subsets $\mathfrak{C}'_{odd} = \{C_{i_{2s+1}} : s \in \{0, \dots, t\}\}$ and $\mathfrak{C}'_{even} = \{C_{i_{2s}} : s \in \{1, \dots, t\}\}$ of \mathfrak{C}' , that contain C_{i_j} with odd and even j respectively.

Proposition 4. *For an arbitrary non-empty $\mathfrak{S} \subset \mathfrak{C} \setminus \{C_1\}$, $\bar{\mathfrak{S}} = \mathfrak{C} \setminus \mathfrak{S}$, such that $\mathfrak{C}'_{odd} \subset \mathfrak{S}$ and $\mathfrak{C}'_{even} \subset \bar{\mathfrak{S}}$,*

$$x(\mathfrak{S}, \bar{\mathfrak{S}}) \geq t + 1 \quad (2.20)$$

is valid for the PCGTSP.

Proof. Indeed, consider an arbitrary feasible tour T . Since clusters $C_{i_1} \dots C_{i_{2t+1}}$ are linearly ordered and $C_1 \notin \mathfrak{S}$, the tour T crosses the border from \mathfrak{S} to $\bar{\mathfrak{S}}$ at least $t + 1$ times. \square

Following [Gouveia and Ruthmair, 2015], without loss of generality, we can assume that $C_{i_j} \in \tilde{\pi}(C_{i_{j+1}})$ for each $j \in \{1, \dots, 2t\}$. Furthermore, we can strengthen inequality (3.16) as follows.

Proposition 5. *For an arbitrary non-empty $\mathfrak{S} \subset \mathfrak{C} \setminus \{C_1\}$, $\bar{\mathfrak{S}} = \mathfrak{C} \setminus \mathfrak{S}$, such that $\mathfrak{C}'_{\text{odd}} \subset \mathfrak{S}$ and $\mathfrak{C}'_{\text{even}} \subset \bar{\mathfrak{S}}$, the condition $\tilde{\sigma}(C_{i_{2t+1}}) \not\subset \mathfrak{S}$ implies the validity of inequality*

$$x(\mathfrak{S} \setminus \mathfrak{S}', \bar{\mathfrak{S}} \setminus \mathfrak{S}') \geq t + 1, \quad (2.21)$$

where $\mathfrak{S}' = \pi(C_{i_1}) \cup \sigma(C_{i_{2t+1}}) \setminus \tilde{\sigma}(C_{i_{2t+1}})$.

2.2.3 Single-option inequalities

In this subsection, we extend the family of simple (but powerful) inequalities proposed in [Gouveia and Ruthmair, 2015] for the PCATSP, whose validity can be easily obtained from (3.2)-(3.7) and precedence constraints.

Proposition 6. *For an arbitrary $\{C_i, C_j\} \subset \mathfrak{C} \setminus \{C_1\}$, the following inequalities*

$$u_{ij} + u_{ji} + u_{kl} + u_{lk} \leq 1 \quad (C_k \in \pi(C_i), C_l \in \sigma(C_j)) \quad (2.22)$$

$$u_{ij} + u_{ji} + \sum_{C_l \in \sigma(C_j)} u_{kl} \leq 1 \quad (C_k \in \pi(C_i)) \quad (2.23)$$

$$u_{ij} + u_{ji} + \sum_{C_l \in \sigma(C_j)} u_{lk} \leq 1 \quad (C_k \in \pi(C_i)) \quad (2.24)$$

$$u_{ij} + u_{ji} + \sum_{C_k \in \pi(C_i)} u_{kl} \leq 1 \quad (C_l \in \sigma(C_j)) \quad (2.25)$$

$$u_{ij} + u_{ji} + \sum_{C_k \in \pi(C_i)} u_{lk} \leq 1 \quad (C_l \in \sigma(C_j)) \quad (2.26)$$

are valid for the PCGTSP.

2.2.4 Strengthened precedence variables and network flow based inequalities

The authors of [Gouveia et al., 2018] introduced a novel exponential-size families of valid inequalities augmented with polynomial-time separation procedures, their strengthened counterparts, and the related formulations for the PCATSP. Comprehensive numerical analysis carried out there showed that more tight lower bounds were provided by the formulations based on strengthened inequalities. Therefore, in this chapter, we restrict ourselves only on extension to the PCGTSP of these families.

Proposition 7. *For an arbitrary clusters C_p and C_q not equal to C_1 , where $p \neq q$, the strengthened simple-cut inequality $x(\mathfrak{S}, \bar{\mathfrak{S}}) \geq y_{pq}$ is valid for the PCGTSP, for any partition*

$$(\mathfrak{S}, \bar{\mathfrak{S}}) \text{ of } (\mathfrak{C} \setminus C_{pq}^1) \cup \{C_p, C_q\}, \text{ such that } C_p \in \mathfrak{S}, C_q \in \bar{\mathfrak{S}}, \quad (2.27)$$

$$(\mathfrak{S}, \bar{\mathfrak{S}}) \text{ of } (\mathfrak{C} \setminus C_{pq}^2) \cup \{C_1, C_p\}, \text{ such that } C_1 \in \mathfrak{S}, C_p \in \bar{\mathfrak{S}}, \quad (2.28)$$

$$(\mathfrak{S}, \bar{\mathfrak{S}}) \text{ of } (\mathfrak{C} \setminus C_{pq}^3) \cup \{C_1, C_q\}, \text{ such that } C_q \in \mathfrak{S}, C_1 \in \bar{\mathfrak{S}}, \quad (2.29)$$

where $C_{pq}^1 = \{C_1\} \cup \pi(C_p) \cup \sigma(C_q)$, $C_{pq}^2 = \{C_q\} \cup \sigma(C_p) \cup \sigma(C_q)$ and $C_{pq}^3 = \{C_p\} \cup \pi(C_p) \cup \pi(C_q)$.

Proposition 8. *For an arbitrary triple (C_p, C_q, C_r) of distinct clusters not equal to C_1 , the strengthened GDDL inequality¹*

$$x(\mathfrak{S}, \bar{\mathfrak{S}}) \geq y_{pr} + y_{rq} \quad (2.30)$$

¹Generalized Disaggregated Desrochers-Laporte inequality

is valid for the PCGTSP for any partition $(\mathfrak{S}, \bar{\mathfrak{S}})$ of $(\mathfrak{C} \setminus \mathfrak{C}_{pqr}) \cup \{C_1, C_p, C_q, C_r\}$, such that $\{C_1, C_r\} \subset \mathfrak{S}$, $\{C_p, C_q\} \subset \bar{\mathfrak{S}}$ and $\mathfrak{C}_{pqr} = (\sigma(C_p) \cap \sigma(C_q)) \cup (\pi(C_r) \cap \sigma(C_p)) \cup (\sigma(C_q) \cap \sigma(C_r))$.

Proposition 9. For an arbitrary triple (C_p, C_q, C_r) of distinct clusters not equal to C_1 , the strengthened Reversed GDDL inequality

$$x(\mathfrak{S}, \bar{\mathfrak{S}}) \geq y_{pr} + y_{rq} \quad (2.31)$$

is valid for the PCGTSP for any partition $(\mathfrak{S}, \bar{\mathfrak{S}})$ of $(\mathfrak{C} \setminus \mathfrak{C}_{pqr}^R) \cup \{C_1, C_p, C_q, C_r\}$, such that $\{C_p, C_q\} \subset \mathfrak{S}$, $\{C_1, C_r\} \subset \bar{\mathfrak{S}}$ and $\mathfrak{C}_{pqr}^R = (\pi(C_p) \cap \pi(C_r)) \cup (\pi(C_p) \cap \pi(C_q)) \cup (\sigma(C_r) \cap \pi(C_q))$.

Proposition 10. For an arbitrary triple (C_p, C_q, C_r) of distinct clusters not equal to C_1 , the strengthened 2-path inequality $x(\mathfrak{S}, \bar{\mathfrak{S}}) \geq y_{pq} + y_{qr} - 1$ is valid for the PCGTSP, for any partition

$$(\mathfrak{S}, \bar{\mathfrak{S}}) \text{ of } (\mathfrak{C} \setminus \mathfrak{C}_{pqr}^1) \cup \{C_1, C_p\}, \text{ such that } C_1 \in \mathfrak{S}, C_p \in \bar{\mathfrak{S}}, \quad (2.32)$$

$$(\mathfrak{S}, \bar{\mathfrak{S}}) \text{ of } (\mathfrak{C} \setminus \mathfrak{C}_{pqr}^2) \cup \{C_p, C_q\}, \text{ such that } C_p \in \mathfrak{S}, C_q \in \bar{\mathfrak{S}}, \quad (2.33)$$

$$(\mathfrak{S}, \bar{\mathfrak{S}}) \text{ of } (\mathfrak{C} \setminus \mathfrak{C}_{pqr}^3) \cup \{C_q, C_r\}, \text{ such that } C_q \in \mathfrak{S}, C_r \in \bar{\mathfrak{S}}. \quad (2.34)$$

$$(\mathfrak{S}, \bar{\mathfrak{S}}) \text{ of } (\mathfrak{C} \setminus \mathfrak{C}_{pqr}^4) \cup \{C_1, C_r\}, \text{ such that } C_r \in \mathfrak{S}, C_1 \in \bar{\mathfrak{S}}, \quad (2.35)$$

where $\mathfrak{C}_{pqr}^1 = \{C_q, C_r\} \cup \sigma(C_p) \cup \sigma(C_q) \cup \sigma(C_r)$, $\mathfrak{C}_{pqr}^2 = \{C_1, C_r\} \cup \pi(C_p) \cup \sigma(C_q) \cup \sigma(C_r)$, $\mathfrak{C}_{pqr}^3 = \{C_1, C_p\} \cup \pi(C_p) \cup \pi(C_q) \cup \sigma(C_r)$, and $\mathfrak{C}_{pqr}^4 = \{C_p, C_q\} \cup \pi(C_p) \cup \pi(C_q) \cup \pi(C_r)$, respectively.

Proposition 11. For an arbitrary quadruple (C_p, C_q, C_r, C_s) of distinct clusters not equal to C_1 , the strengthened 3v GDDL-like inequality $x(\mathfrak{S}, \bar{\mathfrak{S}}) \geq y_{pq} + y_{qr} + y_{rs} - 1$ is valid for the PCGTSP, for any partition:

$$(\mathfrak{S}, \bar{\mathfrak{S}}) \text{ of } (\mathfrak{C} \setminus \mathfrak{C}_{pqrs}^1) \cup \{C_p, C_q, C_r, C_s\}, \text{ such that } \{C_p, C_r\} \subset \mathfrak{S}, \{C_q, C_s\} \subset \bar{\mathfrak{S}}, \quad (2.36)$$

$$(\mathfrak{S}, \bar{\mathfrak{S}}) \text{ of } (\mathfrak{C} \setminus \mathfrak{C}_{pqrs}^2) \cup \{C_1, C_p, C_q, C_s\}, \text{ such that } \{C_p, C_s\} \subset \mathfrak{S}, \{C_q, C_1\} \subset \bar{\mathfrak{S}}, \quad (2.37)$$

$$(\mathfrak{S}, \bar{\mathfrak{S}}) \text{ of } (\mathfrak{C} \setminus \mathfrak{C}_{pqrs}^3) \cup \{C_1, C_p, C_r, C_s\}, \text{ such that } \{C_1, C_r\} \subset \mathfrak{S}, \{C_p, C_s\} \subset \bar{\mathfrak{S}}, \quad (2.38)$$

where $C_{pqrs}^1 = \{C_1\} \cup ((\pi(C_q) \cup \pi(C_r) \cup \sigma(C_s)) \cap (\pi(C_p) \cup \sigma(C_q)))$, $C_{pqrs}^2 = \{C_r\} \cup ((\pi(C_r) \cup \pi(C_s)) \cap (\pi(C_p) \cup \sigma(C_q) \cup \sigma(C_r)))$, and $C_{pqrs}^3 = \{C_q\} \cup ((\sigma(C_p) \cup \sigma(C_q)) \cap (\pi(C_q) \cup \pi(C_r) \cup \sigma(C_s)))$.

Proposition 12. *For an arbitrary quintuple $(C_p, C_q, C_k, C_r, C_s)$ of distinct clusters not equal to C_1 , the strengthened 4v GDDL-like inequality $x(\mathfrak{S}, \bar{\mathfrak{S}}) \geq y_{pq} + y_{qk} + y_{kr} + y_{rs} - 2$ is valid for the PCGTSP, for any partition*

$$(\mathfrak{S}, \bar{\mathfrak{S}}) \text{ of } \mathfrak{C} \setminus \mathfrak{C}_{pqkrs} \cup \{C_p, C_q, C_r, C_s\}, \text{ such that } \{C_p, C_r\} \subset \mathfrak{S}, \{C_q, C_s\} \subset \bar{\mathfrak{S}}, \quad (2.39)$$

where $\mathfrak{C}_{pqkrs} = \{C_1, C_k\} \cup ((\pi(C_p) \cup \sigma(C_q) \cup \sigma(C_k)) \cap (\pi(C_k) \cup \pi(C_r) \cup \sigma(C_s)))$.

Proofs of all the propositions of this subsection can be obtained by extension of the arguments presented in [Gouveia et al., 2018].

2.2.5 Separation procedures

All the aforementioned families of valid inequalities are augmented with polynomial-time separation procedures, which extend the seminal unit flow propagation approach introduced in [Balas et al., 1995]. In Algorithm 1, we present the proposed separation technique for π -inequalities (3.14).

For the sake of brevity, we restrict our further discussion to precedence cycle breaking inequalities (3.16). Other procedures evolve the similar results obtained in [Gouveia and Ruthmair, 2015, Gouveia et al., 2018] for the PCATSP and [Yuan et al., 2020] for the GTSP with time windows and can be retrieved from the supplemented source code (<https://github.com/EnsignDaniels/BnC>).

Algorithm 1 Separation technique for π -inequalities

Input: current (fractional) solution $(x_{ij}, z_i, u_{pq}, y_{pq})$, a non-depot cluster $C \neq C_1$
Output: π -inequality for an appropriate \mathfrak{S} (if any)

- 1: create an auxiliary cluster digraph $G_C = (\mathcal{C}_C, E_C)$, where $\mathcal{C}_C = \mathcal{C} \setminus \pi(C)$ and (C_p, C_q) belongs to E_C and has capacity u_{pq} if and only if $u_{pq} > 0$
- 2: find a maximum C -to- C_1 flow F in the graph G_C
- 3: **if** $val(F) < 1$ **then**
- 4: find a minimum cut $\mathcal{U}', \mathcal{U}'' \subset \mathcal{C}_C$
- 5: set $\mathfrak{S} = \mathcal{U}' \cup \pi(C)$ and $\bar{\mathfrak{S}} = \mathcal{C} \setminus \mathfrak{S} = \mathcal{U}''$
- 6: **return** π -inequality
$$x(\mathfrak{S} \setminus \pi(\mathfrak{S})), \bar{\mathfrak{S}} \setminus \pi(\mathfrak{S})) \geq 1$$

7: **end if**

Indeed, suppose we are given by the current fractional solution (x, z, u, y) . For a sequence of non-depot clusters $C_{i_1} \prec \dots \prec C_{i_{2t+1}}$, we construct an auxiliary cluster digraph $H = (\mathcal{C} \cup \{s, t\}, E')$, where s and t are artificial source and destination nodes connected by incapacitated arcs with clusters from \mathcal{C}'_{odd} and $\mathcal{C}'_{even} \cup \{C_1\}$, respectively. For each other arc $(C_p, C_q) \in E'$, its capacity is defined by u_{pq} . Next, if the value of the maximum s - t -flow in the digraph H appears to be less than $t + 1$, an arbitrary minimum cut $(\mathfrak{S}, \bar{\mathfrak{S}})$, where $\mathfrak{S} \subset \mathcal{C} \cup \mathcal{C}'_{odd} \setminus (\{C_1\} \cup \mathcal{C}'_{even})$ and $\bar{\mathfrak{S}} = \mathcal{C} \setminus \mathfrak{S}$, defines inequality (3.16) violated by the given solution.

2.3 Facets of the PCGTS polytope

In this section, we study a polyhedral structure of the PCGTS polytope. To elaborate this task, we employ the classic approach relying on *dimensions* of the studied polytope and its faces.

By definition, for an arbitrary polytope P , its dimension is equal to the dimension of its affine hull $\dim P = \dim(\text{aff}(P))$, which in turn is one less than the number of affinely independent extreme points this polytope.

An intersection of a polytope P with an arbitrary *support* hyperplane is called a *face* of this polytope. Usually, for the sake of convenience, the family of faces of a polytope is extended by *improper* faces \emptyset and P . A face F of a polytope P is called a *facet* (of this polytope), if $\dim F = \dim P - 1$.

The PCGTSP is an extension of an Equality GTSP (E-GTSP) introduced in [Fischetti et al., 1995], where E-GTSP polytope was denoted by $P^=$. Therefore, we keep the same notation for the PCGTS polytope, i.e. the convex hull of the incidence vectors $[x, z]$ encoding all the feasible tours of the problem in question. As it follows from Observation 1, $P^= = \text{conv}\{[x, z] \in \mathbb{R}^{E \cup V} : (3.2) - (3.12) \text{ holds}\}$. Since $[x, z]$ could be obviously extended to the feasible solution $[x, z, u, y]$ of (3.2)-(3.12), the polytope $P^=$ is isomorphic to the convex hull of the feasible set of the initial non-relaxed MILP model from Subsection 3.2.1. In the sequel, for the simplicity, we will not distinguish them. Our goal is to derive conditions sufficient for an arbitrary inequality

$$\alpha^T x - \beta^T z \geq \gamma \tag{2.40}$$

to induce a facet of the polytope $P^=$.

2.3.1 Dimension of the PCGTS polytope

In this section, we prove the following

Theorem 1. *For any instance of PCGTSP, the following equation:*

$$\dim(P^=) = |E| - n - m + 1 \tag{2.41}$$

holds.

To prove Theorem 1, we employ an inductive approach similar to [Fischetti et al., 1995] on the number of excessive nodes ρ within clusters:

$$\rho = \sum_{h=1}^m (|C_h| - 1) = n - m. \tag{2.42}$$

Here, the base case $\rho = 0$ corresponds to the Precedence Constrained Asymmetric Traveling Salesman Problem (PCATSP) and follows from

Theorem 2 (Balas et al. [1995]). *For an arbitrary instance of PCATSP, dimension of its polytope P_{ATSP}^- is as follows: $\dim P_{ATSP}^- = |E| - 2n + 1$.*

Remark 1. *In the paper by Balas et al. [1995], the polytope is denoted in \mathbb{R}^E . However, it can be unambiguously extended to $\mathbb{R}^{E \cup V}$ by setting $z_v = 1$ for each node $v \in V$ as it was done in [Fischetti et al., 1995].*

In order to prove the inductive step, we need additional notation and technical lemmas. Let inequality (2.40) be valid for P^- , i.e. $P^- \subset \{[x, z] \in \mathbb{R}^{E \cup V} : \alpha^T x \geq \beta^T z + \gamma\}$. Consider the appropriate face $\mathcal{H}(\alpha, \beta, \gamma) = P^- \cap \{[x, z] \in \mathbb{R}^{E \cup V} : \alpha^T x = \beta^T z + \gamma\}$ of the polytope P^- .

Further, to any non-individual node $v \in V$, we assign:

(i) a PCGTSP polytope P_v^- associated with the subgraph of G induced by $V \setminus \{v\}$,

(ii) the v -restriction of inequality (2.40) obtained by dropping variables z_v and x_e for all $e \in \delta(v)$,

(iii) the v -compatibility digraph of (2.40) $G_v^* = (V \setminus C(v), E_v^*)$, where

$$E_v^* = \{(i, j) : i, j \in V \setminus C(v), i \neq j, \exists [x, z] \in \mathcal{H}(\alpha, \beta, \gamma), x_{iv} = x_{vj} = 1\},$$

(iv) its bipartite representation B_v^* (see Bang-Jensen and Gutin [2009] and Fig. 2.1).

Lemma 1. *For any valid inequality $\alpha^T x \geq \beta^T z + \gamma$, and an arbitrary non-individual node $v \in V$, $\dim \mathcal{H}(\alpha, \beta, \gamma) \geq \dim \mathcal{H}(\alpha, \beta, \gamma)_v + \text{rank}(B_v^*)$ where $\mathcal{H}(\alpha, \beta, \gamma)_v$ is the face of polytope P_v^- induced by its v -restriction.*

Proof. Consider the matrix M , whose rows are extreme points of the face $\mathcal{H}(\alpha, \beta, \gamma)$ (Fig. 2.2). By construction, $\mathcal{H}(\alpha, \beta, \gamma)$ is contained in a hyperplane of $\mathbb{R}^{E \cup V}$ not passing through the origin (due to equation (3.2)). Therefore, for any subset of rows of M , the affine independence is equivalent to the linear one. Thus, $\dim \mathcal{H}(\alpha, \beta, \gamma) = \text{rank}(M) - 1$.

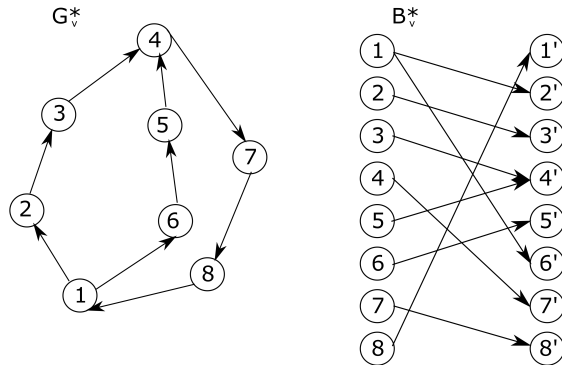


Figure 2.1: Example of a directed graph and its bipartite representation

extreme points	arcs				nodes					
	(1,2)	(1,3)	1	2	...	v	...	n
1	1	0			1	1		1		
2	1	0			1	0		0		
3	0	1						0		
...			•					•		
...				•				•		
...								•		

Figure 2.2: Matrix of extreme points of $\mathcal{H}(\alpha, \beta, \gamma)$

Matrix M can be represented as follows:

$$M = \begin{pmatrix} M_{11} & 0 & 0 \\ M_{21} & M_{22} & 1 \end{pmatrix},$$

where the last column corresponds to node v , and the columns left to it correspond to the arcs incident with v . By construction, block M_{11} corresponds to the extreme points of face $\mathcal{H}(\alpha, \beta, \gamma)_v$. Thus, $\text{rank}(M_{11}) = \dim \mathcal{H}(\alpha, \beta, \gamma)_v + 1$.

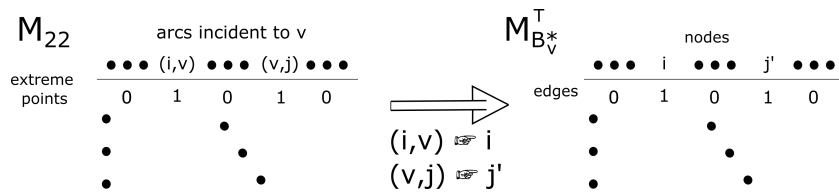


Figure 2.3: Block M_{22} and the incidence matrix of B_v^*

On the other hand, matrix M_{22} is located in the part of the tour visiting node v . By construction, it should visit it only once. Therefore, each row of M_{22} has exactly two 1s. Consider an arbitrary row of block M_{22} . Suppose that 1s are located in the columns (i, v) and (v, j) . Hence, in graph B_v^* , nodes i and j are adjacent and the considered row is a column in the incidence matrix $M_{B_v^*}$ of B_v^* . Thus, $M_{22} = M_{B_v^*}^T$ (see Fig. 2.3).

Therefore, $\text{rank}(M_{22}) = \text{rank}(B_v^*) = N_{B_v^*} - \kappa(B_v^*)$, where $N_{B_v^*}$ is a size of the nodeset of bipartite graph B_v^* and $\kappa(B_v^*)$ is the number of its connected components [see e.g. Biggs, 1974].

Finally, $\text{rank}(M) \geq \text{rank}(M_{11}) + \text{rank}(M_{22})$. Since $\text{rank}(M_{11}) = \dim \mathcal{H}(\alpha, \beta, \gamma)_v + 1$, $\text{rank}(M_{22}) = \text{rank}(B_v^*)$. Lemma 1 is proved. \square

The claim of Lemma 1 is valid for an arbitrary face $\mathcal{H}(\alpha, \beta, \gamma)$. Now, to determine dimension of polytope $P^=$, we consider its improper face $\mathcal{H}(0, 0, 0) = P^=$. To emphasize the associated bipartite graph B_v^* in this special case, denote it by \bar{B}_v^* .

Lemma 2. *For any non-individual node v , $\text{rank}(\bar{B}_v^*) = |\delta(v)| - 1$.*

Proof. We prove Lemma 2 by enumeration of all the possible options to relate cluster $C(v)$ with the given precedence constraints. In the sequel, we use the following notation. By $\tilde{\pi}$ and $\tilde{\sigma}$, for cluster $C(v)$, we denote subsets of nodes belonging to its direct parents and children, respectively. Similarly, we introduce subsets $\hat{\pi}$ and $\hat{\sigma}$ of nodes that belong to other ancestors and descendants of this cluster. In addition, by r , we denote a union of all clusters except C_{Balas} incompatible with $C(v)$.

Observation 2. *For any cluster, its parents (if any) are mutually incompatible. For its children the same claim is valid as well.*

Case 1 ($\tilde{\pi} \neq \emptyset$ and $\tilde{\sigma} = \emptyset$). In this case, cluster $C(v)$ is one of the minimal descendants in the given partial order. Here, for cut $\delta(v)$ in graph

G (see Fig. 2.4), we have $|\delta(v)| = |C_1| + |\tilde{\pi}| + 2|r| + 2$, since $|C_{Balas}| = 1$. Consider the appropriate bipartite graph \bar{B}_v^* (Fig. 2.5). It has

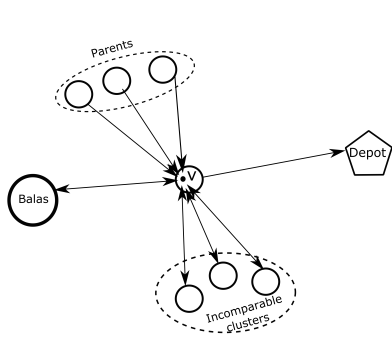


Figure 2.4: Cut $\delta(v)$ for Case 1

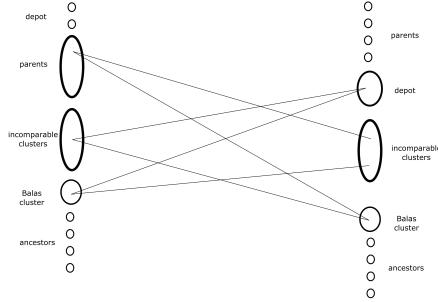


Figure 2.5: Bipartite graph \bar{B}_v^* for Case 1

$N_{\bar{B}_v^*} = 2|C_1| + 2|\hat{\pi}| + 2|\tilde{\pi}| + 2|r| + 2$ nodes. By definition, an arbitrary node i from the left part and j' from the right part of graph \bar{B}_v^* are incident if and only if there is a feasible tour with the fragment $i-v-j'$. If such an arc exists, then graph \bar{B}_v^* has a complete bipartite subgraph, whose parts are induced by clusters $C(i)$ and $C(j')$. In Fig. 2.5, we encode such subgraphs by straight line segments. By construction, all non-isolated nodes of \bar{B}_v^* belong to the only connected component. Furthermore, the number of connected components is $\kappa(\bar{B}_v^*) = 1 + |C_1| + 2|\hat{\pi}| + |\tilde{\pi}|$. Indeed, for instance, verify the incidence of some $i \in \tilde{\pi}$ and node j' corresponding to the only node of cluster C_{Balas} . Take an arbitrary node $v_1 \in C_1$ from the depot

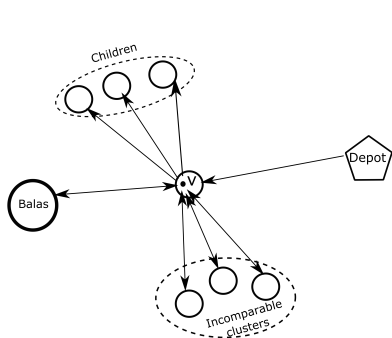


Figure 2.6: Cut $\delta(v)$ for Case 2

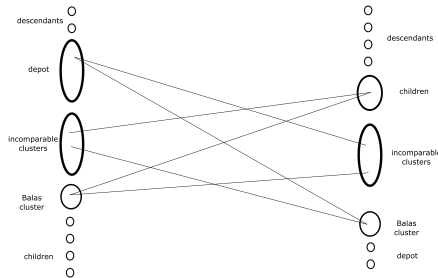


Figure 2.7: Bipartite graph \bar{B}_v^* for Case 2

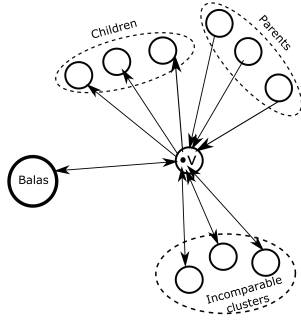


Figure 2.8: Cut $\delta(v)$ for Case 3

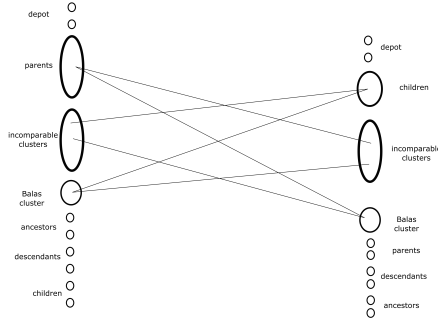


Figure 2.9: Bipartite graph \bar{B}_v^* for Case 3

and construct a feasible tour as follows. Departing from v_1 the tour visits all the clusters preceding $C(v)$ such that the cluster $C(i)$ is visited last, at node i . Then, we traverse arcs (i, v) and (v, j) directly, visit all the remaining clusters (respecting the precedence constraints) and complete the tour by returning to node v_1 .

Finally, $\text{rank}(\bar{B}_v^*) = N_{\bar{B}_v^*} - \kappa(\bar{B}_v^*) = |C_1| + |\tilde{\pi}| + 2|r| + 1 = |\delta(v)| - 1$.

Case 2 ($\tilde{\sigma} \neq \emptyset$ and $\tilde{\pi} = \emptyset$). This case is dual to Case 1, here $C(v)$ is the maximal ancestor in the partial order. In the similar sense (see Fig. 2.6 and Fig. 2.7), we obtain $|\delta(v)| = |C_1| + |\tilde{\sigma}| + 2|r| + 2$, $N_{\bar{B}_v^*} = 2|C_1| + 2|\hat{\sigma}| + 2|\tilde{\sigma}| + 2|r| + 2$, $\kappa(\bar{B}_v^*) = 1 + |C_1| + 2|\hat{\sigma}| + |\tilde{\sigma}|$, and finally, $\text{rank}(\bar{B}_v^*) = |C_1| + |\tilde{\sigma}| + 2|r| + 1 = |\delta(v)| - 1$.

Case 3 ($\tilde{\pi} \neq \emptyset$, $\tilde{\sigma} \neq \emptyset$). The only difference here is that cluster $C(v)$ has both parents and children, which slightly impacts the structure of \bar{B}_v^* . Proceeding with the proof in a similar way, we obtain (Fig. 2.8 and Fig. 2.9): $|\delta(v)| = |\tilde{\pi}| + |\tilde{\sigma}| + 2|r| + 2$, $N_{\bar{B}_v^*} = 2(|C_1| + |\hat{\sigma}| + |\tilde{\sigma}| + |\hat{\pi}| + |\tilde{\pi}| + |r| + 1)$, $\kappa(\bar{B}_v^*) = 1 + 2|C_1| + |\tilde{\sigma}| + |\tilde{\pi}| + 2|\hat{\pi}| + 2|\hat{\sigma}|$, and $\text{rank}(\bar{B}_v^*) = 2|\tilde{\pi}| + 2|\tilde{\sigma}| + 2|r| + 1 = |\delta(v)| - 1$.

Case 4 ($\tilde{\pi} = \emptyset$, $\tilde{\sigma} = \emptyset$). Without loss of generality, we restrict ourselves to the case where the set of free clusters is exhausted by $C(v)$ and C_{Balas} (if the set of free clusters has more than two elements, the case is similar to this one). Since this case is different from the discussed

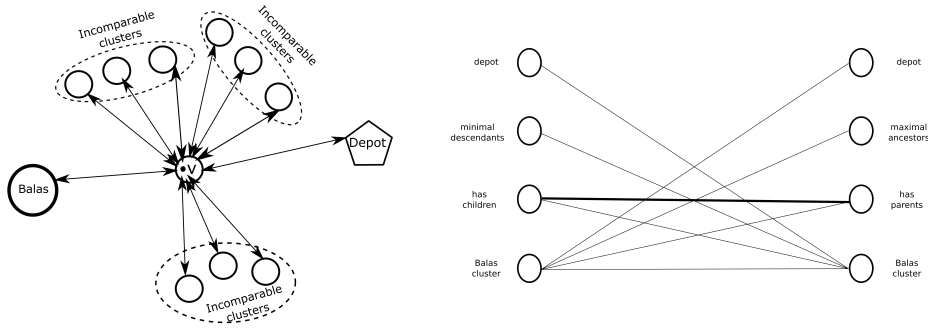


Figure 2.10: Representation of $\delta(v)$ and the connected components of \bar{B}_v^* for Case 4. Bold line provides connectivity of the \bar{B}_v^*

above, we provide an argument in detail. Since v is a non-individual node, $C(v) \neq C_{Balas}$. Again for cut $\delta(v)$, we have $|\delta(v)| = 2|C_1| + 2|r| + 2$. We show that in this case \bar{B}_v^* is a connected bipartite graph. We skip the trivial option of the empty order, since here \bar{B}_v^* is a complete graph. Otherwise, there are always at least two clusters C_p and C_q , such that C_p is the parent of C_q . Obviously, these clusters induce a complete bipartite subgraph of graph \bar{B}_v^* . Since both copies of C_{Balas} are incident with all other clusters from the opposite part, \bar{B}_v^* is connected (see Fig. 2.10). Finally, we obtain $N_{\bar{B}_v^*} = 2|C_1| + 2|r| + 2$, $\kappa(\bar{B}_v^*) = 1$, and $\text{rank}(\bar{B}_v^*) = 2|C_1| + 2|r| + 1 = |\delta(v)| - 1$.

Case 5 ($C(v) = C_1$). This is another unique case. To proceed with our proof, we need additional notation. By Σ , we denote the set of all nodes from minimal descendants, Π consists of all nodes from maximal

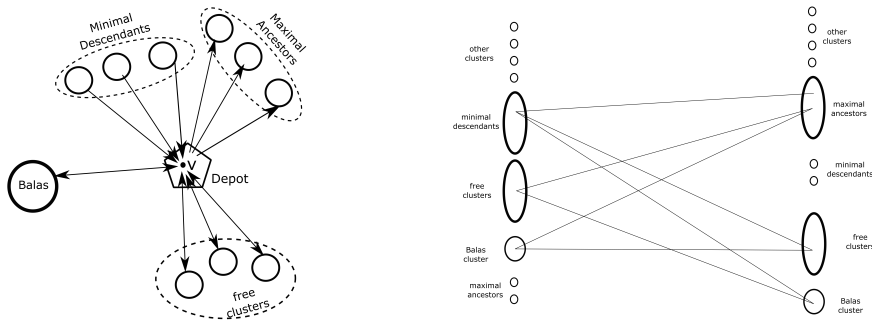


Figure 2.11: Representation of $\delta(v)$ and the connected components of the \bar{B}_v^* for Case 5

ancestors in the given partial order. Also, let F be the set of all nodes from free clusters, except C_{Balas} , and R are the remaining nodes. Then, $|\delta(v)| = 2|F| + |\Pi| + |\Sigma| + 2$.

As for the graph \bar{B}_v^* , it is constructed in the same sense as for the previous cases. The only difference here, is that the depot is departure and arrival node at the same time. However, this won't be a problem, since any feasible tour is closed (see Fig. 2.11). Finally, $N_{\bar{B}_v^*} = 2(|R| + |\Pi| + |F| + 1 + |\Sigma|)$, $\kappa(\bar{B}_v^*) = 1 + |\Pi| + |\Sigma| + 2|R|$, and $\text{rank}(\bar{B}_v^*) = |\delta(v)| - 1$. Lemma 2 is proved. \square

Now, we ready to establish dimension of the polytope $P^=$ and prove Theorem 1.

Proof. By construction, the PCGTS polytope $P^=$ is a part of a solution set of inequality system (3.2)-(3.7). Hence, $\dim P^=$ cannot be greater than dimension of this solution set. In turn, for an arbitrary feasible system of linear equations $Ax = b$ with $m \times d$ coefficient matrix, dimension of its solution set is $d - \text{rank}(A)$.

Let A be the coefficient matrix of system (3.2)-(3.7) (Fig. 2.12). By

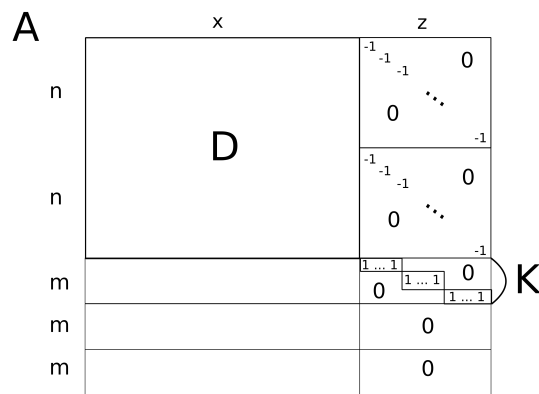


Figure 2.12: Structure of matrix A .

construction, $\text{rank}(A) \geq \text{rank}(D) + \text{rank}(K) = \text{rank}(D) + m$. We show that $\text{rank}(D) = 2n - 1$. To the initial graph G , we assign cluster digraph

$G_c = (\mathfrak{C}, E_c)$, for which $(C', C'') \in E_c$ if and only if there exist $i \in C'$ and $j \in C''$, such that $(i, j) \in E$. Let B_G and B_{G_c} be bipartite representations of digraphs G and G_c respectively.

Observation 3. *Evidently, if $(C', C'') \in E_c$, then $(i, j) \in E \forall i \in C' \forall j \in C''$.*

Observation 4. *By construction, D is the incidence matrix of B_G .*

As a simple consequence, we obtain that graphs B_G and B_{G_c} have the same number of connected components.

Since the initial graph G has at least one free cluster, then by Proposition 5.3 from [Balas et al., 1995], B_{G_c} is connected, i.e. $\kappa(B_{G_c}) = \kappa(B_G) = 1$, $\text{rank}(D) = 2n - 1$, and $\text{rank}(A) \geq 2n + m - 1$. Therefore,

$$\dim P^= \leq |E| + |V| - \text{rank}(A) \leq |E| + n - 2n - m + 1 = |E| - n - m + 1. \quad (2.43)$$

To complete the argument, we need to prove the lower bound

$$\dim P^= \geq |E| - n - m + 1. \quad (2.44)$$

We proceed with induction on the number of excessive nodes within clusters:

$$\rho = \sum_{h=1}^m (|C_h| - 1) = n - m$$

Base Case ($\rho = 0$) follows from Theorem 2 for the PCATSP.

Inductive Step. Assume that inequality (2.44) holds for some ρ . To prove it for $\rho + 1$, take an arbitrary non-individual node v . By Lemma 1 and Lemma 2,

$$\dim P^= \geq \dim P_v^= + \text{rank}(\bar{B}_v^*) = \dim P_v^= + |\delta(v)| - 1.$$

Recall that $P_v^=$ corresponds to the graph of $n - 1$ nodes and $|E \setminus \delta(v)|$ arcs. By induction hypothesis, $\dim P_v^= \geq |E| - |\delta(v)| - n - m + 2$, and the claim follows.

Combination of (2.43) and (2.44) concludes the proof. \square

2.3.2 Facet-inducing inequalities

By extending the results obtained in [Balas et al., 1995], in this subsection we establish the sufficient conditions ensuring that π - and σ -inequalities ((3.14) and (3.15)) introduced in Subsection 2.2.1 are facet-inducing.

Theorem 3. *For $\mathfrak{S} \subset \mathfrak{C} \setminus \{C_1, C_{Balas}\}$ and $\bar{\mathfrak{S}} = \mathcal{C} \setminus \mathfrak{S}$, an inequality*

$$x(\mathfrak{S} \setminus \pi(\mathfrak{S}), \bar{\mathfrak{S}}) \geq 1 \quad (2.45)$$

induces a facet of the polytope $P^=$, if $\pi(\mathfrak{S}) \subset \mathfrak{S}$, $\sigma(\mathfrak{S}) \subset \mathfrak{S}$, and \mathfrak{S} contains at least 3 free clusters.

Theorem 4. *For $\mathfrak{S} \subset \mathfrak{C} \setminus \{C_1, C_{Balas}\}$ and $\bar{\mathfrak{S}} = \mathcal{C} \setminus \mathfrak{S}$, an inequality*

$$x(\bar{\mathfrak{S}}, \mathfrak{S} \setminus \sigma(\mathfrak{S})) \geq 1 \quad (2.46)$$

induces a facet of the polytope $P^=$, if $\pi(\mathfrak{S}) \subset \mathfrak{S}$, $\sigma(\mathfrak{S}) \subset \mathfrak{S}$, and \mathfrak{S} contains at least 3 free clusters.

Similarly to Theorem 1, our proof is based on the inductive framework developed in [Fischetti et al., 1995] for the symmetric GTSP. The induction is carried out on the number of excessive nodes (2.42) in clusters. Since the base case corresponds to the classic PCATSP, our claim follows from the known result (Balas et al. [1995], Theorem 5.5). In turn, proof of the inductive step relies on Lemma 1 and our adaptation of Lemma 2 to the

case of the proper face $\mathcal{H}^\pi = \mathcal{H}(\alpha, \beta, \gamma)$, where

$$\beta = 0, \gamma = 1, \quad \alpha_{i,j} = \begin{cases} 1, & \exists C_p \in \mathfrak{S} \setminus \pi(\mathfrak{S}), \exists C_q \in \bar{\mathfrak{S}}: i \in C_p, j \in C_q, \\ 0, & \text{otherwise} \end{cases}$$

induced by inequality (2.45).

Lemma 3. *Let \mathcal{H}^π be the face of P^\equiv induced by π -inequality (2.45). The hypothesis of Theorem 3 implies that, for an arbitrary non-individual node v , $\text{rank}(B_v^*) = |\delta(v)| - 1$.*

Proof. Our argument is based on enumeration of all the possible options to establish a relation between cluster $C(v)$ and the given partial order. Previously, in the proof of Lemma 2, for each case, we explored properties of the associated cut $\delta(v)$ and bipartite graph \bar{B}_v^* . Now, each of these options can be split into several sub-options in correspondence to the ways to locate $C(v)$ with respect to the face \mathcal{H}^π (see Table 2.1).

Table 2.1: Options for cluster $C(v)$.

case #	relation to the partial order	relation to the face \mathcal{H}_π
1	minimal descendant	$\mathfrak{S}' = \mathfrak{S} \setminus \pi(\mathfrak{S}), \bar{\mathfrak{S}}$
2	maximal ancestor	$\pi(\mathfrak{S}), \bar{\mathfrak{S}}$
3	has parents and children	$\pi(\mathfrak{S}), \bar{\mathfrak{S}}$
4	free cluster	$\mathfrak{S}', \bar{\mathfrak{S}}$
5	depot	$\bar{\mathfrak{S}}$

It is easy to verify that all subcases of any unique case presented at a single line of Table 2.1 share the same cut $\delta(v)$, while their associated bipartite graphs B_v^* are spanning subgraphs of graph \bar{B}_v^* constructed in Lemma 2 for the entire polytope P^\equiv . In its proof, we showed that, for any v , graph \bar{B}_v^* contains a single connected component. Therefore, to prove Lemma 3, it is sufficient to show that the same node subset induces a connected component in any mentioned graph B_v^* as well.

For the sake of brevity, we restrict ourselves to cases 3 and 4 (see Table 2.1), since they appear to be the most common. For the other cases, the argument can be obtained in a similar way.

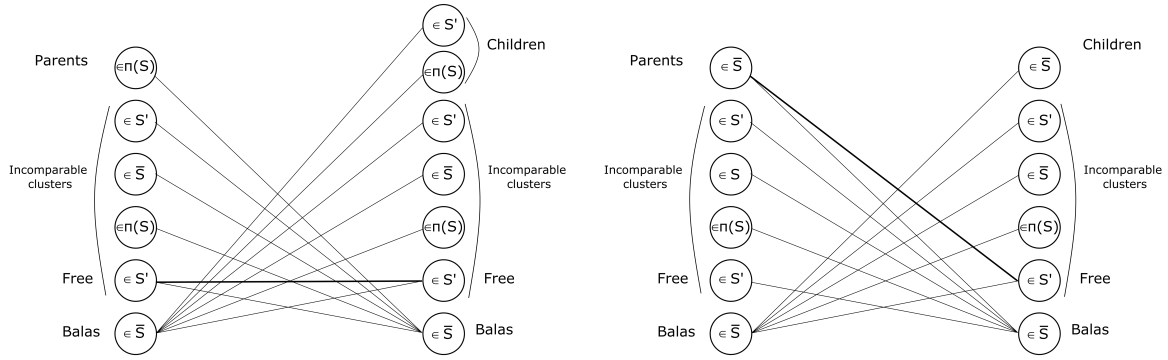


Figure 2.13: Connected components of B_v^* for Case 3. $C(v) \in \bar{\mathfrak{S}}$ (left) and $C(v) \in \bar{\mathfrak{S}}'$ (right)

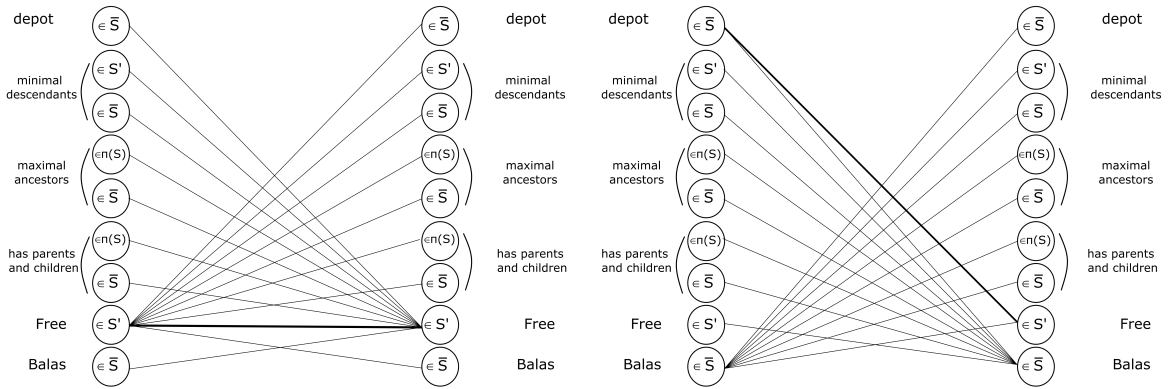


Figure 2.14: Connected components of B_v^* for Case 4. $C(v) \in \bar{\mathfrak{S}}'$ (left) and $C(v) \in \bar{\mathfrak{S}}'$ (right)

As in the proof of Lemma 2, by $\tilde{\pi}$, $\tilde{\sigma}$, and r , we denote the subsets of nodes (in graph G) belonging to parent, child and incomparable clusters with respect to cluster $C(v)$, respectively.

Case 3 ($\tilde{\pi} \neq \emptyset$, $\tilde{\sigma} \neq \emptyset$). In both subcases, for $C(v) \in \pi(\bar{\mathfrak{S}})$ and $C(v) \in \bar{\mathfrak{S}}$, we verify the connectivity of the subgraphs induced by the connected component found in the proof of Lemma 2, Case 3. We present

these subgraphs in Fig. 2.13 in more detail. To prove their connectivity, it is sufficient (a) to show that the single node from C_{Balas} is adjacent to any other node from the opposite part of graph B_v^* ; (b) to present at least one additional arc connecting nodes from any two clusters other than C_{Balas} .

(a) For instance, we establish the existence of an arc connecting node $i \in C_{Balas}$ and some node j belonging to some child cluster $C \in \mathfrak{S}' = \mathfrak{S} \setminus \pi(\mathfrak{S})$ of cluster $C(v)$ (Fig. 2.13). Departing from the depot, we start with construction of a tour T by visiting all the clusters in \bar{S} except C_{Balas} (regarding the precedence constraints). Then, we proceed with all the ancestors of cluster $C(j)$ except $C(v)$. This is possible due to Proposition 2.

Further, we traverse the i - v - j fragment and proceed with visiting all the remaining clusters in $\pi(\mathfrak{S})$. Finally, we randomly visit all the clusters in \mathfrak{S}' and return to the depot by a direct arc. By construction, it is the only arc in the proposed tour that belongs to the cut $\delta^+(\mathfrak{S} \setminus \pi(\mathfrak{S}), \bar{\mathfrak{S}})$ (in graph G). Therefore, for this tour, inequality (2.45) becomes tight.

(b) Without loss of generality, provide an argument for subcase $C(v) \in \bar{\mathfrak{S}}$ (Fig. 2.13). Let i be any node from some parent $C(i)$ of $C(v)$, and j belongs to a free cluster $C(j)$. Again, we consider the tour T departing from an arbitrary depot node. We visit all the ancestors of $C(v)$, except $C(i)$. Next, we pass through the i - v - j fragment and continue from $C(j)$ by visiting all the clusters in $\pi(\mathfrak{S})$. Then, we proceed with traveling over the rest of \mathfrak{S}' . Finally, we return to $\bar{\mathfrak{S}}$ by an arc that belongs to the cut $\delta^+(\mathfrak{S} \setminus \pi(\mathfrak{S}), \bar{\mathfrak{S}})$, and complete the tour by visiting the remaining clusters, arriving at the depot.

Case 4 ($\tilde{\pi} = \emptyset, \tilde{\sigma} = \emptyset$). Generally speaking, the argument for this case is close to the previous one. However, we mention it separately, since this case appears to be the only reason for requiring at least three free clusters from \mathfrak{S}' in the hypothesis of Theorem 3. As it follows from Fig.

2.14, for $C(v) \in \mathfrak{S}'$, cluster C_{Balas} does no longer induce a dominating set in the considered subgraph (of graph B_v^*). Instead, free clusters take its place.

Furthermore, these free clusters ensure the connectivity of the subgraph. Indeed, consider free clusters $C(i), C(j) \in \mathfrak{S}'$, such that $C(i) \neq C(v) \neq C(j)$. Construct a feasible tour T with the fragment $i-v-j$ in graph G . Since $C(i), C(v)$ and $C(j)$ are free and belong to \mathfrak{S}' , we are allowed to move $i-v-j$ directly after the departure from the depot. Then, after visiting all the clusters in $\pi(\mathfrak{S})$, we come to the remaining clusters from \mathfrak{S}' , cross the border between \mathfrak{S}' and $\bar{\mathfrak{S}}$ (at once), move through all the clusters in $\bar{\mathfrak{S}}$ and return to the depot.

In subcase $C(v) \in \bar{\mathfrak{S}}$ (Fig. 2.14), the proof can be obtained in a similar way to the Case 3. Lemma 3 is proved. \square

Now, we are ready to establish the proof of Theorem 3.

Proof. Let \mathcal{H}^π be the face of polytope P^π induced by π -inequality. By Theorem 1, we have $\dim \mathcal{H}^\pi \leq \dim P^\pi = |E| - n - m + 1$. By induction on number ρ (see eqn. (2.42)), we show that

$$\dim \mathcal{H}^\pi \geq |E| - n - m. \quad (2.47)$$

Base case of ($\rho = 0$) is proved in [Balas et al., 1995], since, in this case, the problem at hand is equivalent to the PCATSP.

Inductive step. Assuming that (2.47) holds for some ρ , prove it for $\rho + 1$. Combining claims of Lemma 1 and Lemma 3, we have $\dim \mathcal{H}^\pi \geq \dim \mathcal{H}_v^\pi + \text{rank}(B_v^*) = \dim \mathcal{H}_v^\pi + |\delta(v)| - 1$. Since, by induction hypothesis, $\dim \mathcal{H}_v^\pi \geq |E| - |\delta(v)| - n + 1 - m$, we obtain the desired lower bound (2.47).

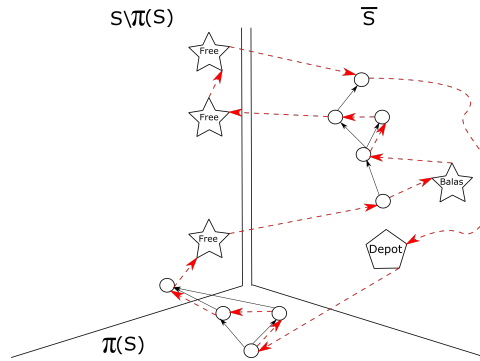


Figure 2.15: Example of the contradicting solution

To finalize the proof, we show that inequality (2.47) is tight. Indeed, suppose by contradiction that it is not. But, under this assumption, $\dim \mathcal{H}^\pi = \dim P^\pi$ and, consequently the face \mathcal{H}^π coincides with the polytope P^π . However, we can always provide a feasible solution crossing the outgoing cut $\delta^+(\mathfrak{S} \setminus \pi(\mathfrak{S}))$ at least twice (see, e.g., Fig. 2.15). Theorem 3 follows from the obtained contradiction. \square

For the sake of brevity, we omit the proof of Theorem 4, which can be obtained in a similar way.

2.4 Formulations

In this section we describe novel MILP-models (formulations) for the PCGTSP. Almost all of them are extensions of the known formulations proposed initially in [Gouveia and Pesneau, 2006, Gouveia and Ruthmair, 2015, Gouveia et al., 2018] for the PCATSP and incorporate exponential size families of valid inequalities introduced for the PCGTSP in Section 2.2.

Following to [Gouveia et al., 2018], we start with the sequence of models obtained incrementally as follows:

- M_1 is our basic compact model described in Subsection 3.2.1,
- M_2 is M_1 augmented with strengthened simple-cut (2.27)–(2.29) and

- both strengthened regular (2.30), and reversed GDDL (2.31) inequalities,
- M_3 is M_2 with strengthened 2-path inequalities (2.32)–(2.35),
 - M_4 is M_3 enforced by strengthened 3v GDDL-like inequalities (2.36)–(2.38),
 - M_5 is M_4 supplied with strengthened 4v GDDL-like inequalities (2.39).

In addition, we propose the model incorporating the inequalities described in Subsection 2.2.1, Subsection 2.2.2, and Subsection 2.2.3:

- M_1^* is M_1 augmented with π -, σ -, and (π, σ) -inequalities (3.14)–(2.19), precedence cycle breaking inequalities (3.16) and (3.17), and inequalities (3.18)–(3.22).

In order to increase the tightness of the lower bounds, we combine M_1^* with other best performers of our exploratory Experiment I (see Subsection 2.6.2):

- M_3^* , which is $M_1^* + M_3$ and
- $M_5^* = M_1^* + M_5$.

In all these models, families of valid inequalities are separated exactly, following to the incremental pattern proposed in [Gouveia et al., 2018]. Although the models M_3 and M_5 clearly benefit from the combination with M_1^* in terms of the lower bounds, they still remain to be rather time-consuming.

Therefore, by evolving the well-known *roulette-wheel* sampling principle [see ex. Gendreau and Potvin, 2019] and simple *online learning* technique, we propose a novel heuristic separation procedure and the corresponding models M_{3s}^* and M_{5s}^* , which we call *sampled* as well. The main idea of the proposed procedure is as follows:

- to each family of valid inequalities, we assign an appropriate probabilistic measure; for instance, in the case of 3v GDDL-like inequalities (2.36)–(2.38), it is sufficient to define a discrete distribution on the set of ordered quadruples (C_p, C_q, C_r, C_s) of non-depot clusters;

- given by a sample size, at each separation epoch, we apply cut generation technique at this epoch only to the entries of a sample drawn from the defined distribution;
- each time, when a tuple managed to produce a cut, we increase its probability.

Generally speaking, the proposed separation heuristic is a compromise between the tightness of the LP-relaxation bounds and numerical performance. However, we decide to evaluate it in our experiments along with the known incremental separation pattern, because the sampling gives us an opportunity to adopt powerful but large families of valid inequalities from the very beginning of the LP-relaxation solution process.

In addition, we introduce the formulation $M_{\text{MTZ-DL}}$, based on the PCGTSP adaptation of the Miller-Tucker-Zemlin model [Miller et al., 1960] lifted by Desrochers and Laporte [Desrochers and Laporte, 1991]. It can be obtained from the compact model proposed in Section 3.2.1 by exclusion y variables and replacing constraints (2.12)–(2.15) with

$$\begin{aligned}
 v_p - v_q + (m - 1)u_{pq} + (m - 3)u_{qp} &\leq m - 2 & (p, q \in \{2, \dots, m\}, p \neq q) \\
 -v_p + (m - 3)u_{p1} + \sum_{q=2}^m u_{qp} &\leq 0 & (p \in \{2, \dots, m\}) \\
 v_p + (m - 3)u_{1p} + \sum_{q=2}^m u_{pq} &\leq m - 2 & (p \in \{2, \dots, m\}) \\
 v_q - v_p &\geq 1 & (p, q \in \{2, \dots, m\}: C_p \prec C_q)
 \end{aligned}$$

for $u_{pq} \geq 0$, $v_p \geq 0$. We should note that variables v_p have the same meaning as the original free variables u_i introduced in [Miller et al., 1960]. For the sake of convenience, we renamed those variables in order to follow up with our notation.

We take this model intentionally for the subsequent performance comparison of the variants of the proposed branch-and-cut algorithm, because

it is one of the lightest known compact models ensuring efficient enumeration of the nodes in a branching tree. By construction, its complexity for the PCGTSP is $O(n^2 + m^2)$, while for M_1 we have $O(n^2 + m^3)$. Furthermore, the choice of this model is motivated by the theoretical and computational results of [Roberti and Toth, 2012] for the ATSP. Finally, we obtain the model $M_{\text{MTZ-DL}}^*$ in a similar way as M_1^* .

2.5 Branch-and-Cut Algorithm

Our branch-and-cut algorithm extends the algorithm proposed in [Gouveia and Ruthmair, 2015] for the SOP and has a component-wise structure based on few main building blocks. Among them are instance preprocessing routine, primal heuristic, and a formulation of the problem in question that specifies a family of cutting planes.

In its current version, the proposed algorithm is restricted to use the same instance preprocessing routine. The arcs violating precedence constraints are excluded from the given graph by preprocessing rules (2.1)-(2.5), previously introduced in [Balas et al., 1995] for the PCATSP. In addition, as the only primal heuristic, the algorithm uses PCGLNS, proposed in Khachay et al. [2020a] and briefly described in Subsection 2.5.1. Thus, all the proposed variants of the algorithm (refer to Subsection 2.6.3 for details) were obtained by varying the problem formulation.

2.5.1 PCGLNS Primal Heuristic

The PCGLNS heuristic extends the recent GLNS algorithm proposed in [Smith and Imeson, 2017] for the common GTSP. PCGLNS is designed to take into account additional precedence constraints defined on a set of clusters. In a nutshell, PCGLNS appears to be an original implementation of the seminal Adaptive Large Neighborhood Search (ALNS) metaheuristic [see, e.g. Gendreau and Potvin, 2019] and combines the well-known *ruin*

and *recreate* principle with online learning over a given sets of basic removal and insertion local search heuristics.

2.5.2 Implementation

The proposed algorithm is implemented on top of the Gurobi 9.3 framework. Primal heuristic and cutting planes are provided as user callback functions. For the sampled models, all the parameters of the heuristic separation including sample sizes and learning rates are tuned within preliminary testing stage. All the built-in Gurobi heuristics and cutting plane algorithms are disabled, while other parameters of the solver keep their default values. The suggested implementation is carried out in Python 3 leveraging NetworkX software package for internal graph processing tasks and fully cross-platform. All source code together with the reported experimental results are open for public access at <https://github.com/EnsignDaniels/BnC>.

2.6 Numerical evaluation

In this section, we report results of the competitive numerical experiments that show how each proposed formulation and variant of the branch-and-cut algorithm could be useful for the PCGTSP. In particular, these results reveal the notable impact contributed by predecessor/successor inequalities in terms of accuracy and running time, which can be considered as an additional support of the theoretical results obtained in Section 2.3. We proceed with two separate experiments. In the former one, we evaluate the proposed formulations with respect to their LP-relaxation bounds and the time consumption. In turn, the purpose of the latter one is to compare the best performers of the first experiment with known results within the branch-and-cut setting. All the computations are carried out

Table 2.2: PCGTSPLIB library.

instance	n	m	PC density	instance	n	m	PC density
ESC07	39	8	14	p43.1	203	43	53
ESC12	65	13	23	p43.2	198	43	76
ESC25	133	26	36	p43.3	211	43	138
ESC47	244	48	79	p43.4	204	43	538
ESC63	349	64	296	prob.100	510	99	139
ESC78	414	79	361	prob.42	208	41	59
br17.10	88	17	31	rbg048a	255	49	495
br17.12	92	17	38	rbg050c	259	51	558
ft53.1	281	53	64	rbg109a	573	110	5438
ft53.2	274	53	82	rbg150a	871	151	10484
ft53.3	281	53	269	rbg174a	962	175	14129
ft53.4	275	53	811	rbg253a	1389	254	30434
ft70.1	346	70	86	rbg323a	1825	324	48525
ft70.2	351	70	117	rbg341a	1822	342	56644
ft70.3	347	70	284	rbg358a	1967	359	56894
ft70.4	353	70	1394	rbg378a	1973	379	63963
kro124p.1	514	100	132	ry48p.1	256	48	59
kro124p.2	524	100	169	ry48p.2	250	48	73
kro124p.3	534	100	365	ry48p.3	254	48	179
kro124p.4	526	100	2404	ry48p.4	249	48	643

n and m are the number of nodes and clusters respectively

'PC density' is the number of arcs in the transitively closed precedence DAG

on the 16-core Intel Xeon 128G RAM server ² against the same public benchmark library PCGTSPLIB.

2.6.1 PCGTSPLIB Benchmark library

The PCGTSPLIB library was derived in [Salman et al., 2020] from the well-known SOPLIB library in order to provide a test-bed for PCGTSP. To the best of our knowledge, it is the only public library for the problem in question. We provide a short overview of this library in Table 2.2.

Since computational complexity of the PCGTSP depends mostly on the number of clusters m (rather than the size of a node set n , as it is for SOP), it is convenient to partition all 40 instances of this library into small (up to 30 clusters), medium (up to 70 clusters), large (up to 120 clusters), and huge ones (more than 120 clusters). In addition, the instances differ substantially in terms the density of the constituent partial orders.

For each instance, we round the transportation costs to the nearest integral values. For the sake of convenience, we provide the converted instances along with our source codes (<https://github.com/EnsignDaniels/BnC>).

²provided by Supercomputer 'Uran' at N.N. Krasovskii Institute of Mathematics and Mechanics

2.6.2 Experiment I: Comparison of the LP-relaxations

Inspired by the results of [Gouveia et al., 2018], we start with the comparison of the formulations M_1 – M_5 and M_1^* in terms of their LP-relaxation bounds and time complexity. In this experiment, for each competing model, computation time was limited to 10 hours (36000 seconds).

Table 2.3: Comparison of formulations M_1 – M_5 and M_1^* .

Instance	OPT	M_1		M_2		M_3		M_4		M_5		M_1^*	
		LPB	t	LPB	t	LPB	t	LPB	t	LPB	t	LPB	t
ESC07	1730	1683	0	1730	0.28	1730	0.29	1730	0.29	1730	0.3	1730	0.09
ESC12	1390	1238	0.02	1387	5.91	1387	7.1	1387	8.21	1387	9.97	1390	0.8
ESC25	1383	1296	0.21	1362	229	1362	300	1362	364	1362	448	1363	7
ESC47	1063	1001	8.46	1012	2982	1012	3545	1013	4655	1016	7420	1023	119
ESC63	62	62	207.36	–	–	–	–	–	–	–	–	<u>62</u>	318
ESC78	[14672, 14808]	14629	3829.32	14640	23287	14641	36000	14641	36000	14641	36000	14659	5477
br17.10	43	15	0.05	43	6.81	–	–	–	–	–	–	<u>32</u>	5
br17.12	43	15	0.05	43	6.54	–	–	–	–	–	–	<u>35</u>	6
ft53.1	6194	4981	9.71	5780	7045	5781	9646	5781	10540	5781	12688	5833	400
ft53.2	[6571, 6619]	5079	19.45	5951	4381	5960	8084	5961	10924	5962	17774	5982	174
ft53.3	[8360, 8446]	5928	114.76	7168	5761	7169	8637	7169	11541	7169	13080	7178	200
ft53.4	11822	9850	2.6	11443	4828	11449	5964	11449	9858	11449	14692	11437	49
ft70.1	32608	31543	228.15	32258	19022	32258	36000	32258	36000	32258	36000	32348	3069
ft70.2	[33008, 33448]	31820	395.64	32556	19258	32556	23596	32556	24649	32556	36000	32561	673
ft70.3	[34807, 35234]	32842	712.56	33960	9728	33961	25165	33961	36000	33961	36000	33856	1764
ft70.4	44436	40068	115.96	41080	9196	42116	23481	42116	36000	42116	36000	42043	182
kro124p.1	[31787, 32825]	29337	3589.11	29647	36000	29647	36000	29647	36000	29647	36000	30663	36000
kro124p.2	[32379, 34253]	29544	3036	29544	36000	29923	36000	29923	36000	29923	36000	30259	9791.41
kro124p.3	[35110, 40906]	30424	17364.51	30424	36000	30425	36000	30425	36000	30425	36000	31840	21149
kro124p.4	[56151, 62818]	43495	2310.91	43495	36000	47023	36000	47023	36000	47023	36000	49019	4776
p43.1	22545	879	3.19	22414	1702	22414	2175	22414	2997	22414	4363	22545	308
p43.2	22837	985	5.17	22651	1858	22651	2465	22651	3525	22651	4350	22645	409
p43.3	23119	1076	3.38	22802	1956	22802	2532	22802	3689	22802	5636	22848	400
p43.4	66848	44854	1.56	53858	1622	53858	2648	53858	3844	66678	4951	56071	73
prob.100	[830, 1343]	803	428.58	815	36000	816	36000	816	36000	816	36000	822	3457.78
prob.42	202	183	5.33	190	1429	191	1632	192	2718	193	3040	188	201
rbg048a	282	273	4.8	282	2901	–	–	–	–	–	–	282	61
rbg050c	378	376	7.14	377	3311	377	5763	377	12218	378	14430	378	38
rbg109a	848	803	1.87	803	36000	832	36000	832	36000	832	36000	840	427
rbg150a	1414	1381	7.44	1381	36000	1381	36000	1381	36000	1381	36000	1411	1519
rbg174a	1641	1606	8.19	1606	36000	1606	36000	1606	36000	1606	36000	1631	2512
rbg253a	2372	2308	20.05	2308	36000	2308	36000	2308	36000	2308	36000	2342	850
rbg341a	[2062, 2147]	1961	82.02	1961	36000	1961	36000	1961	36000	1961	36000	2019	1604.05
rbg358a	[2037, 2172]	1967	7028	1967	36000	1967	36000	1967	36000	1967	36000	2001	36000
rbg378a	[2236, 2385]	2132	35422	2132	36000	2132	36000	2132	36000	2132	36000	2166	36000
ry48p.1	[13109, 13135]	11617	22.54	11952	3413	11966	5311	11984	6772	11988	10910	12158	440
ry48p.2	[13401, 13802]	11721	12.24	12188	3529	12216	4375	12216	6812	12216	8171	12357	379
ry48p.3	[15778, 16533]	12520	112.79	13873	1749	13879	4392	13879	4717	13879	5888	13937	235
ry48p.4	25977	20378	3.46	21888	1844.32	21888	2081	22670	6564	23049	8669	22861	33

Notes: column ‘OPT’ provides optimum values of the instances (if known) or the best bounds; columns M_1 – M_5 and M_1^* present LP-relaxation lower bounds and the corresponding running times; optimum values highlighted in bold, best lower bounds underlined

Since the separation procedures for M_2 – M_5 follow the incremental pattern initially proposed for the SOP, more complex formulations provide tighter lower bounds, perhaps with substantially increased computation time. Therefore, for each instance, whose optimum value is achieved by some M_i model, we do not solve it by M_j , for any $j > i$. As it follows from Table 2.3, the optimum values were found for 8 out of 40 instances: for ESC63 - by model M_1 , for br17.10 and br17.12 - by model M_2 , for

other five - by model M_1^* (along with M_2 and M_5 for `rbg048a` and `rbg050c`, respectively). For the remaining instances, M_2 found the tightest lower bound once, M_3 three times as well as M_5 , and M_1^* — 25 times.

Table 2.4: Performance of the combined and sampled formulations.

Instance	OPT	M_1^*		M_3		M_5		M_3^*		M_5^*		M_{3s}^*		M_{5s}^*	
		LPB	t	LPB	t	LPB	t	LPB	t	LPB	t	LPB	t	LPB	t
ESC07	1730	1730	0.09	1730	0.29	1730	0.3	1730	0.29	1730	0.3	1730	0.04	1730	0.06
ESC12	1390	1390	0.8	1387	7.1	1387	9.97	1387	7.2	1387	10	1390	0.35	1390	0.46
ESC25	1383	1363	7	1362	300	1362	448	1362	205	1362	221	1363	13	1383	4.31
ESC47	1063	1023	119	1012	2982	1016	7420	1014	5899	1018	6658	1026	247	1030	589
ESC63	62	62	318	62	8491	62	36000	62	13790	62	16506	62	250	62	243
ESC78	[14672, 14808]	14659	5477	14640	23287	14641	36000	14660	36000	14660	36000	14660	6106	14660	5312
br17.10	43	32	5	43	7	43	12.63	43	6.87	43	8.1	35	6	33	5
br17.12	43	35	6	43	7	43	35.05	43	6.69	43	7.99	34	5	34	4
ft53.1	6194	5833	400	5781	9646	5781	12688	5803	4786	5803	6144	5803	910	5833	375
ft53.2	[6571, 6619]	5982	174	5960	8084	5962	17774	6035	7282	6035	10144	5981	237	5982	124
ft53.3	[8360, 8446]	7178	200	7169	8637	7169	13080	7169	7717	7171	10286	7176	300	7180	204
ft53.4	11822	11437	49	11449	5964	11449	14692	11450	1450	11457	9649	11498	62	11463	43
ft70.1	32608	32348	3069	32258	36000	32258	36000	32380	13700	32380	36000	32385	4484	32348	2191
ft70.2	[33008, 33448]	32561	673	32556	23596	32556	36000	32563	24445	32564	28444	32598	862	32559	586
ft70.3	[34807, 35234]	33856	1764	33961	25165	33961	36000	33990	21890	33995	29579	33855	1275	33852	1432
ft70.4	44436	42043	182	42116	23481	42116	36000	42120	11022	42120	12040	42019	287	41900	198
kro124p.1	[31787, 32825]	30663	36000	29647	36000	29647	36000	29647	36000	29647	36000	30174	36000	30563	36000
kro124p.2	[32379, 34253]	30259	9791.41	29923	36000	29923	36000	30082	36000	30082	36000	30274	36000	30326	5527
kro124p.3	[35110, 40906]	31840	21149	30425	36000	30425	36000	30425	36000	30425	36000	31780	21225	31871	36000
kro124p.4	[56151, 62818]	49019	4776	47023	36000	47023	36000	47023	36000	47023	36000	48660	2579	48560	2513
p43.1	22545	22545	308	22414	2175	22414	4363	22415	1774	22415	3816	22545	206	22545	167
p43.2	22837	22645	409	22651	2465	22651	4350	22653	2316	22655	3649	22650	445	22642	338
p43.3	23119	22848	400	22802	2532	22802	5636	22870	1272	22872	2361	22915	389	22870	325
p43.4	66848	56071	73	53858	2648	66678	4951	53859	1328	66679	3400	56053	80	66700	340
prob.100	[830, 1343]	822	3457.78	816	36000	816	36000	816	36000	816	36000	823	4736	822	3596
prob.42	202	188	201	191	1632	193	3040	192	1325	193	2043	190	322	188	196
rbg048a	282	282	61	282	3544	282	15951	282	2128	282	21449	282	38	282	46
rbg050c	378	378	38	377	5763	378	14430	378	2880	378	12682	378	29	378	65
rbg109a	848	840	427	832	36000	832	36000	832	36000	832	36000	848	3530	840	952
rbg150a	1414	1411	1519	1381	36000	1381	36000	1381	36000	1381	36000	1411	17762	1411	2280
rbg174a	1641	1631	2512	1606	36000	1606	36000	1606	36000	1606	36000	1635	36000	1632	2455
rbg253a	2372	2342	850	2308	36000	2308	36000	2308	36000	2308	36000	2342	9059	2342	2135
rbg323a	2533	2515	3654.61	2491	36000	2491	36000	2491	36000	2491	36000	2517	5580	2515	7737
rbg341a	[2062, 2147]	2019	1604.05	1961	36000	1961	36000	1961	36000	1961	36000	2017	4190	2021	3533
rbg358a	[2037, 2172]	2001	36000	1967	36000	1967	36000	1967	36000	1967	36000	2013	36000	2013	36000
rbg378a	[2236, 2385]	2166	36000	2132	36000	2132	36000	2132	36000	2132	36000	2189	36000	2191	36000
ry48p.1	[13109, 13135]	12158	440	11966	5311	11988	10910	12052	3539	12053	8878	12458	926	12167	252
ry48p.2	[13401, 13802]	12357	379	12216	4375	12216	8171	12217	5176	12217	8459	12780	1436	12366	253
ry48p.3	[15778, 16533]	13937	235	13879	4392	13879	5888	14011	5084	14597	13794	13783	179	13840	172
ry48p.4	25977	22861	33	21888	2081	23049	8669	22781	1874	23050	4988	22674	34	22677	46

Notes: column ‘OPT’ provides optimum values of the instances (if known) or the best bounds; optimum values highlighted in bold

Although the model M_1^* appears to be the best performer for the most cases, there exist instances, e.g. `ft53.4`, `ft70.4`, and `ry48.p4`, where some other competitors found more tight lower bounds. Therefore, we evaluate models M_3^* and M_5^* obtained by combination M_1^* with M_3 and M_1^* with M_5 , where M_5 and M_3 are chosen for the combination as the most powerful and well-balanced³ models among M_2 – M_5 respectively.

According to results presented in Table 2.4, formulations M_3 , M_5 and M_1^* collaborate quite well. In particular, for instances `ft53.2`, `ft70.1` and `p43.2`, M_3^* provides better lower bounds than both initial models M_3 and

³with respect to accuracy and time consumption

M_1^* . The similar result can be observed for instances ry48p.3 and ft53.4 with respect to formulations M_5 , M_1^* and their combination M_5^* .

While the combined models perform better than their initial counterparts, they still remain to be quite expensive to be applied in the branch-and-cut algorithm. On the other hand, comparing the model M_3^* with the sampled one M_{3s}^* and excluding tiny instances ESC07, ESC12, br17.10 and br17.12, we observe the significant decrease of the time complexity, i.e. LP-relaxation was solved 16 times faster in average. Furthermore, the better lower bounds were obtained in 18 out of 36 remaining instances. For those instances where M_{3s}^* found less accurate results, the lower bound decreased at most by 1.7%. In addition, we should emphasize one large instance rbg109a, where M_{3s}^* found an optimum value of the LP-relaxation faster than all other competitors.

As for the models M_5^* and M_{5s}^* , we observe average speed-up by 59 times and better lower bounds in 22 out the same 36 instances. For that instances, where M_5^* outperform its sampled counterpart, the lower bound decreased at most by 5.2%. In addition, we should emphasize the instance ESC25, for which M_{5s}^* was the only competitor, who found the optimum value.

To summarize, we conclude that the addition of predecessor/successor inequalities and application of the proposed heuristic separation procedure can provide significant improvement in LP-relaxation of the PCGTSP.

2.6.3 Experiment II: Comparison of Branch-and-Cut Algorithms

This experiment is intended to assess variants of the branch-and-cut algorithm proposed in Section 2.5 induced by several formulations introduced in Section 2.4.

For the first competition, we choose variants bc_1^* , bc_{3s}^* , and bc_{5s}^* induced by the best performers of Experiment I, the models M_1^* , M_{3s}^* , and

Table 2.5: Comparison of the branch-and-cut algorithms (a)

Instance	OPT		Gurobi	bc_{DFJ}^*	bc_1^*	bc_{3s}^*	bc_{5s}^*	bc_{MTZ-DL}^*	bc_{3s}	bc_{MTZ-DL}
ESC07	1730	UB	1730	1730	1730	1730	1730	1730	1730	1730
		LB	1730	1730	1730	1730	1730	1730	1730	1730
		gap	0	0	0	0	0	0	0	0
		t	0.05	0.06	0.09	0.04	0.06	0.05	0.06	0.09
ESC12	1390	UB	1390	1390	1390	1390	1390	1390	1390	1390
		LB	1390	1390	1390	1390	1390	1390	1390	1390
		gap	0	0	0	0	0	0	0	0
		t	5.05	0.58	0.8	0.35	0.46	0.15	2.14	0.51
ESC25	1383	UB	1383	1383	1383	1383	1383	1383	1383	1383
		LB	1383	1383	1383	1383	1383	1383	1383	1383
		gap	0	0	0	0	0	0	0	0
		t	7.56	1.98	8	15	4.31	2.88	8.55	3.46
ESC47	1063	UB	1063	1063	1063	1063	1063	1063	1063	1063
		LB	1063	1063	1063	1063	1063	1063	1063	1063
		gap	0	0	0	0	0	0	0	0
		t	6963	43.06	623	1520	837	31.62	1355	191
ESC63	62	UB	62	62	62	62	62	62	62	62
		LB	62	62	62	62	62	62	62	62
		gap	0	0	0	0	0	0	0	0
		t	209	7.52	318	250	243	3.24	244	5
ESC78	[14672, 14808]	UB	14808	14808	14808	14808	14808	14808	14808	14808
		LB	14633	14657	14661	14666	14667	14672	14553	14659
		gap	1.2	1	1	1	1	0.9	1.8	1
		t	72000	72000	72000	72000	72000	72000	72000	72000
br17.10	43	UB	43	43	43	43	43	43	43	43
		LB	43	43	43	43	43	43	43	43
		gap	0	0	0	0	0	0	0	0
		t	232	11.74	12	10	13	9.7	10.76	230
br17.12	43	UB	43	43	43	43	43	43	43	43
		LB	43	43	43	43	43	43	43	43
		gap	0	0	0	0	0	0	0	0
		t	75	58.09	11	12	12	9.9	12	101
ft53.1	6194	UB	6194	6194	6194	6194	6194	6194	6194	6194
		LB	5933	6194	6169	6176	6177	6194	6176	6051
		gap	4.4	0	0.4	0.3	0.3	0	0.3	2.4
		t	72000	3398	72000	72000	72000	1120	72000	72000
ft53.2	[6571, 6619]	UB	6619	6619	6619	6619	6619	6619	6619	6619
		LB	6087	6442	6410	6490	6439	6571	6487	6279
		gap	8.7	2.7	3.3	2	2.8	0.7	2	5.4
		t	72000	72000	72000	72000	72000	72000	72000	72000

Notes: best performers are highlighted in bold

M_{5s}^* respectively. In addition, we adopt variant bc_{MTZ-DL}^* induced by the formulation M_{MTZ-DL}^* . As baselines, we use Gurobi solver applied to the model M_1 with default configuration (including built-in heuristics and cutting planes) and our PCGTSP adaptation bc_{DFJ}^* of the state-of-the-art branch-and-cut algorithm for the SOP proposed in [Gouveia and Ruthmair, 2015]. This algorithm tackles the similar partial classic Dantzig-Fulkerson-Johnson (DFJ) model [Dantzig et al., 1954] ((3.1)–(3.7), and (3.12) without y variables) and separates corresponding families of valid

inequalities (3.14)–(3.22). In addition, we replace the initial primal heuristic with our GLNS-based heuristic PCGLNS, since GLNS appears to be more efficient for the GTSP-like problems [see Smith and Imeson, 2017].

Table 2.6: Comparison of the branch-and-cut algorithms (b)

Instance	OPT		Gurobi	bc _{DFJ} [*]	bc ₁ [*]	bc _{3s} [*]	bc _{5s} [*]	bc _{MTZ-DL} [*]	bc _{3s}	bc _{MTZ-DL}
ft53.3	[8360, 8446]	UB	8446	8446	8446	8446	8446	8446	8446	8446
		LB	7135	7786	7992	8186	8222	8360	8108	7525
		gap	18.4	8.5	5.7	3.2	2.7	1	4.2	12.2
		t	72000	72000	72000	72000	72000	72000	72000	72000
ft53.4	11822	UB	11822	11822	11822	11822	11822	11822	11822	11822
		LB	11253	11687	11822	11822	11822	11822	11822	11822
		gap	5.1	1.2	0	0	0	0	0	0
		t	72000	72000	129	163	192	91.62	733	39271
ft70.1	32608	UB	32614	32608	32614	32614	32614	32608	32614	32614
		LB	31765	32608	32466	32480	32455	32608	32475	32082
		gap	2.7	0	0.5	0.4	0.5	0	0.4	1.7
		t	72000	6523	72000	72000	72000	2385	72000	72000
ft70.2	[33008, 33448]	UB	33448	33448	33448	33448	33448	33448	33448	33448
		LB	32029	32889	32799	32890	32805	33008	32740	32850
		gap	4.4	1.7	2	1.7	2	1.3	2.2	1.8
		t	72000	72000	72000	72000	72000	72000	72000	72000
ft70.3	[34807, 35234]	UB	35234	35234	35234	35234	35234	35234	35234	35234
		LB	33232	34105	34304	34761	34719	34807	34629	34361
		gap	6	3.3	2.7	1.4	1.5	1.2	1.7	2.5
		t	72000	72000	72000	72000	72000	72000	72000	72000
ft70.4	44436	UB	44451	44451	44451	44451	44451	44436	44451	44451
		LB	41634	41388	44051	43998	44033	44436	43990	42345
		gap	6.8	7.4	0.9	1	0.9	0	1	5
		t	72000	72000	72000	72000	72000	59670	72000	72000
kro124p.1	[31787, 32825]	UB	32835	32835	32825	32835	32835	32835	32835	32835
		LB	29704	30858	30827	30174	30182	31787	29530	30555
		gap	10.5	6.4	6.5	8.8	8.8	3.3	11.2	7.5
		t	72000	72000	72000	72000	72000	72000	72000	72000
kro124p.2	[32379, 34253]	UB	34253	34253	34253	34253	34253	34253	34253	34253
		LB	30084	30722	30509	30448	30448	32379	29881	31657
		gap	13.9	11.5	12.3	12.5	12.5	5.8	14.6	8.2
		t	72000	72000	72000	72000	72000	72000	72000	72000
kro124p.3	[35110, 40906]	UB	40906	40906	40906	40906	40906	40906	40906	40906
		LB	30945	31930	32734	32954	32674	35110	31122	33918
		gap	32.2	28.1	25	24.1	25.2	16.5	31.4	20.6
		t	72000	72000	72000	72000	72000	72000	72000	72000
kro124p.4	[56151, 62818]	UB	62818	62818	62818	62818	62818	62818	62818	62818
		LB	46861	45720	54993	55329	53841	56151	51210	51495
		gap	34.1	37.4	14.2	13.5	16.7	11.9	22.7	22
		t	72000	72000	72000	72000	72000	72000	72000	72000

Notes: best performers are highlighted in bold

All the competitors are supplied with the same primal heuristic PCGLNS. The time limit is set to 20 hours (72000 seconds). We report cost of the best found solution (UB), the best lower bound (LB), an accuracy measure (gap, in percentage)

$$\text{gap} = \frac{\text{UB}-\text{LB}}{\text{LB}} \geq \frac{\text{UB}-\text{OPT}}{\text{OPT}} = \varepsilon,$$

for the relative error ε of the obtained solution, and the elapsed time (in seconds).

Table 2.7: Comparison of the branch-and-cut algorithms (c)

Instance	OPT		Gurobi	bc_{DFJ}^*	bc_1^*	bc_{3s}^*	bc_{5s}^*	bc_{MTZ-DL}^*	bc_{3s}	bc_{MTZ-DL}
p43.1	22545	UB	22545	22545	22545	22545	22545	22545	22545	22545
		LB	22408	22545	22545	22545	22545	22545	22545	22545
		gap	0.6	0	0	0	0	0	0	0
		t	72000	195	308	206	167	14	2583	5400
p43.2	22837	UB	22837	22837	22837	22837	22837	22837	22837	22837
		LB	22461	22711	22731	22837	22837	22774	22801	22674
		gap	1.7	0.6	0.5	0	0	0.3	0.2	0.7
		t	72000	72000	72000	22780	39365	72000	72000	72000
p43.3	23119	UB	23119	23119	23119	23119	23119	23119	23119	23119
		LB	22399	22293	22970	23119	23119	23089	23104	22877
		gap	3.2	3.7	0.6	0	0	0.1	0.1	1.1
		t	72000	72000	72000	8672	11665	72000	72000	72000
p43.4	66848	UB	66848	66848	66848	66848	66848	66848	66848	66848
		LB	45266	66848	66848	66848	66848	66848	66848	66743
		gap	47.7	0	0	0	0	0	0	0.2
		t	72000	2596	131	283	587	52	212	72000
prob.100	[830, 1343]	UB	1343	1516	1343	1343	1343	1343	1343	1343
		LB	813	824	826	824	826	830	813	824
		gap	65.2	84	62.6	63	62.6	61.8	65.2	63
		t	72000	72000	72000	72000	72000	72000	72000	72000
prob.42	202	UB	204	202	202	202	202	202	202	202
		LB	198	202	202	202	202	202	202	202
		gap	3	0	0	0	0	0	0	0
		t	72000	832	767	869	3559	120	1155	230
rbg048a	282	UB	282	282	282	282	282	282	282	282
		LB	282	282	282	282	282	282	282	282
		gap	0	0	0	0	0	0	0	0
		t	57	13	61	38	46	5	61	26
rbg050c	378	UB	378	378	378	378	378	378	378	378
		LB	378	378	378	378	378	378	378	378
		gap	0	0	0	0	0	0	0	0
		t	42	21	38	29	65	6,62	34	56
rbg109a	848	UB	848	848	848	848	848	848	848	848
		LB	848	848	848	848	848	848	848	837
		gap	0	0	0	0	0	0	0	1.3
		t	1942	51757	781	3567	3530	790	6583	72000
rbg150a	1414	UB	1414	1414	1414	1414	1414	1414	1414	1414
		LB	1414	1400	1414	1414	1414	1414	1414	1400
		gap	0	1	0	0	0	0	0	1
		t	21725	72000	7674	27154	70000	11950	42549	72000

Notes: best performers are highlighted in bold

As it follows from Tables 2.5–2.8, both baseline algorithms solved to optimality 17 out of 40 instances in total, where the instances rbg150a and rbg174a (which are huge ones) were solved by Gurobi solely, and the instances ft53.1, ft70.1, p43.1, p43.4, prob.42 – by bc_{DFJ}^* . In turn, proposed algorithms bc_1^* , bc_{3s}^* , bc_{5s}^* , bc_{MTZ-DL}^* managed to solve to optimality 24 out of 40 instances in total including all the mentioned above. Regarding to

the new seven instances, ft53.4 and ry48p.4 were solved by all of them, the instance rbg253a was solved by bc_1^* , the instance ft70.4 – by bc_{MTZ-DL}^* , the instance rbg323a – by bc_{3s}^* , bc_{5s}^* and bc_{MTZ-DL}^* . Finally, the optimal solutions of the instances p43.2 and p43.3 were found by both bc_{3s}^* and bc_{5s}^* . In addition, each of 15 instances solved to optimality by bc_{DFJ}^* is also solved exactly by one of the proposed variants about 11 times faster in average. Nevertheless, we should mention the instance ESC25, where bc_{DFJ}^* outperforms other competitors in terms of the elapsed time.

In the residual 16 open instances, the proposed algorithms managed to significantly increase lower bounds and close the average gap value about 3 times better than both baselines and complement each other quite well.

Our second observation is related to the comparison of variants bc_{3s}^* and bc_{MTZ-DL}^* with the corresponding counterparts bc_{3s} and bc_{MTZ-DL} obtained by exclusion the predecessor / successor inequalities from the separation pipeline. Regarding to bc_{3s} and bc_{3s}^* , we observe that inclusion of such inequalities allows to solve to optimality three additional instances (p43.2, p43.3, and rbg323a). Furthermore, for 12 out of 16 instances solved by both competitors exactly, we observe notable decrease of the running time. In addition, for the remaining 21 instances, bc_{3s}^* closed the gap by 1.7 times better in average. In turn, we should note that bc_{MTZ-DL}^* significantly outperforms bc_{MTZ-DL} in terms of instances solved to optimality, gap values and elapsed time.

Therefore, the predecessor/successor inequalities are proved to be useful for the PCGTSP in the branch-and-cut setting as well.

Table 2.8: Comparison of the branch-and-cut algorithms (d)

Instance	OPT		Gurobi	bc_{DFJ}^*	bc_1^*	bc_{3s}^*	bc_{5s}^*	bc_{MTZ-DL}^*	bc_{3s}	bc_{MTZ-DL}
rbg174a	1641	UB	1641	1641	1641	1641	1641	1641	1641	1641
		LB	1641	1602	1641	1636	1637	1638	1630	1607
		gap	0	2.4	0	0.3	0.2	0.2	0.7	2.1
		t	62657	72000	14448	72000	72000	72000	72000	72000
rbg253a	2372	UB	2373	2372	2372	2372	2372	2373	2372	2373
		LB	2369	2358	2372	2357	2357	2366	2350	2301
		gap	0.2	0.6	0	0.6	0.6	0.3	0.9	3.1
		t	72000	72000	27642	72000	72000	72000	72000	72000
rbg323a	2533	UB	2595	2597	2586	2533	2533	2533	2594	2554
		LB	2528	2517	2531	2533	2533	2533	2530	2493
		gap	2.7	3.2	2.2	0	0	0	2.5	2.4
		t	72000	72000	72000	71550	71800	54377	72000	72000
rbg341a	[2062, 2147]	UB	2180	2195	2199	2184	2147	2160	2184	2146
		LB	2047	2017	2056	2061	2062	2058	2060	1932
		gap	6.5	8.8	7	6	4.1	5	6	11.1
		t	72000	72000	72000	72000	72000	72000	72000	72000
rbg358a	[2037, 2166]	UB	2172	2174	2175	2172	2174	2185	2172	2166
		LB	1996	2009	2025	2025	2013	2037	2002	1949
		gap	8.8	8.2	7.4	7.3	8	7.3	8.5	11.1
		t	72000	72000	72000	72000	72000	72000	72000	72000
rbg378a	[2236, 2385]	UB	2390	2385	2404	2400	2402	2408	2400	2399
		LB	2185	2191	2210	2214	2205	2236	2132	2080
		gap	9.4	8.9	8.8	8.4	8.9	7.7	12.6	15.3
		t	72000	72000	72000	72000	72000	72000	72000	72000
ry48p.1	[13109, 13135]	UB	13135	13135	13135	13135	13135	13135	13135	13135
		LB	12065	12732	12634	13084	12914	13109	12669	12625
		gap	8.9	3.2	4	0.4	1.7	0.2	3.7	4.0
		t	72000	72000	72000	72000	72000	72000	72000	72000
ry48p.2	[13401, 13802]	UB	13802	13802	13802	13802	13802	13802	13802	13802
		LB	12217	12963	12917	13401	13327	13275	13019	12789
		gap	13	6.5	6.9	3	3.6	4.0	6	7.9
		t	72000	72000	72000	72000	72000	72000	72000	72000
ry48p.3	[15778, 16533]	UB	16533	16533	16533	16553	16533	16533	16533	16533
		LB	13387	14753	14825	15147	15441	15778	14672	14532
		gap	23.5	12.1	11.5	9.3	7.1	4.8	12.7	13.8
		t	72000	72000	72000	72000	72000	72000	72000	72000
ry48p.4	25977	UB	25977	25977	25977	25977	25977	25977	25977	25977
		LB	22732	24079	25977	25977	25977	25977	25977	24069
		gap	14.3	7.9	0	0	0	0	0	7.9
		t	72000	72000	11182	22106	29865	4376	25000	72000

Notes: best performers are highlighted in bold

Chapter 3

Discrete Cutting Path Problems: General Solution Framework with Accuracy Guarantees

3.1 General modelling approach: from the discrete CPP to the PCGTSP

In this section, we present our general modeling approach based on the reduction of an arbitrary instance of CPP to PCGTSP. The PCGTSP is based on the formulation of GTSP complemented with precedence constraints [see, e.g. [Salman et al., 2020](#), [Khachai et al., 2023](#)].

An instance of the GTSP [[Gutin and Punnen, 2007](#)] is defined by an edge-weighted directed graph $G = (V, E, c)$, a cost function $c: E \rightarrow \mathbb{R}$, and partition $\mathfrak{C} = \{C_1, C_2, \dots, C_m\}$ of the nodeset $V = \{v_1, \dots, v_n\}$ into non-empty mutually disjoint *clusters* (or megalopolises). The goal is to construct a closed route $R = v_{i_1}, \dots, v_{i_m}$ that visits each cluster $C_j \in \mathfrak{C}$ exactly once and has the minimum cost

$$c(R) = c(v_{i_m}, v_{i_1}) + \sum_{j=1}^{m-1} c(v_{i_j}, v_{i_{j+1}}).$$

The GTSP appears to be a natural generalization of the classic TSP corresponding to GTSP where each cluster is a singleton.

In the PCGTSP, precedence constraints define a *partial order* on the set of clusters \mathfrak{C} . This order is given by a directed acyclic graph (DAG) $H = (\mathfrak{C}, A)$, such that $(C_i, C_j) \in A$ if and only if $C_i \prec C_j$, i.e. the cluster C_i *precedes* C_j . Furthermore, in the PCGTSP, the cluster C_1 is called *depot* and plays a specific role: each feasible route R is restricted to depart from and to arrive to some node $v_{i_1} \in C_1$ as well as to visit all other clusters with respect to the order specified by the DAG H .

In the following, we first consider discrete CPP without cutting preemption and then discuss ECP where such a preemption is allowed.

3.1.1 Discrete CPP without cutting preemption

Each part to be cut is defined by a closed contour that can be complemented with a family of mutually disjoint enclosed contours. Each contour should be cut out without preemption, using the following cutting procedure [Petunin and Stylios, 2016]: the tool starts cutting in a dedicated *piercing* point π_i ; then it moves to a special equidistant curve of the contour and cuts it out entirely, from the *entry* point ε_i ; the cutting is finalized at the *exit* point θ_i , where the tool is switched off. Then, it moves *by air* to proceed with another contour or to return to the resting point (see Fig. 3.1). In the considered discrete CPP without cutting preemption, a finite set of feasible entry, piercing and exit points are known for each contour.

An instance \mathcal{I}_0^1 of the considered discrete CPP without cutting preemption is defined by:

- a set of contours K_1, \dots, K_t and the information about their mutual topology,

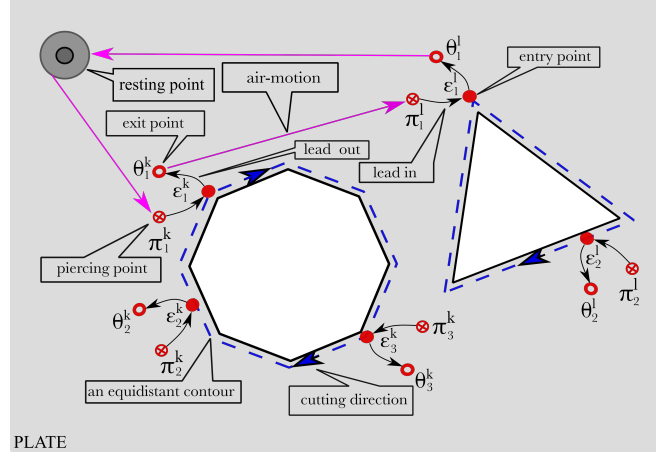


Figure 3.1: An example of the discrete CPP without cutting preemption

- a set of triples $\{(\pi_j^k, \varepsilon_j^k, \theta_j^k) : j \in \{1, \dots, n_k\}\}$ associated with each contour K_k ,
- piercing costs ρ_j^k that include the cost of piercing itself and lead-in / lead-out cutting expenses as well,
- air-motion costs $\delta_{j,i}^{k,l}$ for moving from contour K_k to contour K_l (with respect to the exit point θ_j^k and piercing point π_i^l),
- air-motion costs μ_j^k and ν_j^k for moving from the rest point v_{rest} to the piercing point π_j^k and from the exit point θ_j^k respectively associated with contour K_k .

The goal is to construct a cutting path of a minimum cost. According to the problem statement, actual cutting costs for the contours are constant, therefore they are excluded from the objective function.

In the following, we construct the correspondence between instance \mathcal{I}_0^1 and instance \mathcal{I}_1 of the PCGTSP. Let us define the edge-weighted digraph $G = (V, E, c)$:

$$V = \{v_{rest}\} \cup \{v_j^{k+1} : k \in \{1, \dots, t\}, j \in \{1, \dots, n_k\}\},$$

$$\begin{aligned}
E &= \{(v_{rest}, v_j^{k+1}), (v_j^{k+1}, v_{rest}) : k \in \{1, \dots, t\}, j \in \{1, \dots, n_k\}\} \\
&\cup \{(v_j^{k+1}, v_i^{l+1}) : k \neq l \in \{1, \dots, t\}, j \in \{1, \dots, n_k\}, i \in \{1, \dots, n_l\}\}, \\
c(v_{rest}, v_j^{k+1}) &= \mu_j^k + \rho_j^k, \quad c(v_j^{k+1}, v_{rest}) = \nu_j^k, \quad c(v_j^{k+1}, v_i^{l+1}) = \rho_i^l + \delta_{j,i}^{k,l}.
\end{aligned}$$

where v_{rest} encodes the resting point for the cutting tool, and any node v_j^{k+1} corresponds to the triple $(\pi_j^k, \varepsilon_j^k, \theta_j^k)$.

We define the partition $\mathcal{C} = \{C_1, C_2, \dots, C_m\}$ by splitting the nodeset V into $m = t + 1$ pairwise disjoint clusters, where $C_1 = \{v_{rest}\}$ is the depot cluster, and $C_k = \{v_j^k : j = \{1, \dots, n_{k-1}\}\}$, for any $k \in \{2, \dots, m\}$. Precedence constraints are defined in a natural way, i.e. for an arbitrary $k \neq l \in \{2, \dots, m\}$, $C_p \prec C_q$ if contour K_{p-1} is embedded into K_{q-1} . The following proposition holds.

Proposition 13. *There exists a polynomial time reduction, for which instances \mathcal{I}_0^1 and \mathcal{I}^1 are equivalent, i.e. an arbitrary feasible path of the cutting tool in \mathcal{I}_0^1 corresponds to a feasible solution of \mathcal{I}^1 of the same cost, and vice versa.*

Proof. Consider an arbitrary feasible cutting path for a given instance of the discrete CPP without cutting preemption (see, e.g. Fig. 3.2) that starts / finishes each contour K_k at specified piercing point π_j^k and θ_j^k , respectively. According to the construction procedure, in the corresponding PCGTSP instance, each contour K_k is represented by a cluster C_{k+1} , whose nodes correspond to the triples associated with this contour. Furthermore, by visiting the cluster C_{k+1} in some node v_j^{k+1} , we model the cutting out of the contour K_k with respect to the triple $(\pi_j^k, \varepsilon_j^k, \theta_j^k)$. Therefore, the considered cutting path explicitly corresponds to a feasible solution of the auxiliary PCGTSP instance of the same cost.

On the other hand, to an arbitrary feasible solution of the PCGTSP, we assign a corresponding feasible cutting path of the discrete CPP without cutting preemption in a similar way.

Finally, by construction, the size $|\mathcal{I}^1|$ of the obtained instance \mathcal{I}^1 and the running time of the proposed reduction do not exceed $\psi_1 \cdot |\mathcal{I}_0^1|$, where $|\mathcal{I}_0^1|$ is the size of the initial instance \mathcal{I}_0^1 and $\psi_1 \geq 1$ is some constant. \square

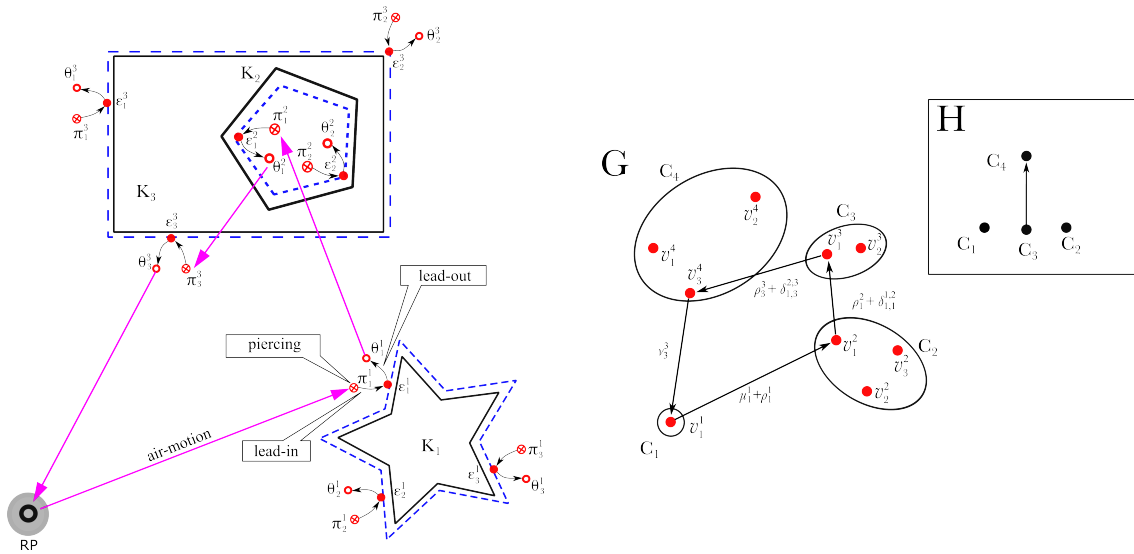


Figure 3.2: Example of the reduction: initial instance with a feasible cutting path(left) and its auxiliary PCGTSP instance (right). In this example, we have the only precedence constraint induced by the embedding of the contour K_2 into K_3 . Therefore, K_2 should be cut out before K_3 , and its PCGTSP counterpart C_3 should be visited by an arbitrary feasible route before C_4 . Thus, the DAG H has the only arc (C_3, C_4)

Remark. The proposed reduction covers the general setting of the discrete CPP without cutting preemption. In the special case of a thin material, piercing costs are low and can be ignored, and each equidistant curve is close to the corresponding contour. Therefore, in this case, without loss of generality, we can assume that any triple $(\pi_i, \varepsilon_i, \theta_i)$ collapses to the single point, and the resulting PCGTSP instance becomes symmetric and equivalent to finding a cutting path with the minimal air-motion cost.

In the next section we consider ECP where cutting preemption is allowed.

3.1.2 ECP: several sources of industrial constraints

In the ECP, each contour can be partitioned into several *elements*, each of them can be cut out separately. As it was shown in [Dewil et al., 2014], such cutting method induces several sources of precedence constraints. For the sake of convenience, in this section, we consider each source separately. The proposed reduction techniques complement each other. If necessary, the decision makers can combine them to treat more complex settings.

We start with the case of a thin material, where the objective is to construct a cutting path minimizing the air-motion costs. We assume that a cutting element corresponds to an arbitrary connected (closed or non-closed) segment of a contour to be cut out. Such an element is represented by a two-node *element* cluster, each its node encodes a moving direction of the cutting tool (Fig. 3.3).



Figure 3.3: Encoding of a cutting element

For a number of reasons, it is important to keep the relation between the original contour and its composing elements. In order to accomplish this, we introduce a concept of a *contour gadget*.

For any contour, such a gadget consists of two-node clusters of all its elements and an *artificial twin* cluster representing this contour itself (see Fig. 3.4). Each node of any element cluster is connected with a single zero-cost arc with the corresponding node in the twin cluster. Furthermore, any outgoing arc for the original element node is doubled for its corresponding twin node. Finally, the artificial cluster is a direct successor of all respective element clusters.

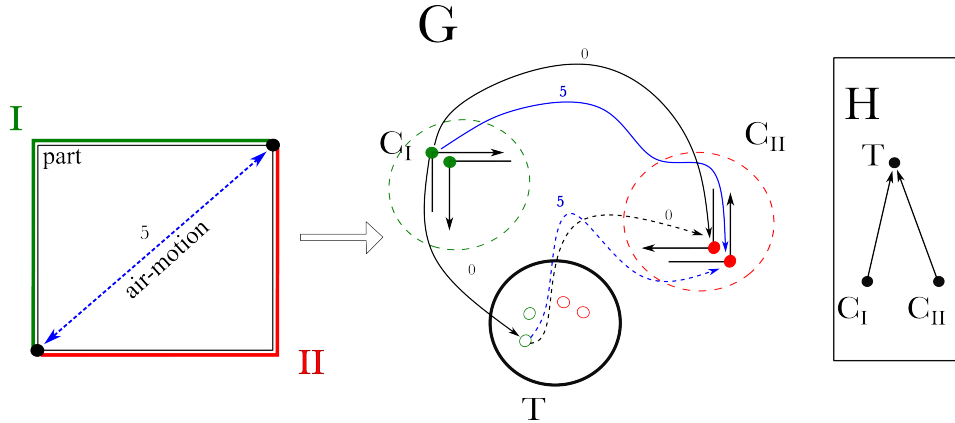


Figure 3.4: Contour gadget: twin nodes, arc doubling and induced precedence constraints. In this case, the initial contour consists of elements I and II. To this end, the corresponding contour gadget is represented by two-node clusters C_I and C_{II} and their successor – artificial twin cluster T . Any arc going from two-node clusters is a bold line, while its corresponding artificial twin arcs are labeled by a dash line

As a consequence, according to the definition of the PCGTSP, a feasible route must visit an artificial twin cluster associated with a contour immediately after its last-visited element cluster.

Case 1. *Inner/outer contours* relation can be encoded in terms of the introduced gadgets almost straightforwardly. In the ECP, this relation induces the following cutting rule: the last element of an outer contour must be cut out only after all the elements of any inner contour have been completely cut.

Consider an instance \mathcal{I}_0^2 of the ECP, specified by contours K_1, \dots, K_t , where each contour K_j consists of elements $E_1^j, \dots, E_{k(j)}^j$ for some $k(j) \geq 1$ and costs charging air motion between non coincident endpoints of different contour elements. It is required to construct a tool path, departing from and arriving to the rest point, while cutting out all the contours with respect to their possible embedding. To such an instance \mathcal{I}_0^2 of the ECP, we assign an auxiliary instance \mathcal{I}^2 of the PCGTSP as follows.

First, to any contour K_j we assign a contour gadget G_j consisting of all its element clusters and a single artificial twin cluster. We obtain

graph $G(V, E)$, where $V = V(G_1) \cup V(G_2) \cup \dots \cup V(G_m)$ and $E = E(G_1) \cup E(G_2) \cup \dots \cup E(G_m) \cup E'$, and E' is composed of all arcs between G_1, \dots, G_m and the depot. The arc costs are defined equal to the respective air-motion costs (see Fig. 3.4). The partition into pairwise disjoint clusters \mathcal{C} is established in a natural way. Finally, the precedence constraints, defined by the graph H , are two-fold. First, each twin cluster T_j is a direct successor of all element clusters with respect to its gadget. Second, $T_j \prec T_l$ if and only if $K_j \prec K_l$. Therefore, the reduction from the ECP to the PCGTSP in the case of inner/outer contours is obtained (see Fig. 3.5).

In order to estimate complexity of the proposed reduction, observe that both sizes $|\mathcal{I}_0^2|$ and $|\mathcal{I}^2|$ of the initial instance \mathcal{I}_0^2 and the obtained auxiliary instance \mathcal{I}^2 appears to be proportional to \mathcal{T}^2 for $\mathcal{T} = \sum_{j=1}^t k(j)$. Therefore, similarly to Subsection 3.1.1, there exist a constant $\psi_2 \geq 1$, such that $|\mathcal{I}^2|$ and reduction time do not exceed $\psi_2 \cdot |\mathcal{I}_0^2|$, and the following proposition holds.

Proposition 14. *Instance \mathcal{I}^2 is equivalent to the initial instance \mathcal{I}_0^2 , and obtained in polynomial time.*

Case 2. *Common cuts* can be tackled in a similar way, however there is a new notation to be introduced. In the cases mentioned above, each contour was assigned to one part. Common cuts can induce several ‘artificial’ contours called here *shapes*. A shape is a closed contour (not necessary assigned to a single part), whose interior is intersected by a common cut. Here, the cutting rule is to ensure that the common cut is carried out before the last element of any shape whose interior is crossed by this cut (Fig. 3.6).

Consider an arbitrary instance \mathcal{I}_0^3 of the ECP, given by some regular elements and common cuts that induce shapes. To this instance, we assign an instance \mathcal{I}^3 of the PCGTSP. To each shape we assign a contour

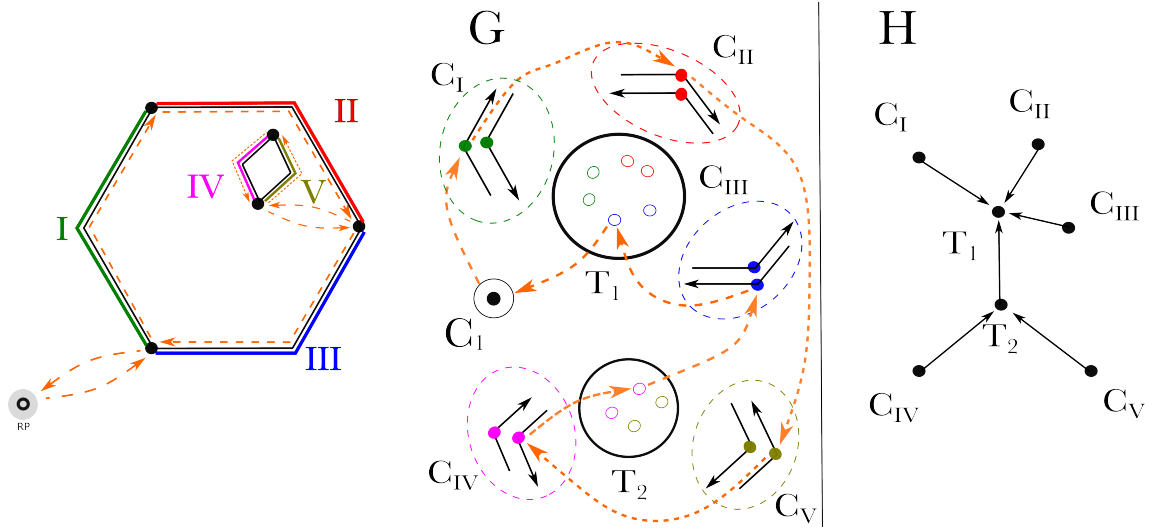


Figure 3.5: Instance of the ECP with inner/outer contours, a feasible cutting path is colored orange. In this example, we have two contours. The first one consists of the elements I, II, and III, while the second one of the elements IV and V. Therefore, the corresponding graph G contains two contour gadgets, specified by clusters $C_I, C_{II}, C_{III}, T_1$, and C_{IV}, C_V, T_2 respectively. Since, the second contour is embedded into the first one, cluster T_2 precedes T_1 , which is encoded by the arc (T_2, T_1) in the graph H . Feasible cutting path and the appropriate same cost PCGTSP solution are presented by orange

gadget. Unlike the previous case, several contour gadgets may contain the same element cluster. As a consequence, some of them may have several related twins. Since every outgoing arc from the element cluster node is being replicated by its twin, those twins create a clique (complete directed sub-graph) with each other (see Fig. 3.7(b)). In order to fulfill the cutting rule, we extend our order by adding the precedence constraints between each common cut cluster and twin clusters of all associated shapes. As a result, the graph H appears to be a bipartite directed graph.

To estimate complexity of the proposed reduction, observe that the number of two-node element clusters in the graph G is equal to $\mathcal{N}_{ce} + \mathcal{N}_{cc}$, where \mathcal{N}_{ce} and \mathcal{N}_{cc} are numbers of contour elements and common cuts of the initial instance \mathcal{I}_0^3 , respectively. Let further \mathcal{N}_S be the number of shapes induced by the common cuts. As it follows from ?, \mathcal{N}_S is at proportional to $(\mathcal{N}_{cc})^2$, while each shape cluster has at most \mathcal{N}_{cc} nodes, by

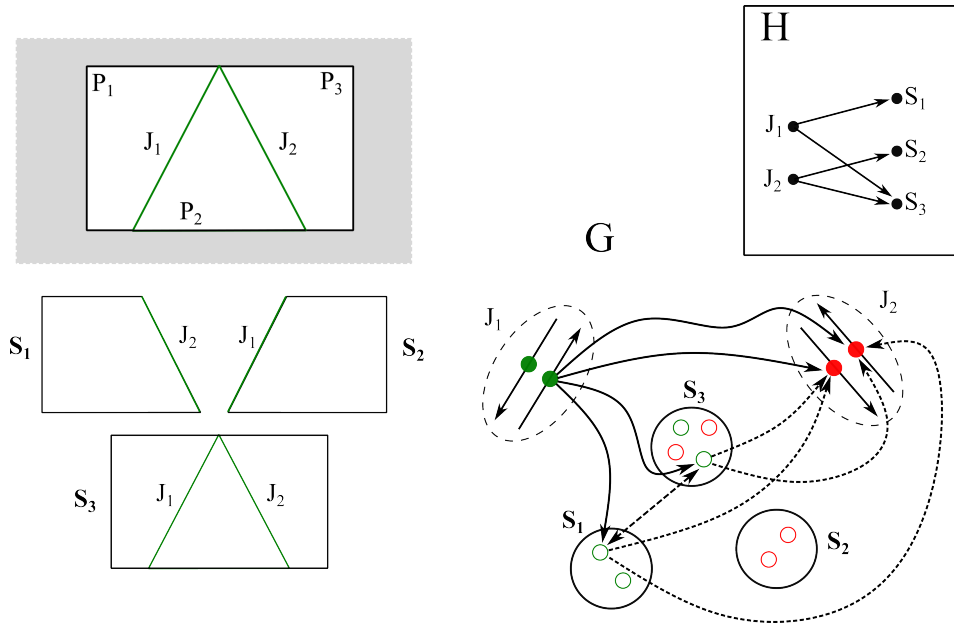


Figure 3.6: Common cuts can induce several shapes. In this example, part P_2 has two cuts, J_1 and J_2 common with parts P_1 and P_3 , respectively. These common cuts induce the shapes S_1 , S_2 , and S_3 , such that J_1 intersects interiors of S_1 and S_3 , while J_2 — interiors of S_2 and S_3 . Therefore, in graph G , each node of the cluster J_1 (J_2) has a twin in the clusters S_1 and S_3 (S_2 and S_3) inheriting all the outgoing arcs. Each twins of the same node are induced cliques. Furthermore, precedence of the cluster J_1 with respect to the clusters S_1 and S_3 (J_2 with respect to S_2 and S_3) is encoded by corresponding arcs of the graph H

construction. Therefore, there exists a constant $\psi_3 \geq 1$, the size $|\mathcal{I}^3|$ of the constructed auxiliary PCGTSP instance \mathcal{I}^3 does not exceed $\psi_3 \cdot (|\mathcal{I}_0^3|)^2$ and the running time of the reduction is bounded by some small degree polynomial of $|\mathcal{I}_0^3|$. Thus, we have the following statement.

Proposition 15. *An instance \mathcal{I}_0^3 of the ECP with common cuts can be reduced to the appropriate instance \mathcal{I}^3 of the PCGTSP in polynomial time.*

As in the previous case, an example given in the Fig.3.7(a) illustrates a possible cutting path for the ECP, and Fig.3.7(b) for its PCGTSP counterpart.

Case 3. Some instances of the ECP can have *islands*, which are bounded areas of waste material encompassed by several contours (Fig. 3.8). This

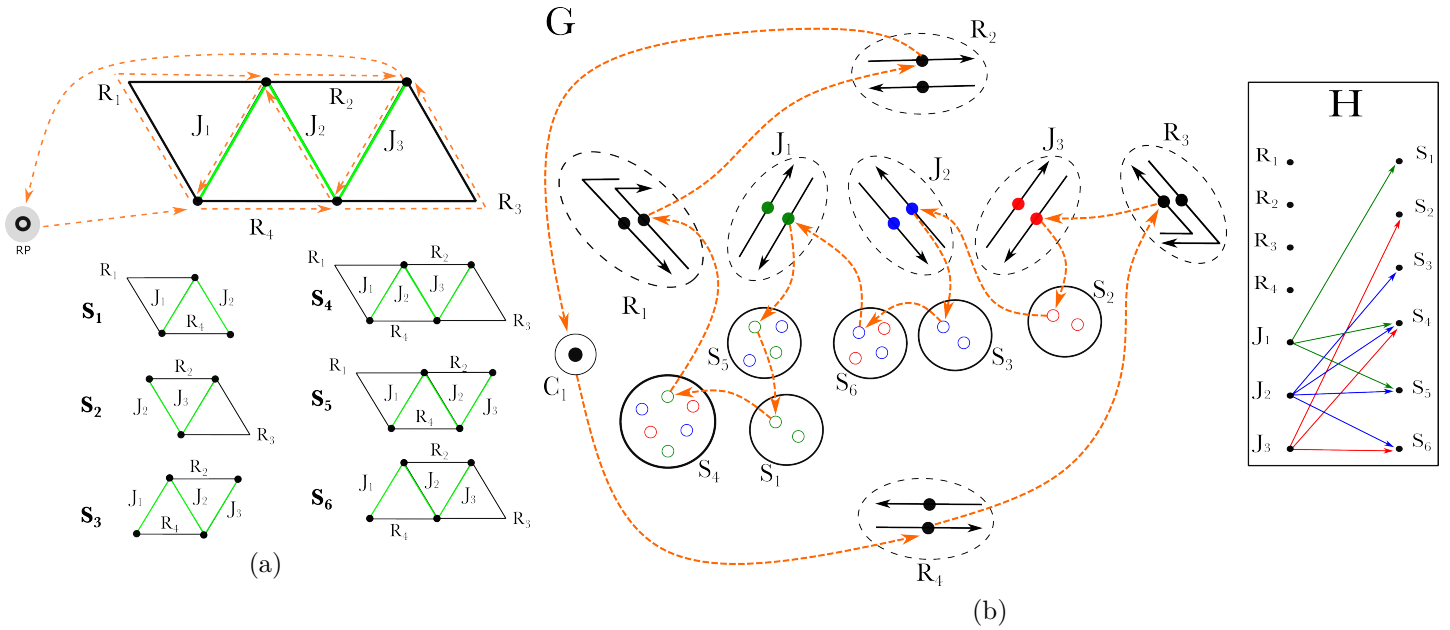


Figure 3.7: An example of the ECP instance with common cuts, and the corresponding auxiliary instance of the PCGTSP. Regular cutting elements are denoted by R_1, \dots, R_4 , common cuts – by J_1, J_2, J_3 , and the induced shapes – by S_1, \dots, S_6 . All of them is represented by the same-name cluster in the graph G . Observe that the cluster J_1 precedes clusters S_1, S_4 , and S_5 , since the respective common cut intersects interior of these shapes, which is encoded by the arcs $(J_1, S_1), (J_1, S_4), (J_1, S_5)$ in the graph H . The rest of the precedence constraints is obtained in the similar way. A feasible cutting path and the appropriate same cost PCGTSP route is colored orange

case can be reduced to the previous one by assuming each island to be a dummy part, whose contour being composed of common cuts. Complexity of the proposed reduction has the same upper bound as in Case 2.

Case 4. Finally, we consider the case of the ECP with *thick* material. In this case, piercing costs cannot be ignored anymore and the optimization criterion needs to include also piercing, pre-cut and optional bridge cutting costs. In practice, before starting any actual cutting, a preliminary phase is performed, it includes piercing into the material, pre-cutting and so on. In order to represent these procedures, we introduce a novel *crossroad* cluster, which encodes a junction between several contour elements (or

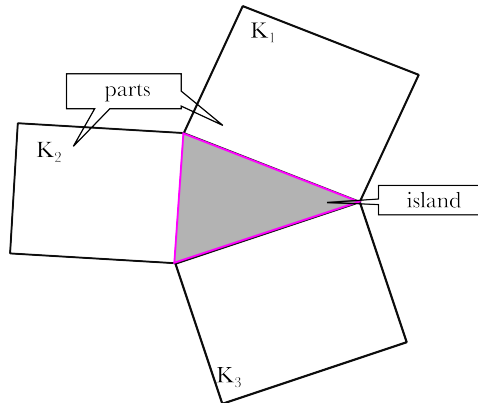


Figure 3.8: Instance of the ECP with an island

bridges). In addition, we introduce novel *element gadgets* and *bridge gadgets* (Fig. 3.9).

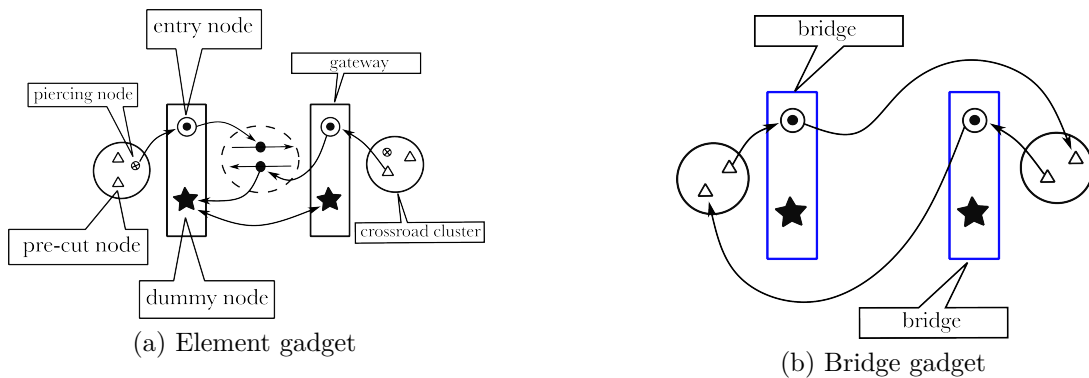


Figure 3.9: Crossroads, cutting elements and bridges

Any crossroad cluster consists of (an optional) piercing node and a dedicated node for each incident cutting elements, which are encoded by an element gadget. An element gadget, besides the known two-node cluster specifying cutting direction long this element, consists of additional *gateway* clusters. Such a cluster consists of a node encoding an entry point of the associated element and a *dummy* node, that encodes completion of the tool path and return to the depot. To complete representation of the aforementioned piercing procedure, we introduce precedence constraints between any crossroad cluster and all the incident gateway clusters.

In turn, bridge gadget has mostly the same structure, except its crossroads do not contain any piercing nodes, and bridge gateways have no element clusters between them. It allows traversing across the bridge in both directions or skipping it completely. According to the construction procedure, a crossroad cluster precedes all the nearby gateways of the incident cutting elements.

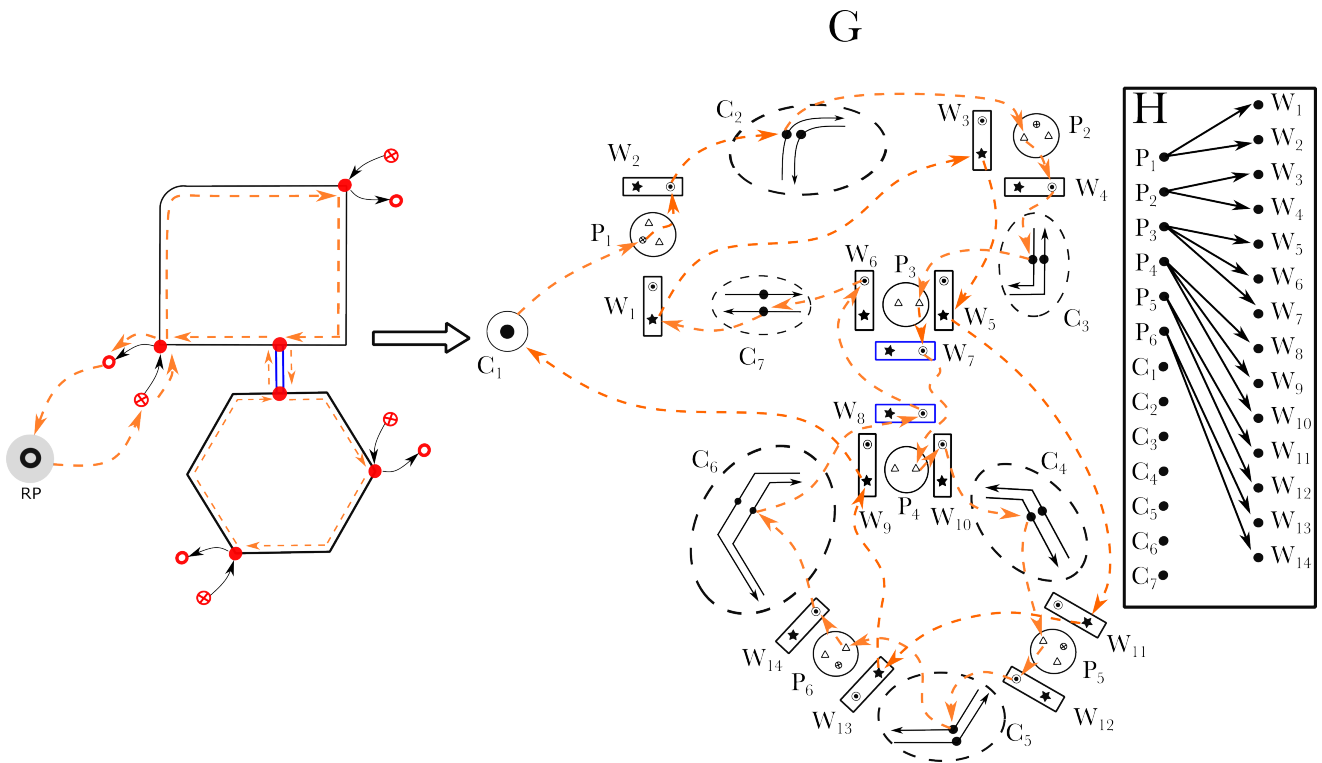


Figure 3.10: Reduction of the ECP with bridges to the equivalent instance of the PCGTSP: P_i are crossroads, W_j are gateways, and C_q are remaining clusters representing contour elements. Observe that in this case, the nature of precedence constraints is specified by the fact that each crossroad cluster precedes all the gateways assigned to it. For instance, cluster P_1 precedes the clusters W_1 and W_2 , which is encoded by the arcs (P_1, W_1) and (P_1, W_2) in the graph H . A feasible cutting path and the same cost appropriate feasible solution of the PCGTSP are presented by orange

Consider an arbitrary instance \mathcal{I}_0^4 of a thick-material setting of the ECP. Similarly to \mathcal{I}_0^1 , this instance is given by a finite set of contours K_1, \dots, K_t , such that each contour K_k is augmented by triples $\{(\pi_j^k, \varepsilon_j^k, \theta_j^k) : j \in \{1, \dots, n_k\}\}$ of piercing, entry and exit points and corresponding air-

motion, piercing, and lead-in / lead-out costs. In addition, there are several bridges B_i , each of them connects a pair of contours (K_i, K_j) . Since, the setting of ECP is free from the ‘no cutting preemption’ constraint, we extend our notation as follows:

- in this case, ρ_j^k denotes piercing cost only at the point π_j^k of the contour K_k , while the corresponding lead-in and lead-out costs are denoted by σ_j^k and τ_j^k respectively;
- along with costs $\delta_{j,i}^{k,l}$ corresponding to the air-motion between θ_j^k and π_i^l , we introduce similar costs $\gamma_{j,i}^{k,l}$ for air-motion between the points θ_j^k and ε_i^l ;
- traversing through the bridge B_i induces the cutting cost β_i ;
- finally, the rest point air-motion costs μ_j^k and ν_j^k remain the same as in the case of the discretized CCP.

By assigning to each cutting element or bridge a dedicated element and bridge gadget, as well as to their junctions - an appropriate crossroad, we naturally obtain the nodeset V of the graph $G(V, E)$ for this case, and its partition into the set of clusters. The precedence constraints are established between crossroad and gateway clusters, as it was mentioned above. The set E consists of:

- arcs connecting the depot with all the piercing nodes (Fig.3.9(a)), their costs are obtained as the sum of the appropriate air-motion costs and piercing costs, i.e. $\mu_j^k + \rho_j^k$;
- internal arcs of an arbitrary element gadget, including zero-cost pre-cut/gateway entry node, entry/same direction element cluster node, element cluster/self pre-cut node arcs, and arcs between the piercing node and self gateway entry node of cost σ_j^k ;

- element cluster/dummy node arcs of sum of lead-out cost and depot air-motion cost $(\tau_j^k + \nu_j^k)$ respectively;
- bridge gadget arcs (Fig.3.9(b)) between entry node and the pre-cut node of an appropriate crossroad cluster, of bridge-cut cost β_i ;
- any arc between bridge entry node and gateway entry node of bridge-cut cost β_i ;
- arcs between any element cluster node and foreign pre-cut node of sum of lead-out and air-motion cost $(\tau_j^k + \gamma_{j,i}^{k,l})$;
- arcs between any element cluster node and foreign piercing node of sum of lead-out, air-motion and piercing costs $(\tau_j^k + \delta_{j,i}^{k,l} + \rho_j^k)$;
- arcs between any pre-cut node and foreign piercing node of sum of lead-out, air-motion and piercing costs $(\tau_j^k + \delta_{j,i}^{k,l} + \rho_j^k)$;
- zero-cost arcs between all pre-cut nodes of an arbitrary crossroad cluster and entry nodes of nearby gateways;
- zero-cost arcs of the clique spanning all the dummy nodes.

Thus, we construct a corresponding instance \mathcal{I}^4 of the PCGTSP (Fig. 3.10). Similarly to Case 1, we can show that both the size $|\mathcal{I}^4|$ of the obtained auxiliary instance and the running time of the proposed reduction depend linearly on the size $|\mathcal{I}_0^4|$ of the initial instance \mathcal{I}_0^4 .

The following proposition summarize our argument.

Proposition 16. *Instance \mathcal{I}^4 is equivalent to the initial instance \mathcal{I}_0^4 , and obtained in polynomial time.*

3.2 Generic solution framework for the discrete CPP

In this section, we introduce the general algorithmic approach for various formulations of the discrete Cutting Path Problem. It relies on:

- the modelling techniques presented in Section 3.1 providing the equivalence to the Precedence Constrained Generalized Traveling Salesman Problem;
- a recent state-of-the-art branch-and-cut algorithm (Khachai et al. [2023]) that can be used both for exact and approximate resolution of the problem with accuracy guarantees;
- PCGLNS heuristic for finding approximate solutions that according to the performed tests on industrial problems in Section 3.3 can be close-to-optimal or even optimal in practice.

The general solution framework is summarized in Algorithm 2.

Algorithm 2 Scheme of the proposed solution framework

Input: an instance \mathcal{I}_0 of the discrete CPP,

Output: a cutting path of the minimal cost or a sub-optimal cutting path with accuracy guarantees.

- 1: Analyze the initial instance of the discrete CPP, its cutting restrictions, such as common cuts, bridges, inner-outer contours etc.
 - 2: Transform \mathcal{I}_0 to the corresponding instance \mathcal{I} of the PCGTSP with the use of the techniques developed in Section 3.1.
 - 3: Find an optimal or close-to-optimal approximate solution of \mathcal{I} with the branch-and-cut algorithm or PCGLNS heuristic.
 - 4: Transform this solution into the corresponding same cost cutting path for the initial discrete CPP instance \mathcal{I}_0 .
 - 5: **return** the required CPP cutting path.
-

In the sequel, we briefly describe the methodological support for our framework then present an illustrative example of industrial application.

3.2.1 Branch-and-cut algorithm

The employed branch-and-cut algorithm extends well-known linear programming relaxation branching technique and consists of the following components:

- Mixed Integer Linear Program (MILP) formulation of the problem in question;

- primal heuristic to produce close-to-optimal feasible solutions in a short time;
- problem-specific cutting planes providing accuracy bounds of the obtained solutions.

MILP formulation

We employ our extension of the well-known compact model for the classic ATSP [Miller et al., 1960], lifted by Desrochers and Laporte [1991], that we further strengthened and adapted to the case of the precedence constraints. This model is chosen because it is one of the best performers for the general PCGTSP in terms of both LP-relaxation bounds and running time [Khachai et al., 2023].

To any arc $(i, j) \in E$ and node $v \in V$, we assign binary decision variables as follows:

$$x_{ij} = \begin{cases} 1, & \text{if } (i, j) \text{ is included to the solution} \\ 0, & \text{otherwise,} \end{cases} \quad z_v = \begin{cases} 1, & \text{if the solution visits } v \\ 0, & \text{otherwise.} \end{cases}$$

In order to describe inter-cluster transitions, we include additional variables

u_{pq}

$$u_{pq} = \begin{cases} 1, & \text{if in the solution, cluster } C_q \text{ follows } C_p \text{ immediately} \\ 0, & \text{otherwise,} \end{cases}$$

and v_p that gives the number of clusters visited in the solution between the depot C_1 and the cluster C_p . Although, for any feasible route, values of the variables u_{pq} and v_p are integer, it is not necessary to require it explicitly in the model.

We use the standard notation

$$\delta^-(C) = \{(j, i) : i \in C, j \notin C\} \text{ and } \delta^+(C) = \{(i, j) : i \in C, j \notin C\}$$

for *inbound* and *outbound cuts* for an arbitrary proper subset $\emptyset \neq C \subset V$ and the appropriate shortcuts $\delta^-(i)$ and $\delta^+(i)$ for any singleton $C = \{i\}$.

Thus, the proposed formulation is as follows:

$$\min \sum_{(i,j) \in E} c_{ij} x_{ij}, \quad (3.1)$$

$$\text{s.t.} \quad \sum_{i \in C_p} z_i = 1 \quad (p \in \{1, \dots, m\}) \quad (3.2)$$

$$\sum_{(i,j) \in \delta^+(i)} x_{ij} = z_i \quad (i \in V) \quad (3.3)$$

$$\sum_{(i,j) \in \delta^-(i)} x_{ji} = z_i \quad (i \in V) \quad (3.4)$$

$$\sum_{q=1, q \neq p}^m u_{pq} = 1 \quad (p \in \{1, \dots, m\}) \quad (3.5)$$

$$\sum_{p=1, p \neq q}^m u_{pq} = 1 \quad (q \in \{1, \dots, m\}) \quad (3.6)$$

$$\sum_{i \in \delta^+(C_p)} \sum_{j \in \delta^-(C_q)} x_{ij} = u_{pq} \quad (p, q \in \{1, \dots, m\}, p \neq q) \quad (3.7)$$

$$v_p - v_q + (m-1)u_{pq} + (m-3)u_{qp} \leq m-2 \quad (p, q \in \{2, \dots, m\}, p \neq q) \quad (3.8)$$

$$-v_p + (m-3)u_{p1} + \sum_{q=2}^m u_{qp} \leq 0 \quad (p \in \{2, \dots, m\}) \quad (3.9)$$

$$v_p + (m-3)u_{1p} + \sum_{q=2}^m u_{pq} \leq m-2 \quad (p \in \{2, \dots, m\}) \quad (3.10)$$

$$v_q - v_p \geq 1 \quad (p, q \in \{2, \dots, m\}: C_p \prec C_q) \quad (3.11)$$

$$x_{ij}, z_i \in \{0, 1\}, u_{pq} \geq 0, v_p \geq 0 \quad (3.12)$$

The goal is to find a feasible solution of the minimal cost (3.1). Constraints (3.2) ensure that each cluster is visited exactly in a single node.

Equations (3.3) and (3.4) are the flow conservation in terms of nodes, while equations (3.5) and (3.6) establish the same constraints for the clusters. In turn, equations (3.7) define an expansion of u_{pq} in terms of decision variables. Finally, to eliminate subtours and ensure precedence relations, we include Desrochers-Laporte-like constraints (3.8)–(3.10) along with (3.11).

PCGLNS primal heuristic

The PCGLNS meta-heuristic is the main primal heuristic in our framework. It was developed recently for the PCGTSP in [Khachay et al., 2020a]. It is a generalization of the well-known Adaptive Large Neighborhood Search (ALNS) metaheuristic [see, e.g. Gendreau and Potvin, 2019].

The built-in pool of removal heuristics \mathcal{R} is used to (partially) destroy current found feasible route. This pool includes:

- *worst removal* heuristic removes the node that maximizes the removal cost for the input route;
- *distance removal* heuristic removes the node, whose distance to some randomly chosen node in a route fulfills some predefined criterion, e.g. takes smallest, largest or random value;
- *segment removal* removes a randomly chosen continuous segment of the input route.

The additional insertion heuristics pool \mathcal{I} provides built-in algorithms applied for recovering of the route and consists of well-known *nearest*, *farthest*, *cheapest* and *random* insertion local-search heuristics. The main routine of the PCGLNS employs an online learning for fine tune distributions on sets \mathcal{R} and \mathcal{I} . At any warm restart, it picks a removal heuristic H_{rem} and an insertion one H_{ins} to obtain the new route, which is compared with the current best known solution. Each taken heuristic receives a

specific score based on the current and best tour costs. The higher score heuristic obtains, the more probable is its selection for the next restart.

To avoid local optima, the PCGLNS applies the standard *simulated annealing* acceptance criterion based on the internal *temperature* parameter \mathcal{T} decreasing successively along the iterations from a given initial value and affecting the acceptance probability $P(T_{new})$ for newly found tour

$$P(T_{new}) = \min\{e^{(c(T)-c(T_{new}))/\mathcal{T}}, 1\}.$$

Problem-specific cutting planes

To improve LP-relaxation bounds and speed-up the overall branch-and-cut pipeline, we extend the MILP-model (3.1)–(3.12) with additional families of *valid* inequalities. An inequality is called valid, if it is fulfilled by an arbitrary feasible solution of the MILP-model corresponding to a feasible route of the initial PCGTSP instance. Meanwhile, this inequality can be violated by some feasible solutions of the appropriate LP-relaxation, making it possible to exploit this constraint to produce cutting planes.

To describe the chosen families of valid inequalities, we use the following notation. If \mathcal{U} and \mathcal{V} are non-empty disjoint subsets of clustering \mathcal{C} , then

$$x(\mathcal{U}, \mathcal{V}) = \sum_{C_p \subset \mathcal{U}} \sum_{C_q \subset \mathcal{V}} u_{pq} \equiv \sum_{C_p \subset \mathcal{U}} \sum_{C_q \subset \mathcal{V}} \sum_{i \in C_p} \sum_{j \in C_q} x_{ij}.$$

In addition, for an arbitrary $p \in \{2, \dots, m\}$, by $\pi(C_p) = \{C_q : q > 1, C_q \prec C_p\}$ and $\sigma(C_p) = \{C_q : q > 1, C_p \prec C_q\}$ we denote the sets of *predecessors* and *successors* of the cluster C_p , respectively. Let $\mathfrak{C}' = \{C_{i_1}, \dots, C_{i_{2t+1}}\} \subset \mathfrak{C} \setminus \{C_1\}$, $t \in \mathbb{N}$ be an ordered set of clusters, such that $C_{i_1} \prec \dots \prec C_{i_{2t+1}}$. By

$$\mathfrak{C}'_{odd} = \{C_{i_{2s+1}} : s \in \{0, \dots, t\}\} \text{ and } \mathfrak{C}'_{even} = \{C_{i_{2s}} : s \in \{1, \dots, t\}\}, \quad (3.13)$$

we denote subsets of \mathfrak{C}' containing C_{i_j} indexed by odd and even numbers j respectively. Finally, we assume that, for each $j \in \{1, \dots, 2t\}$, the cluster C_{i_j} is a direct parent of $C_{i_{j+1}}$. In the sequel, each time when cluster C_p is a direct parent of cluster C_q , we use a short notation $C_p \in \tilde{\pi}(C_q)$ or $C_q \in \tilde{\sigma}(C_p)$.

In our framework, we employ the following families of valid inequalities:

(i) π - and σ -inequalities

$$\begin{aligned} x(\mathfrak{S} \setminus \pi(\mathfrak{S}), \bar{\mathfrak{S}} \setminus \pi(\mathfrak{S})) &\geq 1 \\ x(\bar{\mathfrak{S}} \setminus \sigma(\mathfrak{S}), \mathfrak{S} \setminus \sigma(\mathfrak{S})) &\geq 1 \end{aligned} \quad (\emptyset \neq \mathfrak{S} \subset \mathfrak{C} \setminus \{C_1\} \text{ and } \bar{\mathfrak{S}} = \mathfrak{C} \setminus \mathfrak{S});$$
(3.14)

(ii) (π, σ) -inequality

$$x(\mathfrak{S} \setminus \mathcal{Q}, \bar{\mathfrak{S}} \setminus \mathcal{Q}) \geq 1, \quad (3.15)$$

where $\mathcal{X}, \mathcal{Y} \subset \mathfrak{C} \setminus \{C_1\}$, $C' \in \mathcal{X}$, $C'' \in \mathcal{Y}$, $C' \prec C''$, $\mathcal{Q} = \{C_1\} \cup \pi(\mathcal{X}) \cup \sigma(\mathcal{Y})$, $\mathfrak{S} \subset \mathfrak{C}$, $\bar{\mathfrak{S}} = \mathfrak{C} \setminus \mathfrak{S}$, $\mathcal{X} \subseteq \mathfrak{S}$, and $\mathcal{Y} \subseteq \bar{\mathfrak{S}}$;

(iii) Precedence cycle breaking (PCB) inequalities

$$x(\mathfrak{S}, \bar{\mathfrak{S}}) \geq t + 1, \quad (3.16)$$

where $\mathfrak{S} \subset \mathfrak{C} \setminus \{C_1\}$, $\bar{\mathfrak{S}} = \mathfrak{C} \setminus \mathfrak{S}$, $\mathfrak{C}'_{\text{odd}} \subset \mathfrak{S}$, and $\mathfrak{C}'_{\text{even}} \subset \bar{\mathfrak{S}}$;

(iv) Strengthened PCB inequalities

$$x(\mathfrak{S} \setminus \mathfrak{S}', \bar{\mathfrak{S}} \setminus \mathfrak{S}') \geq t + 1, \quad (3.17)$$

where $\mathfrak{S} \subset \mathfrak{C} \setminus \{C_1\}$, $\bar{\mathfrak{S}} = \mathfrak{C} \setminus \mathfrak{S}$, $\mathfrak{C}'_{\text{odd}} \subset \mathfrak{S}$, $\mathfrak{C}'_{\text{even}} \subset \bar{\mathfrak{S}}$, $\tilde{\sigma}(C_{i_{2t+1}}) \not\subset \mathfrak{S}$, and $\mathfrak{S}' = \pi(C_{i_1}) \cup \sigma(C_{i_{2t+1}}) \setminus \tilde{\sigma}(C_{i_{2t+1}})$;

(v) Single-option inequalities

$$u_{ij} + u_{ji} + u_{kl} + u_{lk} \leq 1 \quad (C_k \in \pi(C_i), C_l \in \sigma(C_j)) \quad (3.18)$$

$$u_{ij} + u_{ji} + \sum_{C_l \in \sigma(C_j)} u_{kl} \leq 1 \quad (C_k \in \pi(C_i)) \quad (3.19)$$

$$u_{ij} + u_{ji} + \sum_{C_l \in \sigma(C_j)} u_{lk} \leq 1 \quad (C_k \in \pi(C_i)) \quad (3.20)$$

$$u_{ij} + u_{ji} + \sum_{C_k \in \pi(C_i)} u_{kl} \leq 1 \quad (C_l \in \sigma(C_j)) \quad (3.21)$$

$$u_{ij} + u_{ji} + \sum_{C_k \in \pi(C_i)} u_{lk} \leq 1 \quad (C_l \in \sigma(C_j)), \quad (3.22)$$

where $\{C_i, C_j\} \subset \mathfrak{C} \setminus \{C_1\}$.

The validity of all aforementioned inequalities and their numerical evaluation are presented in [Khachai et al., 2023] along with proofs of sufficient conditions for inequalities (3.14) to be facet-inducing, i.e. generating the most powerful cutting planes, which substantially improves the overall performance of the algorithm.

3.2.2 An illustrative example

In this section, we provide a detailed illustration of the proposed framework for instance \mathcal{I}_0^4 discussed in Case 4 of Section 3.1. We present each step of Algorithm 2, including the detailed reduction to the corresponding PCGTSP instance, finding its (sub)optimal solution, and reverse interpretation of the obtained PCGTSP solution in terms of the best (or close to the best) cutting path of the initial ECP instance.

In this example, the ECP instance is given by contours K_1 and K_2 complemented with triples $\{(\pi_j^k, \varepsilon_j^k, \theta_j^k) : k, j \in \{1, 2\}\}$ and a single bridge B_1 . All the air-motion costs $\delta_{j,i}^{k,l}$, $\gamma_{j,i}^{k,l}$, μ_j^k , and ν_j^k are specified by the

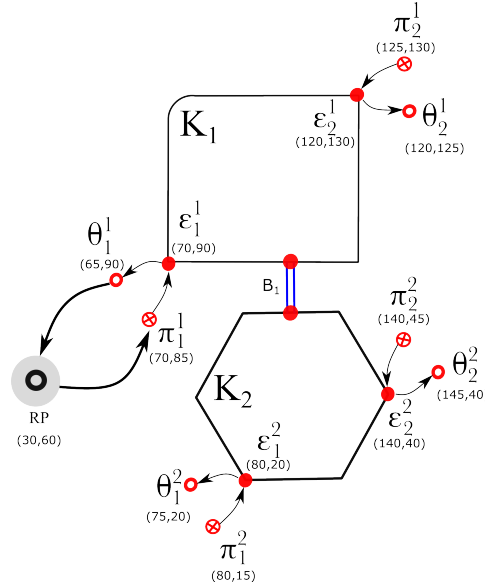


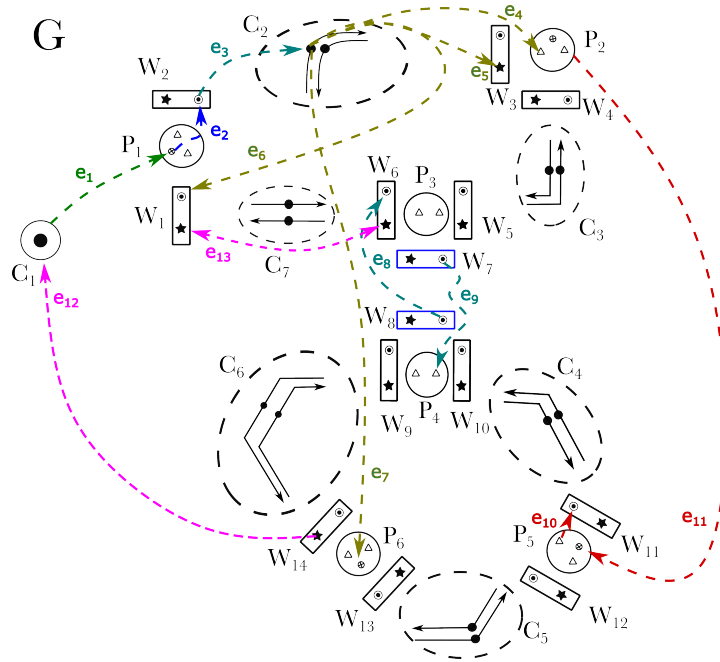
Figure 3.11: Example of the ECP with bridges

Euclidean distances between the appropriate points (Fig. 3.11). Piercing, lead-in, lead-out, and bridge costs are constant, $\rho_j^k = 50$, $\beta_i = 20$, $\sigma_j^k = \tau_j^k = 10$.

We construct the corresponding PCGTSP instance specified by (G, \mathfrak{C}, H) according to the rules of Case 4 in Section 3.1 (see Fig.3.10). In particular, the set of clusters \mathfrak{C} consists of all the contour clusters C_1, \dots, C_7 , including the depot cluster C_1 , crossroad clusters P_1, \dots, P_6 , and gateway clusters W_1, \dots, W_{14} . The graph of the partial order H turns to be bipartite and relates only crossroad clusters P_j with the associated gateway clusters W_k . According to the construction procedure, the set of arcs E of the graph G is partitioned into several disjoint subsets. The belonging of arc (u, v) to some subset is determined by node u .

In Fig. 3.12, we show typical representatives of these subsets:

(i) the unique node of the depot cluster C_1 representing the resting point is connected to each piercing node. In particular, arc e_1 connects the depot with an image of π_1^1 and has a cost $\mu_1^1 + \rho_1^1$;


 Figure 3.12: Specific arcs of the graph G

- (ii) each piercing node is connected with entry nodes of the associated gateway clusters charged by the appropriate lead-in cost, one of such arcs is e_2 of cost σ_1^1 ;
- (iii) each entry node has a zero-cost link (e.g. arc e_3) to the only appropriate element cluster node;
- (iv) each element cluster node has a zero-cost link to the pre-cut node of the same-direction crossroad cluster (e.g., arc e_4) and a cut-termination arc to the dummy node of the neighboring gateway, e.g., arc e_5 of cost $\tau_2^1 + \nu_2^1$. In addition, if this node belongs to the cluster non-adjacent with a bridge gadget (clusters C_2 and C_5 in our case) or it models the cutting motion away from such a gadget, it has two more types of outgoing arcs: to entry node of an arbitrary non-bridge foreign gateway (e.g., e_6 of the cost $\tau_2^1 + \gamma_{2,1}^{1,1}$) and to an arbitrary piercing node (as e_7 of the cost $\tau_2^1 + \delta_{2,1}^{1,2} + \rho_1^2$);
- (v) each entry node of a bridge gateway is connected with an entry node of a gateway neighboring to the opposite side of the bridge (for instance,

e_8), and with an appropriate pre-cut node of the bridge gadget (as arc e_9), both of the cost β_1 ;

(vi) each pre-cut node has a zero-cost link (as e_{10}) to an entry node of the associated gateway and links to each foreign piercing nodes (like e_{11} of cost $\tau_2^1 + \delta_{2,2}^{1,2} + \rho_2^2$);

(vii) finally, each dummy node has a zero-cost link to depot (e_{12} in our case). Furthermore, each pair of dummy nodes is connected by a bi-directional zero-cost link (like e_{13}).

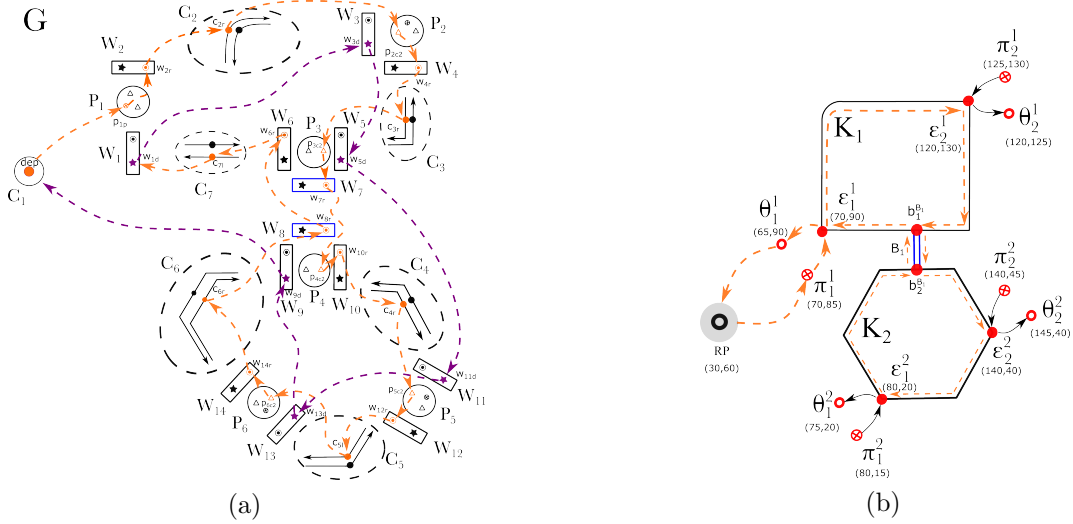


Figure 3.13: The obtained optimal PCGTSP solution (a) and the appropriate optimal cutting path of the same cost (b)

We apply the branch-and-cut algorithm from Subsection 3.2.1 and obtain an optimal solution of the constructed PCGTSP instance:

$$dep, p_{1p}, w_{2r}, c_{2r}, p_{2c2}, w_{4r}, c_{3r}, p_{3c2}, w_{7r}, p_{4c2}, w_{10r}, c_{4r}, p_{5c2}, w_{12r}, c_{5l}, p_{6c2}, \\ w_{14r}, c_{6r}, w_{8r}, w_{6r}, c_{7l}, w_{1d}, w_{3d}, w_{5d}, w_{11d}, w_{13d}, w_{9d}, dep$$

of cost 203. According to Proposition 16, this solution corresponds to an optimal cutting path for the initial ECP instance of the same cost:

$$RP, \pi_1^1, \varepsilon_1^1, \varepsilon_2^1, b_1^{B_1}, b_2^{B_1}, \varepsilon_2^2, \varepsilon_1^2, b_2^{B_1}, b_1^{B_1}, \varepsilon_1^1, \theta_1^1, RP.$$

The obtained cutting path is the output of Algorithm 2.

3.3 Computational Experiment

As it was proved in Section 3.1, various settings of the CPP can be efficiently reduced to appropriate auxiliary instances of the PCGTSP.

Following Section 3.2.1, a recent state-of-the-art branch-and-cut algorithm and PCGLNS metaheuristic were proposed for the PCGTSP. The purpose of this experiment is to evaluate the performance of PCGLNS against a variety of real-life industrial instances of the CPP, using the branch-and-cut algorithm as a baseline. We conclude our evaluation with statistical analysis of PCGLNS in order to demonstrate its efficiency.

3.3.1 Experimental setup

For this experiment, we collected 48 instances of the discrete CPP, which can be found on <https://github.com/EnsignDaniels/ CPP>. The instances are grouped by their size: small (≤ 27 clusters), medium (≤ 67 clusters), and large (≥ 100 clusters). The time limits for the PCGLNS heuristic for these groups were 60, 300, and 600 seconds, respectively. To assess the accuracy of the obtained solutions, we employ the branch-and-cut algorithm within the time limit of 40 hours. The tests were run on the 32-core Intel Xeon 128G RAM, AlmaLinux 8.5 server.

3.3.2 Results

In Table 3.1, we report for each instance, its size (m for number of clusters), optimum value (OPT) or the values of best lower (LB) and upper bounds (UB), and results obtained by PCGLNS up to three predefined time moments (15, 30, and 60 seconds for small instances; 20, 60, and 300 for medium ones; 25, 240, and 600 for the large ones). In column "PCGLNS",

we report the results found by the heuristic for each instance within the time limit, and the corresponding gap calculated as follows:

$$\text{gap} = \frac{\text{PCGLNS} - \text{LB}}{\text{LB}} \geq \frac{\text{PCGLNS} - \text{OPT}}{\text{OPT}} = \varepsilon,$$

where ε is the relative error of the obtained solution.

We can observe that PCGLNS was able to find the optimal solution for all except one small instances in 60 seconds, for the sub-optimal solution the gap was only 0.04%. For medium size instances, the heuristic found the optimum value for 4 instances out of 16 in 20 seconds, and for 5 more in the following 280 seconds.

In the case of large instances, relatively high gap results can be explained by the fact that branch-and-cut algorithm found only approximate solutions within its time limit. Therefore, to assess the accuracy of solutions obtained by PCGLNS, we rely on the best lower bounds. Nevertheless, we believe that PCGLNS still performs well in this case because the best solution of the branch-and-cut is found by PCGLNS as its only primal heuristic.

Table 3.2 reports the distribution of gap values of the solutions obtained by the heuristic for small, medium and large instances for the second experiment, where PCGLNS was applied 10 times for each instance. The first column of this table reports the percentage of PCGLNS runs with the corresponding accuracy. For instance, with probability at least 60%, the heuristic provided an optimal solution for a medium-sized instance in 60 seconds, and with probability at least 80%, the gap was inferior to 4.09% after 300 seconds. In a few instances, the PCGLNS managed to find feasible solutions after 60 seconds. Thus, we highlight these cases by ‘?’.

Table 3.1: PCGLNS performance results

small instances								
instance	m	OPT or [LB, UB]	time (sec)					
			15		30		60	
			PCGLNS	gap (%)	PCGLNS	gap (%)	PCGLNS	gap (%)
elx_6_m	17	1513	1514	0.07	1513	0	1513	0
elx_9_m	17	2926	2926	0	2926	0	2926	0
elx_10_m	18	2587	2588	0.04	2587	0	2587	0
e3x_2_m	18	1578	1584	0.38	1578	0	1578	0
elx_8_m	19	1716	1716	0	1716	0	1716	0
elx_12_m	19	4530	4746	4.77	4530	0	4530	0
elx_7_m	21	1732	1732	0	1732	0	1732	0
e3x_1_m	21	1176	1176	0	1176	0	1176	0
elx_1_m	22	2866	2866	0	2866	0	2866	0
elx_5_m	23	1588	1590	0.13	1588	0	1588	0
e5x_1_m	23	1847	1847	0	1847	0	1847	0
elx_3_m	24	2290	2293	0.13	2291	0.04	2291	0.04
elx_13_m	24	2168	2168	0	2168	0	2168	0
elx_11_m	27	3242	3242	0	3242	0	3242	0
elx_14_m	27	1958	1977	0.97	1968	0.51	1958	0
elx_15_m	27	2751	2752	0.04	2751	0	2751	0
medium instances								
instance	m	OPT or [LB, UB]	time (sec)					
			20		60		300	
			PCGLNS	gap (%)	PCGLNS	gap (%)	PCGLNS	gap (%)
elx_2_m	29	3556	3556	0	3556	0	3556	0
plxj_21	31	3356	3369	0.39	3356	0	3356	0
plxj_22	31	3299	3309	0.3	3309	0.3	3309	0.3
ps-33-m	34	5764	5764	0	5764	0	5764	0
ps-34-m	35	[5815, 6008]	6008	3.32	6008	3.32	6008	3.32
ps34-2-m	35	4951	4951	0	4951	0	4951	0
ps-36-m	37	2012	n/a	n/a	n/a	n/a	2012	0
P1xl_23	45	3778	3778	0	3778	0	3778	0
pm-47-m	48	[5948, 6189]	6191	4.09	6191	4.09	6191	4.09
pm-53-m	55	[6338, 6864]	6888	8.68	6878	8.52	6864	8.3
p7xj_9-m	59	5191	5191	0	5191	0	5191	0
plxj_24	62	[9730, 10581]	10608	9.02	10581	8.75	10581	8.75
p7xj_6-m	63	6001	6006	0.08	6001	0	6001	0
pm-62-m	63	7907	8096	2.39	7907	0	7907	0
pm65-m	66	[3219, 3297]	3352	4.13	3317	3.04	3297	2.42
p7xj_7-m	67	[6499, 7056]	7056	8.57	7056	8.57	7056	8.57
large instances								
instance	m	OPT or [LB, UB]	time (sec)					
			25		240		600	
			PCGLNS	gap (%)	PCGLNS	gap (%)	PCGLNS	gap (%)
P5XJ_9-n-m	68	[6630, 7106]	7137	7.65	7106	7.18	7106	7.18
p7xj_5-m	69	[8311, 8454]	8682	4.46	8454	1.72	8454	1.72
PL71-m	72	[9632, 10379]	10426	8.24	10379	7.76	10379	7.76
pl-75-m	76	[7276, 7498]	7498	3.05	7498	3.05	7498	3.05
pl-76-m	77	4062	4062	0	4062	0	4062	0
p7xj_8-m	79	7548	7548	0	7548	0	7548	0
ph91-m	79	[7481, 7895]	7895	5.53	7895	5.53	7895	5.53
pl-89	90	[5920, 6464]	6464	9.19	6464	9.19	6464	9.19
ph92	93	[8423, 8602]	8718	3.5	8701	3.3	8701	3.3
ph-99-m	100	14206	14206	0	14206	0	14206	0
ph-99	100	[7862, 8574]	8574	9.06	8574	9.06	8574	9.06
ph-102-m	103	[6034, 6604]	6604	9.45	6604	9.45	6604	9.45
ph-103-m	104	[13904, 15138]	15840	13.92	15138	8.88	15138	8.88
ph-104-m	105	[7682, 7943]	8386	9.16	7943	3.4	7943	3.4
ph-110-m	111	[18088, 20944]	21167	17.02	21104	16.67	20944	15.79
ph-119-m	120	[6773, 6816]	6921	2.19	6921	2.19	6816	0.63

Table 3.2: PCGLNS gap percentiles for test instances

Small instances gaps (%)				Medium instances gaps (%)				Large instances gaps (%)			
α level (%)	time			α level (%)	time			α level (%)	time		
	15 sec	30 sec	60 sec		20 sec	60 sec	300 sec		25 sec	240 sec	600 sec
10	0.00	0.00	0.00	10	0.00	0.00	0.00	10	0.00	0.00	0.00
20	0.00	0.00	0.00	20	0.00	0.00	0.00	20	0.00	0.00	0.00
30	0.00	0.00	0.00	30	0.00	0.00	0.00	30	3.05	2.19	2.19
40	0.00	0.00	0.00	40	0.00	0.00	0.00	40	3.50	3.05	3.05
50	0.00	0.00	0.00	50	0.00	0.00	0.00	50	5.53	5.53	3.40
60	0.04	0.00	0.00	60	3.32	0.00	0.00	60	8.24	7.65	7.65
70	0.07	0.00	0.00	70	4.09	3.32	3.32	70	9.06	7.76	7.76
80	0.13	0.00	0.00	80	8.57	8.46	4.09	80	9.19	9.06	9.06
90	0.97	0.00	0.00	90	8.68	8.57	8.30	90	9.45	9.19	9.19
100	4.77	0.51	0.04	100	-	-	8.75	100	17.02	16.67	15.79

To summarize, PCGLNS provided good results in terms of solution time and quality for the majority of the industrial instances of the CPP showing its high potential for practical applications.

Chapter 4

Capacitated Vehicle Routing

Problem: Efficient approximation in metric spaces of a fixed doubling dimension

4.1 Problem statement

The Capacitated Vehicle Routing Problem (CVRP) can be formulated informally as follows. We are given by a set of customers X , each of them has a unit demand on some homogeneous commodity. All the customer's demand should be serviced by identical vehicles of a fixed capacity q located initially at the given depot y . The problem is to construct a minimum cost family of cyclic routes servicing the total customer demand, each of them departs from and arrives at the depot y and satisfies the capacity constraint.

For the sake of convenience, we give a mathematical statement of a slightly more general problem, where each customer is free to have a non-unit integer demand, which can be split between several routes. In the literature, this problem is referred to as the Capacitated Vehicle Routing Problem with Splittable Demand (CVRP-SD).

An instance of the CVRP-SD is given by a complete weighted graph $G = (X \cup \{y\}, E, D, w)$ and a natural number q . Here, $X = \{x_1, \dots, x_n\}$ is a set of customers, y is a depot, the non-negative weighting function $D : X \rightarrow \mathbb{Z}_+$ specifies customer demand, the symmetric weighting function $w : E \rightarrow \mathbb{R}_+$, to any couple of nodes $\{u, v\} \subset X \cup \{y\}$, assigns the transportation cost $w(u, v)$ related to the direct transition along the edge $\{u, v\} \in E$, and q is an upper vehicle capacity bound.

A *route* is an ordered pair $\mathcal{R} = (\pi, S_{\mathcal{R}})$, such that $\pi = y, x_{i_1}, \dots, x_{i_t}, y$ is a cycle in the graph G and the function $S_{\mathcal{R}} : X \rightarrow \mathbb{Z}_+$ defines a distribution of the serviced customer demand. For the route \mathcal{R} , its *cost* $w(\mathcal{R})$ is defined as follows

$$w(\mathcal{R}) = w(y, x_{i_1}) + w(x_{i_1}, x_{i_2}) + \dots + w(x_{i_{t-1}}, x_{i_t}) + w(x_{i_t}, y).$$

The route \mathcal{R} is called *feasible*, if

$$S_{\mathcal{R}}(x) \begin{cases} \leq D(x) & \text{for any } x \in \{x_{i_1}, \dots, x_{i_t}\}, \\ = 0, & \text{otherwise} \end{cases} \quad \text{and} \quad \sum_{x \in X} S_{\mathcal{R}}(x) \leq q.$$

An arbitrary family \mathfrak{S} of feasible routes is called a *feasible solution* of the problem, if it services the total customer demand

$$\sum_{\mathcal{R} \in \mathfrak{S}} S_{\mathcal{R}}(x) = D(x) \quad (x \in X).$$

To any feasible solution \mathfrak{S} , we assign its *cost* $w(\mathfrak{S}) = \sum_{\mathcal{R} \in \mathfrak{S}} w(\mathcal{R})$. Thus, the goal is to construct the cheapest feasible solution \mathfrak{S} , i.e.

$$\begin{aligned} w(\mathfrak{S}) &\rightarrow \min \\ \text{s.t.} \quad &\sum_{\mathcal{R} \in \mathfrak{S}} S_{\mathcal{R}}(x) = D(x) \quad (x \in X). \end{aligned} \tag{4.1}$$

Obviously, the statement of the classic CVRP can be obtained by restriction of the above setting with the additional constraint $D(x) \equiv 1$.

If the function w satisfies the triangle inequality, i.e. $w(v_1, v_2) \leq w(v_1, v_3) + w(v_3, v_2)$ holds for any subset $\{v_1, v_2, v_3\} \subset X \cup \{y\}$, the instance of CVRP is called *metric*. In this case, nodes of the graph G are called *points*, $w(u, v)$ is referred to as a *distance* between the points u and v , and the cost $w(\mathcal{R})$ of an arbitrary route \mathcal{R} is called its *length*.

In this chapter, we consider the metric CVRP restricted as follows:

- (i) the ordered pair (Z, w) , where $Z = X \cup \{y\}$, is a finite metric space of a fixed doubling dimension $d > 1$;
- (ii) the vehicle capacity bound q does not exceed $\text{polylog}(n)$.

Hereinafter, we use the notation $\text{CVRP}(Z, w, q)$ and $\text{CVRP}^*(Z, w, q)$ for the instance specified by the graph $G = (X \cup \{y\}, E, w)$ and capacity q and its optimum value, respectively¹.

4.2 Metric spaces of a fixed doubling dimension

For the subsequent constructions, we need to recall some definitions and preliminary technical results.

Suppose we are given by some metric space (Z, w) . For any $z_0 \in Z$ and a number $R \geq 0$, the set $B(z_0, R) = \{z \in Z : w(z_0, z) \leq R\}$ is called a *metric ball* of a radius R centered at the point $z_0 \in Z$.

Definition 1 (see, e.g. [Abraham et al. \[2011\]](#)). *For a number $d > 1$, the space (Z, w) is referred to as a metric space of the fixed dimension d , if, for an arbitrary $z_0 \in Z$ and $R > 0$, there exist points $z_1, \dots, z_M \in Z$, such that*

$$B(z_0, R) \subseteq \bigcup_{j=1}^M B(z_j, R/2) \text{ and } M \leq 2^d.$$

¹and the notation $\text{CVRP-SD}(Z, D, w, q)$ and $\text{CVRP-SD}^*(Z, D, w, q)$ for the case of CVRP-SD as well

It is easy to verify that, for any $d \geq 1$ and $p \geq 1$, the space l_p^d is a metric space of doubling dimension $O(d)$. On the other hand, there are known many metrics of a fixed dimension that appear to be very far from the finite-dimensional numeric spaces [see, e.g. [Gupta et al., 2003](#)].

Next, let $Z' \subset Z$ be an arbitrary nonempty subspace of the space Z (of doubling dimension d). By $\Delta = \Delta_w(Z') = \sup\{w(u, v) : u, v \in Z'\}$ and $\alpha = \alpha_w(Z') = \inf\{w(u, v) : \{u, v\} \subset Z'\}$ we denote an upper and a lower bounds for the distances between the distinct points in Z' , respectively.

Lemma 4 ([Talwar \[2004\]](#)). *Let $0 < \alpha \leq \Delta < \infty$. Then, the subspace Z' is finite and*

$$|Z'| \leq \left(\frac{2\Delta}{\alpha}\right)^d.$$

In this chapter, we restrict ourselves to finite metric spaces induced by complete weighted graphs $G = (Z, E, w)$. Let, further, $U \subset Z$ be an arbitrary nonempty node subset of the graph G , $\text{MST}(U)$ be the minimum spanning tree for the induced subgraph $G\langle U \rangle$, and $R = R(U)$ be a radius of the minimal ball (centered at some point $z \in Z$) enclosing the subset U .

The following known *small spanning trees* lemma [see, e.g. [Talwar, 2004](#), [Smid, 2010](#)] gives a non-trivial upper bound for the cost $w(\text{MST}(U))$. For the sake of completeness, we provide this result with a proof.

Lemma 5.

$$w(\text{MST}(U)) \leq 12R \cdot |U|^{1-1/d}. \quad (4.2)$$

Proof. Following to [[Smid, 2010](#)], we show that the following equation

$$w(\text{MST}(V)) \leq 12R(V)(|V|^{1-1/d} - 1) \quad (4.3)$$

is valid for an arbitrary $\emptyset \neq V \subseteq U$. We construct our argument by induction on $|V|$.

Base case: $|V| \leq 2$. If $|V| = 1$, equation (4.3) follows from

$$R(V) = w(\text{MST}(V)) = 0.$$

In the case $|V| = \{u, v\}$,

$$w(\text{MST}(V)) = w(u, v) \leq 2R(V) < 4.8R(V) < 12R(V)(|V|^{1-1/d} - 1)$$

holds, since $2^{1-1/d} - 1 \geq \sqrt{2} - 1 > 0.4$ for an arbitrary $d \geq 2$.

Inductive step: Let, further, $|V| \geq 3$. By induction hypothesis, property (4.3) is valid for an arbitrary non-empty subset $V' \subset V$, $|V'| < |V|$. Now, we proceed with the subset V .

Let B be the minimal metric ball of radius $R(V)$ enclosing the subset V . By definition, for some $l \leq 2^d$, there exist balls $B_1, \dots, B_l \subset Z$ of radius $R(V)/2$, such that $\cup_{j=1}^l B_j \supseteq B$.

Defining $V_j = B_j \cap V$, without loss of generality, we can assume that, for any j and $k \neq j$, $V_j \neq \emptyset$ and $V_j \cap V_k = \emptyset$. Furthermore, we can always assume that $l \geq 3$. Indeed, $l > 1$ due to the minimality of the ball B . If $l = 2$, then at least one of the subsets V_1 or V_2 , e.g. V_1 , is not a singleton and can be separated into two nonempty subsets, since $|V| \geq 3$.

Next, for each V_j , the inequality $R(V_j) \leq R(V)/2$ holds by construction. Therefore,

$$w(\text{MST}(V_j)) \leq 12R(V_j)(|V_j|^{1-1/d} - 1) \leq 6R(V)(|V_j|^{1-1/d} - 1),$$

by the induction hypothesis.

Let, further, $H = \{v_j \in V_j : j \in \{1, \dots, l\}\}$ be an arbitrary hitting set for the subsets V_1, \dots, V_l . Consider a tree

$$T = \text{MST}(H) \cup \bigcup_{j=1}^l \text{MST}(V_j).$$

By construction, $w(\text{MST}(H)) \leq 2R(V)(l-1)$, since the diameter $\Delta(V) \leq 2R(V)$. Combining the bounds, we obtain

$$\begin{aligned} w(T) &= \sum_{j=1}^l w(\text{MST}(V_j)) + w(\text{MST}(H)) < 6R(V) \sum_{j=1}^l (|V_j|^{1-1/d} - 1) + 2l \cdot R(V) \\ &\leq 6R(V) \sum_{j=1}^l |V_j|^{1-1/d} - 4l \cdot R(V) \leq 6R(V) \sum_{j=1}^l \left(\frac{|V|}{l}\right)^{1-1/d} - 4l \cdot R(V) \end{aligned} \quad (4.4)$$

$$= 6R(V)l^{1/d}|V|^{1-1/d} - 4l \cdot R(V) \leq 12R(V) \cdot |V|^{1-1/d} - 12R(V), \quad (4.5)$$

where inequalities (4.4) and (4.5) follow from the concavity of the function $f(x) = x^{1-1/d}$ and the inequalities $l \leq 2^d$ and $l \geq 3$, respectively.

Thus, the inductive step is proved. To complete the proof of Lemma 5, just consider the case $V = U$. \square

4.3 Extended Das and Mathieu approximation scheme

In this section, we show that the well-known QPTAS proposed by A. Das and C. Mathieu [Das and Mathieu, 2015] for the Euclidean CVRP can be extended to the case of metric spaces of any fixed doubling dimension $d > 1$. Supplementing the main idea of their scheme with the technical results underlying the recent PTAS of Y. Bartal et al. [Bartal et al., 2016] for the metric TSP formulated in such spaces, we propose an algorithm that, for an arbitrary $0 < \varepsilon < 1/8$ finds a $(1 + O(\varepsilon))$ -approximate solution of the CVRP in a metric space of any doubling dimension $d > 1$. On the other hand, we show that the resulting algorithm, generally speaking, ceases to be a QPTAS, even for a fixed capacity q . Further, in Subsection 4.3.5, we propose a novel version of Das and Mathieu scheme, whose time complexity is quasi-polynomial provided $q = O(\text{polylog}(n))$.

Similarly to the original scheme, our algorithm consists of several consecutive stages, as follows:

- (i) **Preliminary processing and accuracy driven rounding.** At this stage, given by $\varepsilon > 0$, to the instance in question, we assign an auxiliary instance of more simple structure, called *rounded*, such that an arbitrary $(1 + \varepsilon)$ -approximate solution of this instance can be transformed efficiently to the appropriate $(1 + O(\varepsilon))$ -approximate solution of the initial problem.
- (ii) **Randomized hierarchical clustering.** Given by values of random parameters, at this stage, we construct a number of mutually nested partitions of the set $X \cup \{y\}$. Then, in each cluster located at any level of the resulting hierarchy, we point out some number of special points (we call them *portals*). Following to the approach proposed in [Talwar, 2004], we show that, for any rounded instance, there exist $(1 + \varepsilon)$ -approximate solutions, each their route crosses any cluster at most r times (for some number r , which will be defined later) and at portals exclusively. Such routes are referred to as *net-respecting and r -light* [see, e.g. Bartal et al., 2016].
- (iii) **Dynamic Programming and Iterated Tour Partition.** At this stage, following to [Das and Mathieu, 2015], we allow some routes of the constructed solutions (we call them *relaxed*) slightly violate the capacity constraint. Then, to obtain a required feasible approximate solution,
 - (a) we apply dynamic programming to find a relaxed net-respecting and r -light solution minimizing some specially penalized objective function
 - (b) applying a randomized rank procedure for the demand covered by the routes of the solution obtained, we ensure that each route

covers at most q demand units of the highest rank; following to [Das and Mathieu, 2015], we call such units *black*

- (c) all other units (we call them *red*) are excluded from these routes and covered separately, by the additional routes constructed using the well-known Iterated Tour Partition (ITP) heuristic [Haimovich and Rinnooy Kan, 1985]
- (d) thus, we obtain two partial solutions \mathfrak{S}_{black} and \mathfrak{S}_{red} , such that their combination is a feasible solution of the problem in question.

Finally, we show that the expected cost of this combined solution over random clustering and demand ranking fulfils the following equation

$$\mathbb{E}(w(\mathfrak{S}_{black}) + w(\mathfrak{S}_{red})) = (1 + O(\varepsilon)) \cdot \text{CVRP}^*(Z, w, q).$$

- (iv) **Derandomization.** Relying on the arguments from [Das and Mathieu, 2015] and [Talwar, 2004], we show that the proposed algorithm admits polynomial time derandomization.

4.3.1 Accuracy driven rounding

This stage dates back to the classic PTAS proposed by S. Arora for the Euclidean TSP [Arora, 1998]. As above, let $\Delta = \Delta_w(Z) = \max\{w(u, v) : u, v \in Z = X \cup \{y\}\}$ be the diameter of the set Z . Without loss of generality, we assume that $\Delta = n/\varepsilon$. Indeed, otherwise, to the initial instance $\text{CVRP}(Z, E, w)$, we can easily assign an equivalent (in terms of optimality sets) scaled instance $\text{CVRP}(Z, E, w')$ with the following weighting function: $w'(u, v) = w(u, v) \cdot \frac{n}{\varepsilon \cdot \Delta}$.

We define the desired *rounded* instance in terms of metric nets.

Definition 2. A subset $N \subseteq Z$ is called a δ -net in the metric space (Z, w) for some given $\delta > 0$, if the following conditions holds

- (i) for any $u \in Z$, there exists $v = v(u) \in N$, such that $w(u, v) \leq \delta$;
- (ii) for an arbitrary distinct points $v_1, v_2 \in N$, the distance $w(v_1, v_2) > \delta$.

Let $N_1 = \{\xi_1, \dots, \xi_J\}$ be an arbitrary 1-net of the set Z . We assign to the initial instance $\text{CVRP}(Z, w, q)$ the rounded one $\text{CVRP-SD}(N_1, D, w_1, q)$ as follows:

- (i) breaking tights arbitrarily, we define a mapping $\xi: Z \rightarrow N_1$ such that $w(z, \xi(z)) \leq 1$ holds for any $z \in Z$;
- (ii) we exclude all the customers associated with the node $\eta = \xi(y)$;
- (iii) as a result, to any node $\xi_j \in N_1 \setminus \{\eta\}$, we assign the accumulated customer demand

$$D(\xi_j) = \begin{cases} |\xi^{-1}(\xi_j)|, & \xi_j \neq \eta, \\ 0, & \text{otherwise,} \end{cases}$$

- (iv) as new weighting function w_1 , we take a restriction $w|_{N_1}$ of the function w to the set $N_1 \subset Z$.

Lemma 6 establishes a close relation between optimum values of the initial and rounded instances.

Lemma 6.

$$\text{CVRP}^*(Z, w, q) - 7n \leq \text{CVRP-SD}^*(N_1, D, w_1, q) \leq \text{CVRP}^*(Z, w, q) + 4n.$$

Proof. I. To prove the upper bound we consider an arbitrary optimal solution $\mathfrak{S} = \{\mathcal{R}\}$ of the given instance $\text{CVRP}(Z, w, q)$. To each route $\mathcal{R} = (\pi, S_{\mathcal{R}}) \in \mathfrak{S}$, $\pi = y, x_{i_1}, \dots, x_{i_t}, y$ we assign the *rounded* route $\bar{\mathcal{R}} = (\bar{\pi}, S_{\bar{\mathcal{R}}})$ (see Fig. 4.1a), where $\bar{\pi} = \eta, \xi(x_{i_1}), \dots, \xi(x_{i_t}), \eta$ and distribution

$S_{\bar{\mathcal{R}}}: N_1 \rightarrow \mathbb{Z}_+$ is defined as follows

$$S_{\bar{\mathcal{R}}}(\xi_j) = \begin{cases} \sum_{x: \xi(x)=\xi_j} S_{\mathcal{R}}(x), & \xi_j \neq \eta \\ 0, & \text{otherwise.} \end{cases}$$

Without loss of generality, we assume that all $\xi(x_{i_j})$ are distinct and do not coincide with η .

By construction, the obtained route family $\bar{\mathfrak{S}} = \{\bar{\mathcal{R}}\}$ is a feasible solution of the rounded instance CVRP-SD(N_1, D, w_1, q). Estimate its cost $w_1(\bar{\mathfrak{S}}) = \sum_{\bar{\mathcal{R}} \in \bar{\mathfrak{S}}} w_1(\bar{\mathcal{R}})$. By selection of the weighting function w_1 and mapping ξ and by the triangle inequality, we obtain

$$\begin{aligned} w_1(\bar{\mathcal{R}}) &= w_1(\eta, \xi(x_{i_1})) + \sum_{j=1}^{t-1} w_1(\xi(x_{i_j}), \xi(x_{i_{j+1}})) + w_1(\xi(x_{i_t}), \eta) \\ &\leq w(y, x_{i_1}) + \sum_{j=1}^{t-1} w(x_{i_j}, x_{i_{j+1}}) + w(x_{i_t}, y) + 2 \sum_{j=1}^t w(x_{i_j}, \xi(x_{i_j})) + 2w(y, \eta) \\ &\leq w(\mathcal{R}) + 2t + 2. \end{aligned}$$

Hence,

$$\begin{aligned} \text{CVRP-SD}^*(N_1, D, w_1, q) &\leq w_1(\bar{\mathfrak{S}}) = \sum w_1(\bar{\mathcal{R}}) \leq \sum w(\mathcal{R}) + 2n + 2n \\ &\leq w(\mathfrak{S}) + 4n = \text{CVRP}^*(Z, w, q) + 4n, \end{aligned}$$

since an arbitrary customer $x \in X$ is being visited by exactly one route \mathcal{R} of the optimal solution \mathfrak{S} .

II. Proof of the lower bound can be obtained in a similar way. We fix an arbitrary optimal solution $\bar{\mathfrak{S}} = \{\bar{\mathcal{R}}_1, \dots, \bar{\mathcal{R}}_K\}$ of CVRP-SD(N_1, D, w_1, q). By definition, the demand satisfied by the route $\bar{\mathcal{R}} = (\bar{\pi}, S_{\bar{\mathcal{R}}}) \in \bar{\mathfrak{S}}$ in an arbitrary node ξ of the net N_1 is defined by $S_{\bar{\mathcal{R}}}(\xi)$ such that

$$\sum_{\xi \in N_1} S_{\bar{\mathcal{R}}}(\xi) \leq q.$$

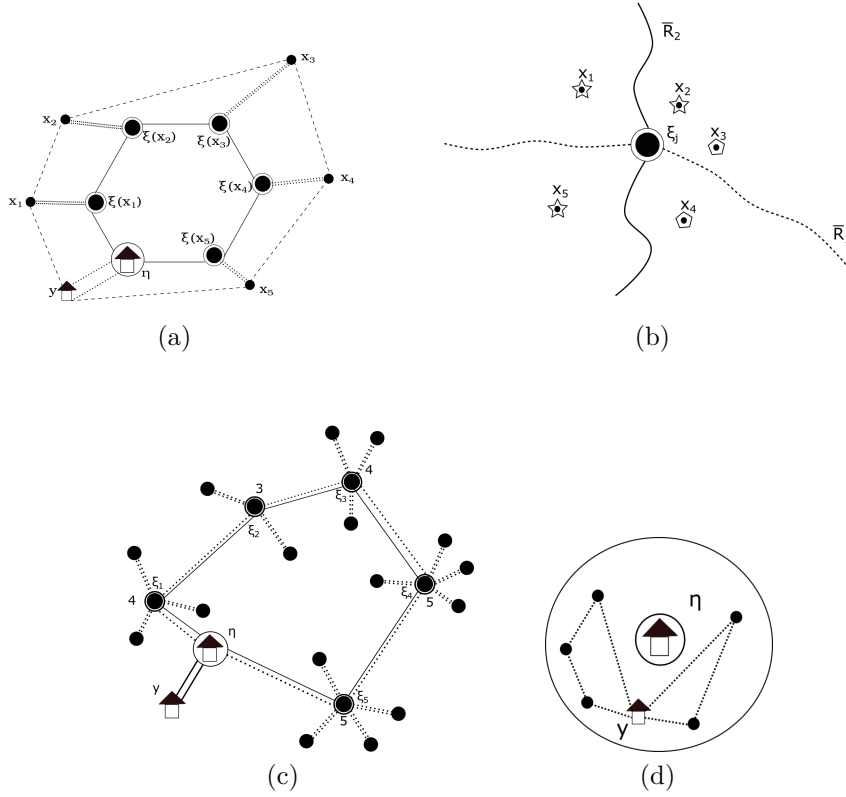


Figure 4.1: Proof of Lemma 6

To an arbitrary $\xi \in N_1$ except η , we introduce the mapping $\zeta_j: X_j \rightarrow \bar{\mathfrak{S}}$ assigning to any customer $x \in X_j$ the rounded route $\bar{\mathcal{R}}$ that fulfils its demand (see Fig 4.1b). By construction,

$$|\zeta_j^{-1}(\bar{\mathcal{R}})| = S_{\bar{\mathcal{R}}}(\xi_j), \quad (1 \leq j \leq J, \bar{\mathcal{R}} \in \bar{\mathfrak{S}}).$$

Next, to any route $\bar{\mathcal{R}} = (\bar{\pi}, S_{\bar{\mathcal{R}}}) \in \bar{\mathfrak{S}}$, $\bar{\pi} = \eta, \xi_{j_1}, \dots, \xi_{j_t}, \eta$ we assign the *unrounded* route $\mathcal{R} = \mathcal{R}(\bar{\mathcal{R}})$ (see Fig. 4.1c) departing from (and arriving to) the depot y and consecutively visiting all the nodes of subsets $\zeta_{j_1}^{-1}(\bar{\mathcal{R}}), \dots, \zeta_{j_t}^{-1}(\bar{\mathcal{R}})$ (for each subset $\zeta_j^{-1}(\bar{\mathcal{R}})$, all its nodes can be visited by an arbitrary order). Estimate the cost of the family of such routes $\mathfrak{S} = \{\mathcal{R}_1, \dots, \mathcal{R}_K\}$. By construction of the net N_1 and due to the triangle

inequality, we obtain

$$w(\mathcal{R}_k) \leq w_1(\bar{\mathcal{R}}_k) + 2 \sum_{i=1}^t S_{\bar{\mathcal{R}}_k}(\xi_{j_i}) + 2,$$

hence,

$$w(\mathfrak{S}) = \sum_{k=1}^K w(\mathcal{R}_k) \leq \sum_{k=1}^K w_1(\bar{\mathcal{R}}_k) + 2 \sum_{k=1}^K \sum_{j=1}^J S_{\bar{\mathcal{R}}_k}(\xi_j) + 2K \leq w_1(\bar{\mathfrak{S}}) + 4n,$$

since $\sum_{k=1}^K \sum_{j=1}^J S_{\bar{\mathcal{R}}_k}(\xi_j) \leq n$ and $K \leq n$.

Notice that the family of routes $\mathfrak{S} = \{\mathcal{R}_1, \dots, \mathcal{R}_K\}$ is not necessarily a feasible solution of CVRP(Z, w, b) because all the customers assigned to η are excluded from the instance CVRP-SD(N_1, D, w_1, q). To make the family \mathfrak{S} feasible, we service these customers by at most $\lceil \frac{n}{q} \rceil$ dedicated routes \mathfrak{S}' , such that all of them except maybe one visit q customers exactly (see Fig. 4.1d). Thus, it is easy to verify that their total cost is at most $3n$. Therefore, we obtain the total upper bound for CVRP*(Z, w, q) as follows

$$\text{CVRP}^*(Z, w, q) \leq w(\mathfrak{S} \cup \mathfrak{S}') \leq \text{CVRP-SD}^*(N_1, D, w_1, q) + 7n. \quad (4.6)$$

Lemma 6 is proved. \square

Notice that all the aforementioned procedures, i.e. construction of the net N_1 , assigning to the initial CVRP(Z, w, q) its rounded instance CVRP-SD(N_1, D, w_1, q), and reconstruction the solution $\mathfrak{S} \cup \mathfrak{S}'$ associated to the (rounded) solution $\bar{\mathfrak{S}}$, can be done in polynomial time.

As a simple corollary, we show that an arbitrary approximate solution of CVRP-SD(N_1, D, w_1, q) corresponds to the suitable approximate solution of the initial CVRP(Z, w, q).

Corollary 1. *For any $\varepsilon > 0$, an $(1 + \varepsilon)$ -approximate solution of the rounded instance CVRP-SD(N_1, D, w_1, q) can be transformed efficiently to an appropriate $(1 + O(\varepsilon))$ -approximate solution of CVRP(Z, w, q).*

Proof. By construction, $\mathfrak{S} \cup \mathfrak{S}'$ is a feasible solution of $\text{CVRP}(Z, w, q)$, whose cost $w(\mathfrak{S} \cup \mathfrak{S}')$ is defined by (4.6). Taking into account that

$$w_1(\bar{\mathfrak{S}}) \leq (1 + \varepsilon)\text{CVRP-SD}^*(N_1, D, w_1, q) \quad \text{and} \quad \Delta_w(Z) = n/\varepsilon,$$

we obtain

$$\begin{aligned} w(\mathfrak{S} \cup \mathfrak{S}') &\leq (1 + \varepsilon)\text{CVRP-SD}^*(N_1, D, w_1, q) + 7n \\ &\leq (1 + \varepsilon)(\text{CVRP}^*(Z, w, q) + 4n) + 7n \leq (1 + \varepsilon)\text{CVRP}^*(Z, w, q) + 2\varepsilon\Delta_w(Z)(7/2 + 2\varepsilon) \\ &= (1 + O(\varepsilon))\text{CVRP}^*(Z, w, q), \end{aligned}$$

since the triangle inequality obviously implies $2\Delta_w(Z) \leq \text{CVRP}^*(Z, w, q)$. Corollary 1 is proved. \square

Thus, in the sequel, without loss of generality, we assume that we are given by a rounded instance.

4.3.2 Randomized hierarchical clustering

Following to [5], we fix a number $s \geq 6$ and put $L = \lceil \log_s \Delta_w(Z) \rceil = O(\log n - \log \varepsilon)$. Then, for each $l = 0, 1, \dots, L + 1$, we fix an arbitrary s^{L-l} -net $N(l)$ of the set Z . Without loss of generality, assume that $N(l) \subset N(l + 1)$ for any $0 \leq l \leq L$. Notice, that the net $N(L + 1) = Z$, whilst the net $N(0)$ is a singleton.

In the following, we construct a randomized hierarchical clustering of Z by induction on level $l = 0, \dots, L + 1$ as proposed in the paper [Talwar, 2004].

We start with level $l = 0$, where we have a single cluster C_1^0 . Further, let $Z = C_1^l \cup C_2^l \dots \cup C_K^l$ be a clustering at the level $l < L$. To proceed with the clustering at level $l + 1$, we partition each cluster C_j^l separately, applying the following simple procedure

- (i) pick a random permutation σ of the $s^{L-(l+1)}$ -net $N(l+1) = \{h_1, \dots, h_{t_{l+1}}\}$;

- (ii) to an arbitrary $h_{\sigma(i)} \in N(l + 1)$, assign a number $\mu = \mu_i$ picked at random from $[1, 2)$;
- (iii) define a subset C_{ji}^{l+1} by the formula

$$C_{ji}^{l+1} = B(h_{\sigma(i)}, \mu \cdot s^{L-(l+1)}) \cap C_j^l \setminus \bigcup_{k=1}^{i-1} C_{jk}^{l+1};$$

- (iv) construct a partition of the cluster C_j^l from all non-empty subsets C_{ji}^{l+1} .

Finally, we obtain the resulting clustering of the set Z at level $l + 1$ by combining individual partitions for all clusters C_j^l .

By construction, at level $L + 1$, all the clusters are singletons, while, at level $l = 0$, we have the only cluster C_1^0 . Thus, the total number of clusters is at most $(n + 1) \cdot (L + 1) = O(n(\log n - \log \varepsilon))$.

For the further constructions, we need to introduce a special type of routes.

Definition 3. A route $\mathcal{R} = (\pi, S_{\mathcal{R}})$ is called *net-respecting relatively to a given hierarchy $N(l), l = 0, 1, \dots, L + 1$ and value $\varepsilon > 0$* , if, for any edge $\{u, v\}$ of the cycle π , there are two possible options

(i) $w(u, v) < 1/\varepsilon$

(ii) $s^{L-l} \leq \varepsilon \cdot w(u, v) < s^{L-l+1}$ for some $0 \leq l \leq L$ and both endpoints u and v belong to the net $N(l)$.

Obviously, the aforementioned options appears to be quite similar to each other, since in case (i), $u, v \in N(L + 1)$ by construction.

We say that a route $\mathcal{R} = (\pi, S_{\mathcal{R}})$ crosses the boundary of some cluster C_j^l at level $l > 0$, if π contains an edge $\{u, v\}$, such that $|\{u, v\} \cap C_j^l| = 1$. In the following, we introduce a special type of the net-respecting routes, each of them is restricted to cross the boundary of any cluster not too often and at *portals* exclusively.

Definition 4. Let M be some power of s , for which

$$\frac{M}{s} \leq \frac{dL}{\varepsilon} < M. \quad (4.7)$$

We call a portal an arbitrary point from $C_j^l \cap N(l + \log_s M)$.

Applying Lemma 4 we obtain the following upper bound for the number m of portals of any cluster C_j^l .

$$m \leq \left(2 \frac{4s^{L-l}}{s^{L-l}/M}\right)^d = (8M)^d = O\left(\left(\frac{d \cdot (\log n - \log \varepsilon)}{\varepsilon}\right)^d\right). \quad (4.8)$$

Definition 5. A route \mathcal{R} crossing the boundary of any cluster C_j^l at most r times, is called r -light.

The main result of Subsection 4.3.2 is the following Structure Theorem.

Theorem 5. Let $r = m$ and $d > 1$. For any fixed $\varepsilon \in (0, 1/8)$ and an arbitrary feasible solution \mathfrak{S} of $\text{CVRP}(Z, E, w)$, there exists an appropriate feasible solution $\tilde{\mathfrak{S}}$ consisting of net-respecting and r -light routes, such that

$$\mathbb{E}(w(\tilde{\mathfrak{S}})) = (1 + O(\varepsilon))w(\mathfrak{S}),$$

where the expectation is made over the random hierarchical clustering.

Proof of Theorem 5 is based on consecutive transformation the routes of the given feasible solution \mathfrak{S} using the following technical lemmas, which are the straightforward generalizations of the similar results presented in [Bartal et al., 2016].

Lemma 7. Let $\varepsilon \in (0, 1/8)$. For an arbitrary route $\mathcal{R} = (\pi, S_{\mathcal{R}})$, there exists an appropriate net-respecting route $\tilde{\mathcal{R}} = (\tilde{\pi}, S_{\tilde{\mathcal{R}}})$, $S_{\tilde{\mathcal{R}}} = S_{\mathcal{R}}$, whose cost admits the following upper bound

$$w(\tilde{\mathcal{R}}) \leq (1 + 16\varepsilon)w(\mathcal{R}). \quad (4.9)$$

By Lemma 7, we can assign to an arbitrary feasible solution of the problem in question an appropriate feasible solution consisting of the net-respecting routes exclusively (Fig. 4.2).

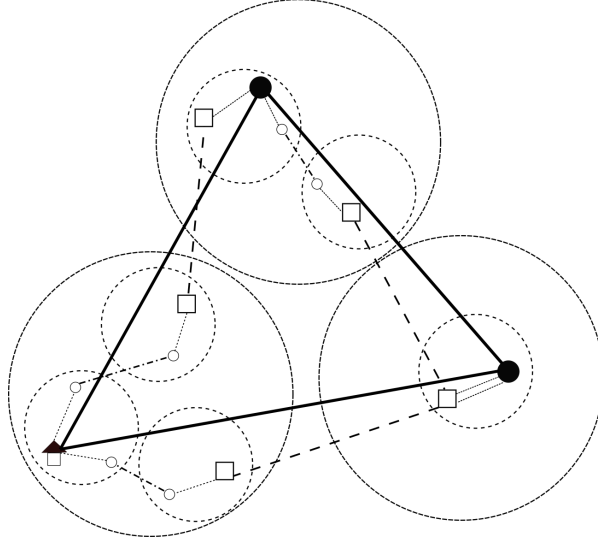


Figure 4.2: An appropriate net-respecting route, Portals of two levels are denoted by \square and \circ

The next lemma states that for an arbitrary $r \geq 2$, without loss of generality, we can assume that all routes of this solution are r -light.

Lemma 8. *For some cluster C_j^l , let $C' \subset C_j^l, |C'| = \tilde{r} > r \geq 2$ be a set of crossing points for the boundary of this cluster by some route $\mathcal{R} = (\pi, S_{\mathcal{R}})$. There exists a route $\tilde{\mathcal{R}} = (\pi, S_{\tilde{\mathcal{R}}}), S_{\mathcal{R}} = S_{\tilde{\mathcal{R}}}$ crossing the boundary of the cluster C_j^l twice, such that its cost can be bounded as follows*

$$w(\tilde{\mathcal{R}}) \leq w(\mathcal{R}) + 4w(\text{MST}(C')). \tag{4.10}$$

Notice, that the claims of Lemma 7 and Lemma 8 remain valid for an arbitrary metric. In turn, for a metric of doubling dimension $d > 1$, the bound (4.10) can be specified in more detail. Indeed, since any cluster C_j^l belongs to a metric ball of radius at most $2s^{L-l}$, for $w(\text{MST}(C'))$,

equation (4.2) is valid, by Lemma 5. Therefore,

$$w(\tilde{\mathcal{R}}) = w(\mathcal{R}) + O\left(s^{L-l} \cdot \tilde{r}^{(1-1/d)}\right). \quad (4.11)$$

Also, we need the following probabilistic result.

Statement 1 (Bartal et al. [2016]). *The probability that two distinct nodes u and v belong to different clusters at level l is at most*

$$c' \cdot w(u, v) \cdot d/s^{L-l}, \quad (4.12)$$

where c' is some absolute constant.

of Theorem 5. Consider an arbitrary route $\mathcal{R} \in \mathfrak{S}$ crossing \tilde{r} times the boundary of some cluster C_j^l for some $\tilde{r} > r$. Thanks to Lemma 8 (and equation (4.11)), we can transform \mathcal{R} to an r -light $\bar{\mathcal{R}}$, such that the surplus cost does not exceed $O\left(s^{L-l} \cdot \tilde{r}^{(1-1/d)}\right)$ per cluster. Thus, the mean surplus cost per each edge of the route \mathcal{R} among $\tilde{r} > r$ edges crossing the boundary of the cluster C_j^l admits the following bound

$$O\left(\frac{s^{L-l} \tilde{r}^{(1-1/d)}}{\tilde{r}}\right) = O\left(\frac{s^{L-l}}{r^{1/d}}\right) = O\left(\frac{s^{L-l}}{M}\right) = O\left(\frac{s^{L-l} \varepsilon}{d \cdot L}\right), \quad (4.13)$$

where the last term follows immediately from inequality (4.7).

Combining equation (4.13) with equation (4.12) proven in Statement 1 and summing on $l = 1, \dots, L + 1$, we ascertain the existence of a feasible solution \mathfrak{S}' consisting of r -light routes, for which

$$\mathbb{E}(w(\mathfrak{S}')) = (1 + O(\varepsilon))w(\mathfrak{S}).$$

Further, applying to each route $\mathcal{R}' \in \mathfrak{S}'$ the claim of Lemma 7, we obtain the desired net-respecting and r -light solution. Theorem 5 is proved. \square

As it follows from Theorem 5, any time, when we need to find an approximate solution of the initial problem, we can restrict ourselves to

the solutions consisting of net-respecting and r -light routes exclusively. In the sequel, we call such solutions net-respecting and r -light as well.

4.3.3 Demand ranking and relaxed solutions

The aforementioned approach relies upon the minimization of total transportation cost in the class of net-respecting and r -light solutions yields a number of seminal approximation results for intractable routing problems, including the well-known Arora’s PTAS for the Euclidean TSP [Arora, 1998] and its extension to metric spaces of a fixed doubling dimension [Bartal et al., 2016]. Unfortunately, it is well-known that, for the CVRP, this approach results in tremendously time expensive algorithms. In this subsection, following to the main idea of the paper [Das and Mathieu, 2015], we outline another approach that leads us to really efficient approximation algorithms based on a concept of *relaxed solutions*.

We start with some necessary definitions and notation. Consider a net-respecting route \mathcal{R} that enters and leaves the cluster C_j^l (located at some level $l > 0$) at portals p^{in} and p^{out} respectively. We call an arbitrary maximal by inclusion fragment

$$\sigma = p^{in}, x_{i_1}, \dots, x_{i_k}, p^{out}, \tag{4.14}$$

which entirely belongs to the cluster C_j^l , a *crossing segment* of the route \mathcal{R} with respect to the cluster C_j^l (or just a *segment*).

Definition 6. Let $\Lambda = \lceil \log_{1+\varepsilon/(L+1)}(q\varepsilon) + 1/\varepsilon \rceil$. Numbers t_i , $i = \overline{1, \Lambda}$ are called *rounding thresholds for covered customer demand*, if

$$t_i = \begin{cases} i & \text{for all } i = 1, \dots, \lfloor 1/\varepsilon \rfloor \\ t_{i-1}(1 + \varepsilon/(L + 1)) & \text{otherwise .} \end{cases} \tag{4.15}$$

Next, we proceed with *ranking* of customer demand. We assume that each unit of the demand has an integer *rank* from the range $0, 1, \dots, L + 1$.

Each customer can have demand units of different ranks. An arbitrary demand unit can be either *active* or *non-active* depending on its rank and level of the considered enclosing cluster. Namely, a demand unit of rank \mathbf{r} is called *active* with respect to any enclosing cluster located at level $l > \mathbf{r}$ (otherwise, this unit is called inactive). By convention, demand units of rank 0 are active at any level.

A segment σ is called *unrounded*, if it covers all the active demand assigned to it. On the other hand, we call a segment σ *rounded*, if it covers exactly t_i units of the assigned active demand, where t_i is the largest rounding threshold (4.15), which does not exceed that demand.

Definition 7. *A set of tours \mathfrak{S} is called a relaxed solution if it covers the total customer demand and there exists an assignment of ranks for all demand units, such that*

- (i) *each route $\mathcal{R} \in \mathfrak{S}$ covers at most q units of the rank 0;*
- (ii) *if a route \mathcal{R} covers exactly t units of active demand at level l , then at level $l + 1$, it covers at most $t(1 + \varepsilon/(L + 1))$ such units;*
- (iii) *for any route $\mathcal{R} \in \mathfrak{S}$, if the number of its segments crossing some cluster C exceeds $\gamma = \left(\frac{L+1}{\varepsilon}\right)^{2d}$, then all these segments are rounded. Otherwise, all of them are unrounded.*

In the following, we call any relaxed solution \mathfrak{S} that is also net-respecting and r -light a *structured solution*. Such solutions are essential point of our approach. Given a random hierarchical clustering, we find a structured solution minimizing the following auxiliary objective function

$$F(\mathfrak{S}) = \sum_{\mathcal{R} \in \mathfrak{S}} w(\mathcal{R}) + \frac{\varepsilon}{L + 1} \sum_{\mathcal{R} \in \mathfrak{S}} \sum_{l=1}^{L+1} c(\mathcal{R}, l) \cdot s^{L-l}, \quad (4.16)$$

where, for any route $\mathcal{R} \in \mathfrak{S}$, $c(\mathcal{R}, l)$ is the number of crossings the boundaries of all clusters at level l .

Notice that with respect to feasible solutions the initial objective function $w(\mathfrak{S})$ and the introduced above function F behave quite similarly.

Theorem 6. *The hypothesis of Theorem 5 implies*

$$\mathbb{E}(F(\tilde{\mathfrak{S}})) = (1 + O(\varepsilon))w(\mathfrak{S}),$$

where the expectation is made over the random hierarchical clustering.

Proof. Indeed, by definition of the auxiliary objective function F , we have

$$F(\tilde{\mathfrak{S}}) = \sum_{\tilde{\mathcal{R}} \in \tilde{\mathfrak{S}}} F(\tilde{\mathcal{R}}) = \sum_{\tilde{\mathcal{R}} \in \tilde{\mathfrak{S}}} \left(w(\tilde{\mathcal{R}}) + \frac{\varepsilon}{L+1} \sum_{l=1}^{L+1} c(\tilde{\mathcal{R}}, l) \cdot s^{L-l} \right).$$

Thanks to Theorem 5, we can estimate

$$\mathbb{E}(w(\tilde{\mathfrak{S}})) \leq (1 + O(\varepsilon)) \cdot w(\mathfrak{S}). \tag{4.17}$$

Further, consider an arbitrary route $\tilde{\mathcal{R}} \in \tilde{\mathfrak{S}}$. By Statement 1 we obtain an upper bound for the expected number of crossings the boundaries of the level l clusters by the route $\tilde{\mathcal{R}}$ as follows:

$$\mathbb{E}(c(\tilde{\mathcal{R}}, l) \cdot s^{L-l}) \leq c' \cdot d \cdot w(\tilde{\mathcal{R}}) = O(1) \cdot w(\tilde{\mathcal{R}}).$$

Summing over all levels $l = \overline{1, L+1}$ and routes $\tilde{\mathcal{R}}$, we get

$$\begin{aligned} \sum_{\tilde{\mathcal{R}} \in \tilde{\mathfrak{S}}} \frac{\varepsilon}{L+1} \sum_{l=1}^{L+1} \mathbb{E}(c(\tilde{\mathcal{R}}, l) \cdot s^{L-l}) &\leq O(\varepsilon) \cdot w(\tilde{\mathfrak{S}}) \leq \\ &O(\varepsilon)(1 + O(\varepsilon)) \cdot w(\mathfrak{S}). \end{aligned} \tag{4.18}$$

Combining bounds (4.17) and (4.18) we obtain the desired bound. Theorem 6 is proved. \square

Let, further, for a given random clustering, \mathfrak{S}_{DP} be a minimizer of the function F in the class of structured solutions². To address the possible infeasibility of \mathfrak{S}_{DP} , we introduce a random ranking of the customer demand by Algorithm 3.

Algorithm 3 Demand Ranking Algorithm

Input: a structured solution \mathfrak{S}_{DP} with respect to some random hierarchical clustering

Output: ranking of all units of demand

- 1: initialize rank of each unit of demand by 0
 - 2: **for each** level l from $L + 1$ to 0 **do**
 - 3: **for each** cluster C_j^l crossed by more than γ segments **do**
 - 4: **for each** segment σ crossing the cluster C_j^l **do**
 - 5: Let \mathbf{a} be the number of active demand units covered by the segment σ and t be the largest threshold, such that $t \leq \mathbf{a}$.
 - 6: Pick an active demand unit at random and $\mathbf{a} - t - 1$ consecutive units (wrapping around to the start of σ if necessary) and assign to them the rank l .
 - 7: **end for**
 - 8: **end for**
 - 9: **end for**
-

Given by a demand ranking, we color each demand unit of the rank 0 in black and all other units in red. After that, we transform the solution \mathfrak{S}_{DP} to the partial solution \mathfrak{S}_{black} by exclusion all the red units. Then, we employ the ITP heuristic to find an approximate CVRP solution \mathfrak{S}_{red} that covers the remaining red demand. Obtain upper bounds for $\mathbb{E}(w(\mathfrak{S}_{black}))$ and $\mathbb{E}(w(\mathfrak{S}_{red}))$ individually. Indeed, by definition of the function F , for any fixed hierarchical clustering,

$$w(\mathfrak{S}_{black}) \leq F(\mathfrak{S}_{black}) \leq F(\mathfrak{S}_{DP}) \leq F(\tilde{\mathfrak{S}}^*),$$

where $\tilde{\mathfrak{S}}^*$ is the net-respecting and r -light feasible solution associated with an arbitrary optimal solution \mathfrak{S}^* of the initial problem, whose existence is guaranteed by Theorem 5. The right-most inequality is valid, since $\tilde{\mathfrak{S}}^*$ is a structured solution, by Lemma 5 from [Das and Mathieu, 2015]. Then,

²In Section 4.3.4, we provide a dynamic programming algorithm, which finds such a solution for any given random clustering

by Theorem 6, we obtain

$$\mathbb{E}(w(\mathfrak{S}_{black})) \leq \mathbb{E}(F(\tilde{\mathfrak{S}}^*)) = (1+O(\varepsilon))w(\mathfrak{S}^*) = (1+O(\varepsilon))\text{CVRP}^*(Z, w, q), \quad (4.19)$$

where the expectation is taken over random clustering. The latter upper bound is given by Lemma 9.

Lemma 9. *For an arbitrary clustering and the expected value of $w(\mathfrak{S}_{red})$ over random ranking of the demand, the following equation*

$$\mathbb{E}(w(\mathfrak{S}_{red})) = O(\varepsilon) \cdot (F(\mathfrak{S}_{DP}) + \text{CVRP}^*(Z, w, q)) \quad (4.20)$$

is valid.

of Lemma 9. As a result of the performed random demand ranking, the initial rounded instance $\text{CVRP-SD}(X \cup \{y\}, D, w, q)$ is decomposed into two separate subinstances

$$\text{CVRP-SD}(X \cup \{y\}, D_{black}, w, q) \quad \text{and} \quad \text{CVRP-SD}(X \cup \{y\}, D_{red}, w, q),$$

where $D_{black}(x) + D_{red}(x) = D(x)$ for any $x \in X$. Then, \mathfrak{S}_{black} is an approximate solution for the former instance, whose expected accuracy bound is given by equation (4.19). To approximate the latter instance, we employ the Iterated Tour Partition (ITP) heuristic [Haimovich and Rinnooy Kan, 1985] providing the solution \mathfrak{S}_{red} .

For the sake of simplicity, we restrict ourselves to the special case, where $D(x) \equiv 1$. This allows us do not distinguish customers and their demand. In the general case, the proof can be obtained in a similar way.

Let, further, $X_{red} = \{x \in X : D_{red}(x) = 1\}$. As it follows from [Haimovich and Rinnooy Kan, 1985], the weight $w(\mathfrak{S}_{red})$ admits the following upper bound

$$w(\mathfrak{S}_{red}) \leq \text{Rad}(X_{red}) + 2\beta \cdot \text{TSP}^*(X_{red} \cup \{y\}), \quad (4.21)$$

where, for any $X' \subset X$, $Rad(X') = \frac{2}{q} \sum_{x \in X'} w(x, y)$, $TSP^*(X_{red} \cup \{y\})$ is an optimum of the internal instance of the metric TSP, and β is an accuracy bound³ of a polynomial time algorithm used to find an approximate solution of this instance. The remainder of the proof follows straightforwardly from Lemma 10 and Lemma 11. Lemma 9 is proved. \square

Hereinafter in this section, unless explicitly stated otherwise, we assume that expectations are taken over random ranking of the demand units.

Lemma 10.

$$\mathbb{E}(Rad(X_{red})) = \varepsilon \cdot CVRP^*(Z, w, q).$$

Proof. By construction, $X_{red} = X_{red}^1 \cup \dots \cup X_{red}^{L+1}$, where X_{red}^l is the subset consisting from the customers with demand of rank l . Then, any customer x belongs to the subset X_{red} with probability

$$\mathbb{P}[x \in X_{red}] = \sum_{l>0} \mathbb{P}[x \in X_{red}^l]$$

with respect to random ranking of the demand.

In turn, the customer x belongs to X_{red}^l , if and only if, for some cluster C_j^l , there exists a crossing segment σ containing x , whose demand unit was picked by Algorithm 3 at Step 6. Let \mathbf{a} be an amount of active demand units covered by the segment σ and t be the appropriate threshold, for which $t \leq \mathbf{a} < t(1 + \varepsilon/(L + 1))$. Then, by construction,

$$\mathbb{P}[x \in X_{red}^l] = \frac{\mathbf{a} - t}{\mathbf{a}} < \frac{t(1 + \varepsilon/(L + 1)) - t}{t} = \frac{\varepsilon}{L + 1}.$$

Thus, $\mathbb{P}[x \in X_{red}] < \varepsilon$. Finally, we have

$$\mathbb{E}(Rad(X_{red})) = \frac{2}{q} \sum_{x \in X} w(x, y) \mathbb{P}[x \in X_{red}] < Rad(X) \cdot \varepsilon \leq \varepsilon \cdot CVRP^*(Z, w, q),$$

³e.g., $\beta = 3/2$ for the well-known Christofides-Serdyukov algorithm

where the last inequality follows from Theorem 1 of [Haimovich and Rinnooy Kan, 1985]. Lemma 10 is proved. \square

Lemma 11.

$$\mathbb{E}(\text{TSP}^*(X_{red} \cup \{y\})) = O(\varepsilon) \cdot F(\mathfrak{S}_{DP}).$$

Proof. To obtain the desired upper bound, due to the obvious inequalities

$$w(\text{MST}(X_{red} \cup \{y\})) \leq \text{TSP}^*(X_{red} \cup \{y\}) \leq 2 \cdot w(\text{MST}(X_{red} \cup \{y\})),$$

which are valid for any metric TSP instance, and

$$w(\text{MST}(X_{red} \cup \{y\})) \leq \sum_{l=1}^{L+1} w(\text{MST}(X_{red}^l \cup \{y\})),$$

it is sufficient to prove that

$$\mathbb{E}(\text{MST}(X_{red}^l \cup \{y\})) \leq O(\varepsilon/(L+1)) \cdot F(\mathfrak{S}_{DP})$$

To proceed, we introduce the following additional notation:

- (i) let \mathcal{C}^l be the set of clusters C of level l , for which $C_{red} = (X_{red}^l \cap C) \neq \emptyset$; without loss of generality, we assume that the depot y belongs to some cluster $C \in \mathcal{C}^l$;
- (ii) for any $C \in \mathcal{C}^l$, let $P_C \subset C$ be a subset of portals of the cluster C (augmented by the depot y , if $y \in C$).

In addition, we introduce the complete edge-weighted graph H with node set \mathcal{C}^l , where the cost of any edge $\{C_1, C_2\}$ is specified by the shortest distance $w(p, q)$, where p and q are portals in clusters C_1 and C_2 ,

respectively. Evidently,

$$w(\text{MST}(X_{red}^l \cup \{y\})) \leq \sum_{C \in \mathcal{C}^l} w(\text{MST}(C_{red} \cup \{p\})) + \sum_{C \in \mathcal{C}^l} w(\text{MST}(P_C)) + w(\text{MST}(H)), \quad (4.22)$$

where p be an arbitrary portal of the cluster C (Fig. 4.3).

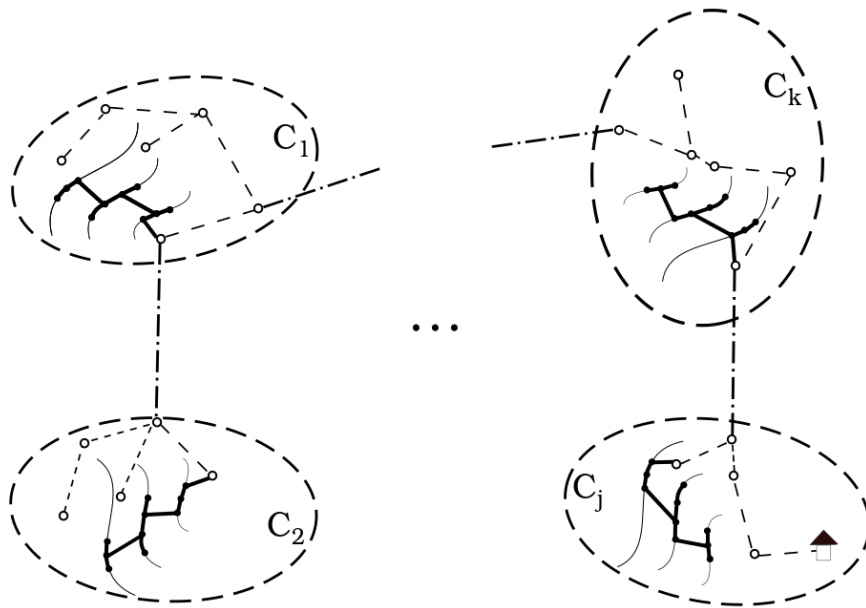


Figure 4.3: To obtain an upper bound for the weight of $\text{MST}(X_{red}^l \cup \{y\})$, we combine the spanning trees for the red points C_{red} (bold solid lines) and the portals (dashed) in any cluster at level l with the minimum spanning tree in the auxiliary graph of clusters H

Remainder of the proof is based on Statement 2 and the claims of Lemma 12, Lemma 13, and Lemma 14, respectively. Lemma 11 is proved. \square

For any cluster $C \in \mathcal{C}^l$, by $F|_C$ we denote a restriction of the objective function F onto the cluster C . For an arbitrary structured solution \mathfrak{S} ,

$F(\mathfrak{G})|C$ is defined by the formula

$$F(\mathfrak{G})|C = \sum_{\mathcal{R} \in \mathfrak{G}} w(\mathcal{R} \cap C) + \frac{\varepsilon}{L+1} \cdot s^{L-l} \cdot \sum_{\mathcal{R} \in \mathfrak{G}} c(\mathcal{R}, C),$$

where

- (i) $w(\mathcal{R} \cap C)$ is equal to $w(\mathcal{R})$, if the route \mathcal{R} is enclosed by the cluster C , and the total cost of all segments of the route \mathcal{R} crossing this cluster, otherwise; (ii) $c(\mathcal{R}, C)$ is the number of crossings the boundary of the cluster C by the route \mathcal{R} .

Statement 2. For any $l > 0$,

$$\sum_{C \in \mathcal{C}^l} F(\mathfrak{G}_{DP})|C \leq F(\mathfrak{G}_{DP}).$$

The proof of Statement 2 can be obtained immediately from the additivity of the objective function F .

Lemma 12. For any cluster $C \in \mathcal{C}^l$,

$$\mathbb{E}(w(\text{MST}(C_{red} \cup \{p\}))) = O\left(\frac{\varepsilon}{L+1}\right) \cdot F(\mathfrak{G}_{DP})|C.$$

Proof. To obtain the desired bound, consider the tree T_C constructed as follows:

- (i) for an arbitrary crossing segment σ of the cluster C , we take an interval (maybe wrapped) σ_{red} induced by the red customers picked at Step 6 by Algorithm 3;
- (ii) let H_C be any minimum hitting set for all connected fragments of such intervals and p be an arbitrary portal located in the cluster C ;
- (iii) the tree T_C is made up by augmentation of $\text{MST}(H_C \cup \{p\})$ by all intervals σ_{red} .

Establish an upper bound for

$$w(T_C) = \sum_{\sigma} w(\sigma_{red}) + w(\text{MST}(H_C \cup \{p\})).$$

According to Algorithm 3, $\mathbb{E}(w(\sigma_{red})) < \varepsilon/(L+1) \cdot w(\sigma)$ for any segment σ . Therefore,

$$\mathbb{E} \left(\sum_{\sigma} w(\sigma_{red}) \right) < \frac{\varepsilon}{(L+1)} \cdot F(\mathfrak{S}_{DP})|_C.$$

Further, the number N_C of segments σ crossing the cluster C exceeds γ , since \mathfrak{S}_{DP} is a structured solution. i.e. $N_C = \gamma \cdot g_C + r_C$ for some $g_C \geq 1$ and $0 \leq r_C < \gamma$. Then, $|H_C| \leq 2N_C \leq 4\gamma \cdot g_C$. Therefore, by Lemma 5,

$$w(\text{MST}(H_C \cup \{p\})) = O(s^{L-l} \cdot |H_C|^{1-1/d}) = O(s^{L-l} \cdot (\gamma \cdot g_C)^{1-1/d})$$

for any fixed $d > 1$. On the other hand,

$$\begin{aligned} F(\mathfrak{S}_{DP})|_C &\geq \frac{\varepsilon}{L+1} \cdot s^{L-l} \cdot 2N_C \geq \frac{\varepsilon}{L+1} \cdot s^{L-l} \cdot \gamma \cdot g_C \\ &= \Omega \left(\frac{\varepsilon}{L+1} \right) \cdot \gamma^{1/d} \cdot w(\text{MST}(H_C \cup \{p\})). \end{aligned}$$

Hence,

$$w(\text{MST}(H_C \cup \{p\})) = O \left(\frac{L+1}{\varepsilon} \right) \cdot \gamma^{-1/d} \cdot F(\mathfrak{S}_{DP})|_C = O \left(\frac{\varepsilon}{L+1} \right) \cdot F(\mathfrak{S}_{DP})|_C,$$

since $\gamma = ((L+1)/\varepsilon)^{2d}$, by Definition 7. Lemma 12 is proved. □

Lemma 13. *For any cluster $C \in \mathcal{C}^l$,*

$$w(\text{MST}(P_C)) = O \left(\left(\frac{\varepsilon}{L+1} \right)^d \right) F(\mathfrak{S}_{DP})|_C.$$

Proof. By construction (see, also equations (4.7) and (4.8),

$$|P_C| \leq m + 1 = O\left(\left(\frac{dLs}{\varepsilon}\right)^d\right) = O\left(\left(\frac{L+1}{\varepsilon}\right)^d\right).$$

By Lemma 5, we obtain

$$w(\text{MST}(P_C)) = O\left(s^{L-l} \left(\frac{L+1}{\varepsilon}\right)^{d-1}\right).$$

Meanwhile, the cluster C is crossed by at least γ segments. Therefore,

$$\begin{aligned} F(\mathfrak{S}_{DP})|_C &\geq 2\gamma \frac{\varepsilon}{L+1} \cdot s^{L-l} \geq 2 \left(\frac{L+1}{\varepsilon}\right)^{2d-1} \cdot s^{L-l} \\ &= \Omega\left(\left(\frac{L+1}{\varepsilon}\right)^d\right) \cdot w(\text{MST}(P_C)). \end{aligned}$$

Thus,

$$w(\text{MST}(P_C)) = O\left(\left(\frac{\varepsilon}{L+1}\right)^d\right) F(\mathfrak{S}_{DP})|_C.$$

Lemma 13 is proved. □

Lemma 14.

$$\mathbb{E}(w(\text{MST}(H))) = O\left(\frac{\varepsilon}{L+1}\right) \cdot F(\mathfrak{S}_{DP})$$

Proof. Lemma 14 follows immediately from Lemma 10 proved in [Das and Mathieu, 2015]. □

Finally, relying on equation (4.19), Lemma 9, and Theorem 6, we easily obtain the main result of this subsection.

Theorem 7. *Let an instance of the CVRP be given in a metric space of a fixed dimension $d > 1$ and $r = m$. Then, for any $\varepsilon \in (0, 1/8)$, Das and Mathieu randomized scheme provides an approximate solution $\mathfrak{S}_{black} \cup \mathfrak{S}_{red}$,*

such that

$$\mathbb{E}(w(\mathfrak{S}_{black}) + w(\mathfrak{S}_{red})) = (1 + O(\varepsilon))\text{CVRP}^*(Z, w, q),$$

where the expectation is taken over random clustering and ranking of the demand.

Proof. Indeed, equation (4.19) gives us an upper bound for the black routes

$$\mathbb{E}(w(\mathfrak{S}_{black})) \leq (1 + O(\varepsilon))\text{CVRP}^*(Z, w, q). \quad (4.23)$$

In turn, an expected cost of the red routes provided by the statement of lemma (9) is as follows

$$\mathbb{E}(w(\mathfrak{S}_{red})) = O(\varepsilon) \cdot (F(\mathfrak{S}_{DP}) + \text{CVRP}^*(Z, w, q)). \quad (4.24)$$

As we showed above, $F(\mathfrak{S}_{DP}) \leq F(\tilde{\mathfrak{S}}^*)$, where $\tilde{\mathfrak{S}}^*$ is the net-respecting and r -light solution associated with an arbitrary optimal solution \mathfrak{S}^* of the initial problem. Due to this fact and statement of Theorem 6,

$$\mathbb{E}(F(\mathfrak{S}_{DP})) \leq (1 + O(\varepsilon)) \cdot \text{CVRP}^*(Z, w, q). \quad (4.25)$$

Combining bounds (4.23)-(4.25), we obtain the theorem statement. Theorem 7 is proved. \square

The obtained results shed new light on the approximation of the Capacitated Vehicle Routing Problem formulated in metric spaces of a fixed dimension. Actually, Theorem 7 implies that any structured solution \mathfrak{S}_{DP} minimizing the auxiliary objective function F can be transformed into a required approximate solution if the given problem. Furthermore, as it follows from the seminal paper [Haimovich and Rinnooy Kan, 1985], such post-processing can be carried out in polynomial time. In the sequel, we develop an efficient procedure for finding such structured solutions.

4.3.4 Baseline Dynamic Programming

In this section, we present a short overview of our adaptation of the initial Das and Mathieu dynamic program to the case of metric spaces of a fixed doubling dimension.

We start with some necessary notation. We encode an arbitrary crossing segment (4.14) by a tuple $(p^{in}, p^{out}, \mathbf{s}, \mathbf{d})$, where \mathbf{s} is the amount of covered active demand units and \mathbf{d} indicates whether this segment should visit the depot y .

Given by a fixed hierarchical clustering, we index entries of the lookup table of our dynamic program by couples (C, \mathfrak{C}) , where C is a cluster and \mathfrak{C} is a *configuration* defining behavioral patterns for all segments crossing the boundary of the cluster C . Depending on the number of segments described, we distinguish two kinds of configurations, *unrounded* and *rounded*.

An *unrounded configuration* is just a finite sequence of at most γ tuples

$$((p_i^{in}, p_i^{out}, \mathbf{s}_i, \mathbf{d}_i) : i = \overline{1, k_u}),$$

each of them represents a single unrounded crossing segment. On the other hand, a *rounded configuration* is set of ordered pairs

$$\{(s_i, m_i) : i = \overline{1, k_r}\}, \quad s_{i_1} \neq s_{i_2},$$

each of them defines a common behavior pattern $s_i = (p_i^{in}, p_i^{out}, t_i, \mathbf{d}_i)$ for exactly m_i rounded segments. Namely, all such segments should enter and leave the cluster in portals p_i^{in} and p_i^{out} respectively, cover t_i units of active demand exactly (for some threshold t_i), and visit the depot according to the value of \mathbf{d}_i .

To define the concept of a *feasible* lookup table entry, we need some technical notation. A family Σ of segments crossing the boundary of some

cluster C augmented by a number of routes enclosed within this cluster is called a *partial relaxed solution* for the cluster C , if this family covers all the active customer demand in this cluster and fulfills conditions (i), (ii), and (iii) enlisted in Definition 7 (with respect to this cluster).

Definition 8. *An entry (C, \mathfrak{C}) is called feasible, if there exists a partial relaxed solution $\Sigma = \Sigma(C)$, such that*

- (i) *if \mathfrak{C} is unrounded, then all the segments of Σ are unrounded and are too, s.t. there exists a one-to-one correspondence between them and the entries of the configuration \mathfrak{C} ;*
- (ii) *otherwise, if \mathfrak{C} is rounded, then the family Σ is partitioned into k_r subfamilies, such that the i -th subfamily consists of m_i rounded crossing segments sharing the same behavior pattern s_i .*

As usual, the lookup table entries are computed bottom-up. The base case corresponds to the level $L + 1$, where all the clusters are singletons. Thus, all the entries can be computed trivially.

To proceed with the recurrence, assume that all the entries for the levels $l + 1, \dots, L + 1$ are already calculated. Fix an arbitrary cluster C_j^l and try to compute the entry (C_j^l, \mathfrak{C}) for some configuration \mathfrak{C} . By the given clustering, we have a partition $C_j^l = C_{j1}^{l+1} \cup \dots \cup C_{jK}^{l+1}$ for some $K = 2^{O(d)}$. Guided by the approach proposed in [Das and Mathieu, 2015], to compute the entry (C_j^l, \mathfrak{C}) , it is necessary to employ the two-stage exhaustive search as follows:

Stage (i) to enumerate all the combinations

$$((C_{j1}^{l+1}, \mathfrak{C}_1), \dots, (C_{jK}^{l+1}, \mathfrak{C}_K)) \quad (4.26)$$

of the computed already entries induced by the child subclusters;

Stage (ii) for any given combination (4.26), enumerate all the ways to stitch child configurations to fulfill the initial configuration \mathfrak{C} . Any time when this stitching is possible, the record value of the function F is updated.

Thus, the entry (C_j^l, \mathfrak{C}) is filled by the resulting record value, if this value was updated at least once. Otherwise, the entry is set to be infeasible and excluded from the consideration. To obtain the desired structured solution \mathfrak{S}_{DP} minimizing the objective function F , it is sufficient to compute the only entry (C_1^0, \mathfrak{C}) at level 0 for the empty configuration \mathfrak{C} .

The point is that although for the finite-dimensional Euclidean spaces considered by Das and Mathieu, Stage (ii) can be calculated efficiently, in metric spaces even of a fixed doubling dimension, its running time is no longer quasi-polynomial.

Indeed, at Stage (ii), the calculations are specified in terms of *concatenation profiles* and *interface vectors*. A concatenation profile defines the stitching order for any single segment crossing the boundary of the cluster C_j^l (or a route contained in it).

Namely, a finite sequence of tuples $\varphi = ((p_k^{in}, p_k^{out}, t_k, \mathbf{d}_k) : k = \overline{1, \theta_\varphi})$ is called a concatenation profile, if, for each tuple,

- (i) p_k^{in} and p_k^{out} are some child portals
- (ii) t_k is either a threshold or a natural number from $[1, \gamma]$
- (iii) \mathbf{d}_k indicates whether depot should be visited.

In turn, each entry of an interface vector specifies the number of times when some concatenation profile is used during the stitching procedure. By definition, an interface vector has the form $\mathfrak{I} = (n_1, \dots, n_{|\Phi|})$, where $n_i \in [0, n \cdot r]$ and Φ is the number of all possible concatenation profiles. Since, by construction, $|\Phi| = (\log n)^{\Omega(r)}$, the number of distinct interface

vectors enumerated at Stage (ii) is at least

$$(nr)^{|\Phi|} = (nr)^{(\log n)^{\Omega(r)}}. \quad (4.27)$$

Evidently, the lower bound (4.27) is not quasi-polynomial for an arbitrarily slow increasing function $r = r(n)$. Therefore, we cannot claim that the aforementioned algorithm retains quasi-polynomial running time-bound in metric spaces of a fixed doubling dimension, even for any fixed $q > 2$, since at the moment no structure theorems are known for such spaces, proved for a constant r [see, e.g. [Bartal et al., 2016](#)]. This shortcoming of the straightforward use of the Das and Mathieu scheme becomes even more obvious in the context of recent results [[Khachai and Ogorodnikov, 2019a](#)] showing that the CVRP in a metric space of any fixed doubling dimension admits a PTAS for any constant q .

In the following subsection, we propose our modification of this scheme, where, at Stage (ii) of the recursive step, the exhaustive search for the optimal interface vector is replaced with an internal dynamic program, such that the resulting scheme becomes QPTAS again, at least for $q = \text{polylog}(n)$.

4.3.5 Modification based on internal DP

To shorten the redundancy, in this subsection we use the simplified notation (C, \mathfrak{C}) and $(C_1, \mathfrak{C}_1), \dots, (C_K, \mathfrak{C}_K)$ for the entry to be computed and its child entries, respectively.

We start with the following simple remark

Remark 2. *For any entry (C, \mathfrak{C}) , there exist four possible options:*

1. *the configuration \mathfrak{C} is rounded and the cluster C encloses the depot y*
2. *the configuration \mathfrak{C} is rounded and the cluster C contains no depots*
3. *the configuration \mathfrak{C} is unrounded and the cluster C encloses the depot*

4. the configuration \mathfrak{C} is unrounded and the cluster C contains no depots.

In the sequel, we consider option (i) in detail, since it appears to be the most general and involves the greatest time consumption.

Indeed, let the depot y be enclosed in the cluster C and the configuration $\mathfrak{C} = \{(s_i, m_i) : i = \overline{1, k_{\mathfrak{C}}}\}$ be rounded. Without loss of generality, we assume that all the given child configurations

$$\mathfrak{C}_u = \{(s_v^u, m_v^u) : v = \overline{1, k_u}\}, \quad (u \in \{1, \dots, K\})$$

are rounded as well.

To match the configuration \mathfrak{C} with child configurations $\mathfrak{C}_1, \dots, \mathfrak{C}_K$, we need to assign to each s_i a sequence $\Phi_i = (\varphi_{i,1}, \dots, \varphi_{i,m_i})$ of not necessarily distinct concatenation profiles, such that

1. each profile $\varphi_{i,j}$ consists of tuples s_v^u only;
2. for any tuple $s_i = (p_i^{in}, p_i^{out}, t_i, \mathbf{d}_i)$, the following inequality

$$t_i \leq D_{\varphi_{i,j}} < t_i \left(1 + \frac{\varepsilon}{L+1}\right)$$

holds, where $D_{\varphi_{i,j}}$ is total active demand covered by the profile $\varphi_{i,j}$;

3. the surplus active demand in the cluster C (not covered by the segments specified by the configuration \mathfrak{C}) is serviced by some sequence

$$\Phi_{k_{\mathfrak{C}}+1} = (\varphi_{(k_{\mathfrak{C}}+1),1}, \dots, \varphi_{(k_{\mathfrak{C}}+1),T_{\mathfrak{C}}})$$

of additional closed profiles, each whose profile φ visits the depot y and fulfills the capacity constraint $D_{\varphi} \leq q$;

4. each tuple s_v^u is contained in the profiles with total multiplicity m_v^u ;
5. total length of the constructed concatenation profiles

$$\sum_{i=1}^{k_{\mathfrak{C}}} \sum_{j=1}^{m_i} \text{cost}(\varphi_{i,j}) + \sum_{j=1}^{T_{\mathfrak{C}}} \text{cost}(\varphi_{(k_{\mathfrak{C}}+1),j}) \quad (4.28)$$

takes the smallest value.

Here $\text{cost}(\varphi)$ is defined as

$$\text{cost}(\varphi) = \sum_{k=1}^{\theta-1} \tilde{w}(p_k^{\text{out}}, p_{k+1}^{\text{in}}) + \frac{2\theta \cdot \varepsilon}{L+1} \cdot s^{L-l-1} \quad (4.29)$$

for any non-closed profile $\varphi = ((p_k^{\text{in}}, p_k^{\text{out}}, x_k, \mathbf{d}_k) : k = \overline{1, \theta})$, whilst

$$\text{cost}(\varphi) = \sum_{k=1}^{\theta-1} \tilde{w}(p_k^{\text{out}}, p_{k+1}^{\text{in}}) + \frac{2\theta \cdot \varepsilon}{L+1} \cdot s^{L-l-1} + \tilde{w}(p_{\theta}^{\text{out}}, p_1^{\text{in}}) \quad (4.30)$$

for any closed one and

$$\tilde{w}(p_k^{\text{out}}, p_{k+1}^{\text{in}}) = \begin{cases} w(p_k^{\text{out}}, p_{k+1}^{\text{in}}), & \text{if } p_k^{\text{out}} \text{ and } p_{k+1}^{\text{in}} \text{ satisfy Definition 3,} \\ +\infty, & \text{otherwise.} \end{cases}$$

Notice, that criterion (4.28) and the reduced costs of concatenation profiles (4.29)-(4.30) can be obtained straightforwardly from the auxiliary objective function (4.16). Indeed, for any given configuration \mathfrak{C} and child ones $\mathfrak{C}_1 \dots, \mathfrak{C}_K$, thanks to condition (iv), the total cost of all child subsegments does not depend on the choice of profiles $\varphi_{i,j}$. Therefore, we exclude this cost from (4.29) and (4.30).

Further, notice that each concatenation profile $\varphi_{i,j}$ to be constructed can have its own size $\theta_{\varphi_{i,j}}$ fulfilling the condition $\theta_{\varphi_{i,j}} \leq K \cdot r$, due to the r -lightness of the resulting solution. To ensure that all profiles have the same size $\bar{r} = K \cdot r$, we *pad* each of them by enough copies of the dummy tuple σ_0 .

For the sake of convenience, we introduce the set

$$\bar{S} = \{\sigma_0\} \cup \bigcup_{u=1}^K \{s_1^u, \dots, s_{k_u}^u\} = \{\sigma_0, \sigma_1, \dots, \sigma_{\mathcal{K}}\}, \quad \mathcal{K} = \sum_{u=1}^K k_u$$

containing all the tuples s_v^u from the child configurations and augmented by the dummy tuple σ_0 .

To proceed, we propose a three-dimensional nonnegative integer-valued *resource matrix* A of size $[(k_{\mathfrak{C}} + 1) \times (\mathcal{K} + 1) \times \bar{r}]$, whose entry $a_{i,\nu}^p$ specifies how many times the tuple σ_ν is used in the sequence of concatenation profiles Φ_i at position p . For any fixed i , we call the submatrix $A_i = \|a_{i,\nu}^p\|$, where $p = \overline{1, \bar{r}}$ and $\nu = \overline{0, \mathcal{K}}$, an i -th *resource row*.

By construction, for any $i = \overline{1, K_{\mathfrak{C}}}$, the row A_i corresponds to the appropriate tuple (s_i, m_i) of the configuration \mathfrak{C} , whilst the last row $A_{k_{\mathfrak{C}}+1}$ is related to the collection of additional routes enclosed in the cluster C .

The resource matrix A is called *feasible* with respect to the configurations \mathfrak{C} and $\mathfrak{C}_1, \dots, \mathfrak{C}_K$, if the following equations

$$\sum_{\nu=0}^{\mathcal{K}} a_{i,\nu}^p = m_i, \quad (p = \overline{1, \bar{r}}, i = \overline{1, k_{\mathfrak{C}}}) \quad (4.31)$$

$$\sum_{\nu=0}^{\mathcal{K}} a_{k_{\mathfrak{C}}+1,\nu}^p = T_{\mathfrak{C}}, \quad (p = \overline{1, \bar{r}}) \quad (4.32)$$

$$m_i t_i \leq \sum_{p=1}^{\bar{r}} \sum_{\nu=1}^{\mathcal{K}} a_{i,\nu}^p \cdot t_\nu < m_i t_i \left(1 + \frac{\varepsilon}{L+1}\right), \quad (i = \overline{1, k_{\mathfrak{C}}}), \quad (4.33)$$

$$\sum_{p=1}^{\bar{r}} \sum_{\nu=1}^{\mathcal{K}} a_{k_{\mathfrak{C}}+1,\nu}^p \cdot t_\nu \leq T_{\mathfrak{C}} \cdot q \quad (4.34)$$

are valid for some non-negative integer constant $T_{\mathfrak{C}}$.

Our Algorithm 4 comes as a replacement of Stage (ii) of the initial Das and Mathieu's scheme. Its main idea is based on the construction of

Algorithm 4 Our ‘Stage (ii)’

Input: a parent cluster C with associated configuration \mathfrak{C} and the child DP table entries $(C_1, \mathfrak{C}_1), \dots, (C_K, \mathfrak{C}_K)$

Output: the minimum value of the objective function (4.28) for the given configurations $\mathfrak{C}, \mathfrak{C}_1, \dots, \mathfrak{C}_K$

- 1: **for each** resource matrix A **do**
- 2: check the validity of the feasibility constraints (4.31)-(4.34)
- 3: **if** the matrix A is feasible **then**
- 4: **for each** $i \in \{1, \dots, k_{\mathfrak{C}}\}$ **do**
- 5: employ the Internal Dynamic Program (Algorithm 5) to obtain the minimum cost family Φ_i of m_i concatenation profiles compatible with A_i (or show that it is impossible)
- 6: **end for**
- 7: employ the similar version of the internal DP (Algorithm 6) adapted for construction of the minimum cost routes $\Phi_{k_{\mathfrak{C}}+1}$ enclosed in the cluster C
- 8: **if** all Φ_i are constructed **then**
- 9: sum up their costs and update the record
- 10: **end if**
- 11: **end if**
- 12: **end for**

a minimum cost family of concatenation profiles Φ_i *compatible* with any given resource row A_i .

Table 4.1: An example of a resource row A_i and related sequence of the concatenation profiles for $\mathcal{K} = 3$, $m_i = 5$, and $\bar{S} = \{\sigma_0, \dots, \sigma_3\}$

A_i	\backslash	p		1		2		\dots
	σ							
				0		1		
				1		2		
				2		3		
				3		0		2

$$\Phi_i = \begin{bmatrix} (\sigma_1, \sigma_2, \dots) \\ (\sigma_1, \sigma_3, \dots) \\ (\sigma_2, \sigma_1, \dots) \\ (\sigma_2, \sigma_3, \dots) \\ (\sigma_2, \sigma_0) \end{bmatrix}$$

Skipping the rigorous definition of the aforementioned compatibility, we illustrate it by the simple example (see Table 4.1).

Internal dynamic programming algorithm

For the sake of convenience, we present two versions of the internal dynamic program. The former one (Algorithm 5) is intended to construct the route segments related to the parent configuration \mathfrak{C} , whilst the latter (Algorithm 6) is adapted to stitch additional routes enclosed in the cluster C . Although these algorithms are fairly close in general, there are a few substantial nuances in their implementation. Therefore, in the sequel, we discuss each of the algorithms separately.

Case 1: stitching the route segments related to the configuration \mathfrak{C} . The goal of Algorithm 5 is to construct a family of the minimum total cost (induced by the objective function F), which consists of m_i segments crossing the boundary of the cluster C , each of them corresponds to the behavior pattern $s_i = (p_i^{in}, p_i^{out}, t_i, \mathbf{d}_i)$. Every such a segment is stitched from the child subsegments (defined by the patterns $\sigma_\nu \in \bar{S}$) in accordance to some concatenation profile $\varphi_{i,j} \in \Phi_i$. For the sake of simplicity, in the sequel, we do not distinguish those segments and the concatenation profiles that specify them and call the desired family Φ_i as well.

We construct the desired family Φ_i by recursion on the position p in concatenation profiles.

Each entry of the internal dynamic programming lookup table is indexed by a couple (p, H_p) , where $p = 1, \dots, \bar{r}$ indicates the current position, and the matrix $H_p = \|h_{\nu,c,\mathbf{d}}^p\|$, $\nu = \overline{0, \mathcal{K}}$, $c = \overline{0, q}$, $\mathbf{d} = \overline{0, \bar{r}}$ specifies terminal constraints on a family $\Phi_i^{(p)}$ of m_i partial concatenation profiles of length p .

Actually, each entry $h_{\nu,c,\mathbf{d}}^p$ of the matrix H_p denotes the number of such profiles (in this family), that cover exactly c units of active demand in total, visit the depot \mathbf{d} times and have the same tuple σ_ν at position p .

Algorithm 5 Internal Dynamic Program for constructing segments associated with a tuple s_i of a given configuration \mathfrak{C}

Input: a resource row A_i .

Output: a family Φ_i of concatenation profiles compatible with A_i and minimizing the function \bar{F} .

1: base case: the only feasible entry $(1, H_1)$, where

$$h_{\nu,c,\mathbf{d}}^1 = \begin{cases} a_{i,\nu}^1, & \text{if } \nu > 0, c = t_\nu, \text{ and } \mathbf{d} = \mathbf{d}_\nu \\ 0, & \text{otherwise} \end{cases}$$

and $\bar{D}(1, H_1) = 0$

2: recursive step: assume that all feasible entries are computed for any $p' < p$

3: **for each** H_p compatible with the p -th column of the resource row A_i **do**

4: apply the Bellman equation to compute an entry (p, H_p)

$$\begin{aligned} \bar{D}(p, H_p) = & \min_{X=\|x_{\nu_1,\nu_2}^{c,\mathbf{d}}\|} \{ \bar{D}(p-1, H_{p-1}(X)) + \\ & \sum_{\nu_1=1}^{\mathcal{K}} \sum_{\nu_2=0}^{\mathcal{K}} \sum_{c=0}^q \sum_{\mathbf{d}=0}^{\bar{r}} x_{\nu_1,\nu_2}^{c,\mathbf{d}} \text{conn}(\sigma_{\nu_1}, \sigma_{\nu_2}) \}, \end{aligned} \quad (4.35)$$

where the minimization is carried out over the feasible entries $(p-1, H_{p-1}(X))$ only. In turn, $H_{p-1}(X)$ is obtained by the following formulas

$$\begin{aligned} h_{0,c,\mathbf{d}}^{p-1} &= x_{0,0}^{c,\mathbf{d}}, \quad (c = \bar{0}, q, \mathbf{d} = \bar{0}, \bar{r}) \\ h_{\nu_1,c,\mathbf{d}}^{p-1} &= \sum_{\nu_2=0}^{\mathcal{K}} \{ x_{\nu_1,\nu_2}^{c+t_{\nu_2}, \mathbf{d}+\mathbf{d}_{\nu_2}} : c+t_{\nu_2} \leq q \}, \quad (\nu_1 = \bar{1}, \mathcal{K}, c = \bar{0}, q, \mathbf{d} = \bar{0}, \bar{r}). \end{aligned}$$

If at least one such an entry is found, then the result is stored in (p, H_p)

5: **end for**

6: **if** there are no feasible entries $(\bar{r}, H_{\bar{r}})$ or $\inf\{\bar{D}(\bar{r}, H_{\bar{r}})\} = \infty$ **then**

7: output ‘no profile families compatible with A_i .’

8: **else**

9: the cost of the desired family Φ_i is contained within the entry

$$(\bar{r}, H_{\bar{r}}^*) = \arg \min\{\bar{D}(\bar{r}, H_{\bar{r}})\}. \quad (4.36)$$

10: output the optimal solution Φ_i , which can be obtained from (4.36) by backtracking.

11: **end if**

A matrix H_p is called *compatible* with the p -th column of a resource A_i , if $\sum_{c=0}^q \sum_{\mathbf{d}=0}^{\bar{r}} h_{\nu,c,\mathbf{d}}^p = a_{i,\nu}^p$ is valid for any $\nu = \bar{0}, \mathcal{K}$. In addition, $H_{\bar{r}}$ is

compatible if and only if, for any ν , $h_{\nu, c, \mathbf{d}}^{\bar{r}} > 0$ implies

$$t_i \leq c < t_i \left(1 + \frac{\varepsilon}{L + 1} \right), \text{ and } \mathbf{d}_i \leq \mathbf{d} \leq \mathbf{d}_i \cdot \bar{r}.$$

Notice, that, if the cluster C is located at level l , then, for any given resource row A_i , the sum of terms penalizing for crossings all the boundaries of the child subclusters (at level $l + 1$) is fixed and does not depend on Φ_i . Therefore, we can restrict ourselves to the minimization of the stitching costs for child subsegments only.

Thus, we define our reduced internal objective function \tilde{F} as follows. Let $\Phi_i^{(p)}$ be a family of partial concatenation profiles $\varphi_{i,1}^{(p)}, \dots, \varphi_{i,m_i}^{(p)}$, each of them consists of p tuples. Then, $\bar{F}(\Phi_i^{(p)}) = \sum_{j=1}^{m_i} \overline{\text{cost}}(\varphi_{i,j}^{(p)})$, where, for any partial profile $\varphi^{(p)} = (\sigma_{i_1}, \dots, \sigma_{i_p})$, its reduced cost is defined by

$$\overline{\text{cost}}(\varphi^{(p)}) = \sum_{k=1}^{p-1} \text{conn}(\sigma_{i_k}, \sigma_{i_{k+1}}) = \sum_{k=1}^{p-1} \tilde{w} (p^{out}(\sigma_{i_k}), p^{in}(\sigma_{i_{k+1}})).$$

Further, the Bellman function \bar{D} takes the form

$$\bar{D}(p, H_p) = \min \{ \bar{F}(\Phi_i^{(p)}) : \Phi_i^{(p)} \text{ satisfies the constraints imposed by the matrix } H_p \}.$$

Thus, to define the Bellman equation, we introduce a special kind of matrices, establishing relationships between any pair of consecutive entries $(p - 1, H_{p-1})$ and (p, H_p) .

A four-dimensional nonnegative integer-valued matrix

$$X = \|x_{\nu_1, \nu_2}^{c, \mathbf{d}}\|, \quad (c = \overline{0, q}, \nu_1, \nu_2 = \overline{1, \mathcal{K}}, \mathbf{d} = \overline{0, \bar{r}})$$

is called a *transition matrix* relating some entries $(p - 1, H_{p-1})$ and (p, H_p) , if $x_{\nu_1, \nu_2}^{c, \mathbf{d}}$ coincides with the number of partial concatenation profiles, each of them covers exactly c units of active demand, visits the depot exactly

\mathbf{d} times and has the same tuples σ_{ν_1} and σ_{ν_2} at positions $p - 1$ and p , respectively.

By construction, an arbitrarily transition matrix X fulfills the following obvious constraints

$$\sum_{\nu_1=1}^{\mathcal{K}} x_{\nu_1, \nu_2}^{c, \mathbf{d}} = h_{\nu_2, c, \mathbf{d}}^p, \quad (\nu_2 = \overline{1, \mathcal{K}}, \quad c = \overline{0, q}, \quad \mathbf{d} = \overline{0, \bar{r}}), \quad (4.37)$$

$$\sum_{\nu_1=0}^{\mathcal{K}} x_{\nu_1, 0}^{c, \mathbf{d}} = h_{0, c, \mathbf{d}}^p, \quad (c = \overline{0, q}, \quad \mathbf{d} = \overline{0, \bar{r}}). \quad (4.38)$$

The pseudo-code of these version of the internal dynamic program is presented in Algorithm 5.

Case 2: stitching the auxiliary routes enclosed in the cluster C . In this subsection, we discuss Algorithm 6, which is intended to construct a family $\Phi_{k_{\mathfrak{e}}+1}$ of the additional closed routes related to the last resource row $A_{k_{\mathfrak{e}}+1}$. To ensure the closedness of these routes, we append our dynamic program with an additional $(\bar{r} + 1)$ -th position (which comes as a fictitious copy of the first one). In addition, we should take into account that some routes can be padded with the empty segment σ_0 .

To make them closed, we should keep their last non-empty segments. To this end, we slightly modify the structure of each matrix H_p . Namely, for each p , c , and \mathbf{d} , we consider the entry $h_{0, c, \mathbf{d}}^p$ as a \mathcal{K} -dimensional vector $(h_{0, c, \mathbf{d}}^p(1), \dots, h_{0, c, \mathbf{d}}^p(\mathcal{K}))$, such that, for any $\nu > 0$, $h_{0, c, \mathbf{d}}^p(\nu)$ denotes the number of partial concatenation profiles, whose last segment is σ_0 and the last non-empty segment is σ_ν . Notice that, for any $\nu \geq 1$, the entries $h_{\nu, c, \mathbf{d}}^p$ retain their meanings (introduced in Case 1). Thus, an entry $(\bar{r} + 1, H_{\bar{r}+1})$ is feasible if the following equations

$$h_{\nu, c, \mathbf{d}}^{\bar{r}+1} > 0 \Rightarrow (c > 0) \wedge (\mathbf{d} > 0) \quad (\nu = \overline{0, \mathcal{K}}), \quad (4.41)$$

$$\sum_{c=1}^q \sum_{\mathbf{d}=1}^{\bar{r}} (h_{0, c, \mathbf{d}}^{\bar{r}+1}(\nu) + h_{\nu, c, \mathbf{d}}^{\bar{r}+1}) = h_{\nu, t_\nu, \mathbf{d}_\nu}^1 \quad (\nu = \overline{1, \mathcal{K}}) \quad (4.42)$$

Algorithm 6 Internal Dynamic Program for enclosed routes in cluster C

Input: a resource row $A_{k_{\mathfrak{e}}+1}$.

Output: a minimum cost family $\Phi_{k_{\mathfrak{e}}+1}$ of concatenation profiles compatible with the resource row $A_{k_{\mathfrak{e}}+1}$.

1: base case: the only feasible entry $(1, H_1)$, where

$$h_{\nu,c,\mathbf{d}}^1 = \begin{cases} a_{k_{\mathfrak{e}}+1,\nu}^1, & \text{if } \nu > 0, c = t_{\nu}, \text{ and } \mathbf{d} = \mathbf{d}_{\nu} \\ 0, & \text{otherwise} \end{cases}$$

and $\bar{D}(1, H_1) = 0$

2: recursive step: assume that all feasible entries are computed for any $p' < p$

3: **for each** H_p compatible with the p -th column of the resource row $A_{k_{\mathfrak{e}}+1}$ **do**

4: apply the Bellman equation to compute an entry (p, H_p)

$$\begin{aligned} \bar{D}(p, H_p) = \min_{X=\|x_{\nu_1,\nu_2}^{c,\mathbf{d}}\|} \{ & \bar{D}(p-1, H_{p-1}(X)) + \\ & \sum_{\nu_1=1}^{\mathcal{K}} \sum_{\nu_2=0}^{\mathcal{K}} \sum_{c=0}^q \sum_{\mathbf{d}=0}^{\bar{r}} x_{\nu_1,\nu_2}^{c,\mathbf{d}} \text{conn}(\sigma_{\nu_1}, \sigma_{\nu_2}) \}, \end{aligned} \quad (4.39)$$

where the minimization is carried out over the feasible entries $(p-1, H_{p-1}(X))$ only. In turn, $H_{p-1}(X)$ is obtained by the following formulas

$$\begin{aligned} h_{0,c,\mathbf{d}}^{p-1}(\nu) &= x_{0,0}^{c,\mathbf{d}}(\nu), \quad (\nu = \overline{1, \mathcal{K}}, \quad c = \overline{0, q}, \quad \mathbf{d} = \overline{0, \bar{r}}), \\ h_{\nu_1,c,\mathbf{d}}^{p-1} &= \sum_{\nu_2=0}^{\mathcal{K}} \{ x_{\nu_1,\nu_2}^{c+t_{\nu_2}, \mathbf{d}+\mathbf{d}_{\nu_2}} : c+t_{\nu_2} \leq q \}, (\nu_1 > 0). \end{aligned}$$

If at least one such an entry is found, then the result is stored in (p, H_p)

5: **end for**

6: **for each** feasible matrix $H_{\bar{r}+1}$ **do**

7: compute an entry $(\bar{r}+1, H_{\bar{r}+1})$ by the Bellman Equation (4.39) minimizing over the feasible entries $(\bar{r}, H_{\bar{r}}(X))$, where $H_{\bar{r}}(X)$ is calculated as follows

$$\begin{aligned} h_{0,c,\mathbf{d}}^{\bar{r}}(\nu_1) &= \sum_{\nu_2=1}^{\mathcal{K}} x_{\nu_1,\nu_2}^{c,\mathbf{d}}, \quad \nu_1 = \overline{1, \mathcal{K}}, \quad c = \overline{1, q}, \quad \mathbf{d} = \overline{1, \bar{r}}, \\ h_{\nu_1,c,\mathbf{d}}^{\bar{r}} &= \sum_{\nu_2=1}^{\mathcal{K}} x_{\nu_1,\nu_2}^{c,\mathbf{d}}, \quad (\nu_1 > 0), \quad c = \overline{1, q}, \quad \mathbf{d} = \overline{1, \bar{r}}. \end{aligned}$$

8: **end for**

9: **if** there are no feasible entries $(\bar{r}+1, H_{\bar{r}+1})$ or $\inf\{\bar{D}(\bar{r}+1, H_{\bar{r}+1})\} = \infty$ **then**

10: output ‘no profile families compatible with $A_{k_{\mathfrak{e}}+1}$ ’.

11: **else**

12: the cost of the desired family $\Phi_{k_{\mathfrak{e}}+1}$ is contained within the entry

$$(\bar{r}+1, H_{\bar{r}+1}^*) = \arg \min\{\bar{D}(\bar{r}+1, H_{\bar{r}+1}) : (\bar{r}+1, H_{\bar{r}+1}) \text{ is feasible}\}. \quad (4.40)$$

13: output the optimal solution $\Phi_{k_{\mathfrak{e}}+1}$, which can be obtained from (4.40) by backtracking.

14: **end if**

are valid.

Further, for any $p \leq \bar{r}$, a matrix H_p is compatible with the p -th column of the resource row $A_{k_{\mathfrak{e}}+1}$ if

$$\sum_{c=0}^q \sum_{\mathbf{d}=0}^{\bar{r}} h_{\nu,c,\mathbf{d}}^p = a_{k_{\mathfrak{e}}+1,\nu}^p, \quad (\nu = \overline{1, \mathcal{K}}), \quad (4.43)$$

$$\sum_{c=0}^q \sum_{\mathbf{d}=0}^{\bar{r}} \sum_{\nu=1}^{\mathcal{K}} h_{0,c,\mathbf{d}}^p(\nu) = a_{k_{\mathfrak{e}}+1,0}^p \quad (4.44)$$

are valid.

By the similar way, we transform transition matrices X . Keeping the sense of all the entries $x_{\nu_1,\nu_2}^{c,\mathbf{d}}$ for the case of $\nu_1 > 0$, for an arbitrary c and \mathbf{d} , we replace the entry $x_{0,0}^{c,\mathbf{d}}$ with the vector $(x_{0,0}^{c,\mathbf{d}}(1), \dots, x_{0,0}^{c,\mathbf{d}}(\mathcal{K}))$, such that $x_{0,0}^{c,\mathbf{d}}(\nu)$ denotes the number of partial concatenation profiles, each of them covers exactly c units of active demand, visits the depot \mathbf{d} times, has the same tuple σ_0 at last two positions, and the same last non-empty tuple σ_ν , respectively. Thus, any transition matrix X should fulfill the following constraints (similar to equations (4.37)-(4.38))

$$\begin{aligned} \sum_{\nu_1=1}^{\mathcal{K}} x_{\nu_1,\nu_2}^{c,\mathbf{d}} &= h_{\nu_2,c,\mathbf{d}}^p, \quad (\nu_2 = \overline{1, \mathcal{K}}, \quad c = \overline{0, q}, \quad \mathbf{d} = \overline{0, \bar{r}}), \\ x_{0,0}^{c,\mathbf{d}}(\nu) + x_{\nu,0}^{c,\mathbf{d}} &= h_{0,c,\mathbf{d}}^p(\nu), \quad (\nu = \overline{1, \mathcal{K}}, \quad c = \overline{0, q}, \quad \mathbf{d} = \overline{0, \bar{r}}). \end{aligned}$$

The pseudo-code of the internal dynamic programming for resource row $A_{k_{\mathfrak{e}}+1}$ is presented in Algorithm 6.

Thus, we conclude a detailed algorithmic analysis of option (i) mentioned in Remark 2. It can be shown that this analysis can be easily extended to all other options. Indeed, option (ii) (considered in detail in [Khachay et al., 2020b]) can be obtained by setting $\mathbf{d} \equiv 0$, option (iii) by $m_i \equiv 1$, whilst the last option (iv) corresponds to the case of $\mathbf{d} \equiv 0$ and $m_i \equiv 1$.

4.4 Complexity bounds

In this section, we establish an upper time complexity bound for the proposed scheme, which can be easily obtained as a product of the size of dynamic programming lookup table and the maximum time consumed during processing each its entry.

Since the entries of the master dynamic program lookup table are indexed by couples (C, \mathfrak{C}) , where C is a cluster and \mathfrak{C} is an associated configuration, respectively, the size of this table does not exceed the amount of clusters scaled by the maximum number of configurations, which can be associated with each of them.

Taking into account that we have at most $O(n \log(n/\varepsilon))$ clusters and, there are at most $(2m^2q)^\gamma$ and $(n \cdot r)^{2m^2O(L \log q)}$ options to assign to each of them an unrounded or rounded configuration, respectively, we obtain the following upper bound

$$O(n \log(n/\varepsilon)) \cdot \mathfrak{C}_{\max}$$

where

$$\mathfrak{C}_{\max} = (n \cdot r)^{2m^2O(L \log q)} + (2m^2q)^\gamma = (n \cdot r)^{2m^2O(L \log q)},$$

since $m = r = O\left(\left(\frac{d(\log n - \log \varepsilon)}{\varepsilon}\right)^d\right)$, $L = O(\log n - \log \varepsilon)$, $\gamma = \left(\frac{L+1}{\varepsilon}\right)^{2d}$, and $d > 1$ is a fixed constant.

Further, establish the complexity bound for computing of an arbitrary entry (C, \mathfrak{C}) of this lookup table. Obviously, the running time of such computing is maximized in the case of option (i) mentioned in Remark 2. Therefore, in the sequel, we restrict ourselves to this option. To proceed with such a computation, we apply Algorithm 4 to each possible combination (4.26) of child configurations. Those combinations are at most $(\mathfrak{C}_{\max})^K$.

In turn, Algorithm 4 enumerates all feasible resource matrices A . The amount N_A of such matrices does not exceed

$$(n \cdot r)^{(k_{\mathcal{E}}+1)(\mathcal{K}+1)\bar{r}}.$$

Given by any feasible matrix A , Algorithm 4 successively applies Algorithm 5 to process all of the resource rows A_i for $i = \overline{1, k_{\mathcal{E}}}$ and Algorithm 6 for the last row $A_{k_{\mathcal{E}}+1}$.

Therefore, the running time of Algorithm 4 does not exceed the complexity bound for Algorithm 6 scaled by the factor $N_A \cdot (k_{\mathcal{E}} + 1)$, since Algorithm 6 is more time-consuming than Algorithm 5.

Next, the time complexity of Algorithm 6 is determined by the number of entries in the internal dynamic programming lookup table and the processing time for each entry, i.e.

$$(\bar{r} + 1) \cdot (n \cdot r)^{2\mathcal{K}q\bar{r}} \times (n \cdot r)^{4\mathcal{K}^2q\bar{r}} = (n \cdot r)^{O(\mathcal{K}^2 \cdot q \cdot r)},$$

since

$$\bar{r} = K \cdot r = 2^{O(d)} \cdot r.$$

Finally, by combining all the terms, we obtain the desired time complexity bound

$$\text{poly}(n) \cdot ((n \cdot r)^{2m^2 L \log q})^{2^{O(d)}} \cdot (n \cdot r)^{O(m^4 L^2 \log^2 q + \mathcal{K}^2 q r)} = \text{poly}(n) \cdot n^{O(m^5 L^2 q \log^2 q)}.$$

Leveraging the techniques proposed in [Das and Mathieu, 2015] and [Talwar, 2004], we derandomize our scheme in polynomial time.

Theorem 8. *For the CVRP in a metric space of an arbitrary doubling dimension $d > 1$, an $(1 + O(\varepsilon))$ -approximate solution can be found by the randomized approximation algorithm within time $\text{poly}(n) \cdot n^{O(m^5 L^2 q \log^2 q)}$,*

where $m = O\left(\left(\frac{d(\log n - \log \varepsilon)}{\varepsilon}\right)^d\right)$, and $L = O(\log n - \log \varepsilon)$. The algorithm can be derandomized efficiently.

The proposed scheme is QPTAS any time when $q = O(\text{polylog}(n))$.

Conclusion

4.5 Summary

This research was motivated by the challenges in the global economy and the optimization techniques that are widely employed to overcome those challenges. According to the literature, there are several combinatorial optimization problems employed to model the aforementioned techniques, including TSP, VRP and their variants. Due to the NP-hard nature of those problems, there are known three major solution approaches, namely optimal (or exact) algorithms, heuristics and meta-heuristics, and approximation algorithms with accuracy guarantees. In this thesis, we propose a novel algorithmic results for each solution approach and for a specific deterministic routing problems with various constraints.

In particular, in Chapter 2, we address the Precedence Constrained Generalized Traveling Salesman Problem (PCGTSP) both in terms of the polyhedral study and algorithmic analysis. By evolving the results previously introduced for PCATSP, we propose several novel families of the valid inequalities. Then, we establish dimension of the PCGTS polytope and proved sufficient conditions for the predecessor/successor inequalities to be facet-inducing. Further, we offer a sequence of novel formulations for the PCGTSP and propose the first branch-and-cut algorithm relying on these formulations. In the numerical evaluation, we report the most efficient formulations in terms of LP-relaxation bounds and suggested several well-collaborating variants of the proposed branch-and-cut. As a result, the

number of PCGTSPLIB instances solved to optimality became 24 out of 40, where for 12 instances it was done for the first time. In addition, the obtained results confirmed the importance of the predecessor/successor inequalities for the PCGTSP, both for LP-relaxation and branch-and-cut framework.

In Chapter 3, we propose a general framework that allows us to solve the discrete Cutting Path Problem by reducing all its known settings to the auxiliary instances of the Precedence Constrained Generalized Traveling Salesman Problem. As it follows from the obtained theoretical results, the proposed reduction is efficient (polynomial time) and cost preserving, that is an arbitrary feasible solution of the auxiliary problem induces the corresponding same cost cutting path for the CPP, which can be efficiently decoded from this solution. In turn, the results of the numerical evaluation carried out against real industrial instances show that close to optimal or even optimal solutions of these auxiliary instances can be obtained efficiently as well. In particular, the performed statistical analysis of the results obtained by PCGLNS heuristic in comparison with solutions of the branch-and-cut algorithm shows the capability of this heuristic to provide high quality solutions in a few minutes. Therefore, we believe that the contribution of this chapter is promising in the field of cutting problems.

Finally, the results of Chapter 4 extend the famous approximation framework proposed by A. Das and C. Mathieu for the Euclidean Capacitated Vehicle Routing Problem to the case of metric spaces of a fixed doubling dimension. To establish quasi-polynomial time upper bound for our scheme, we replace exhaustive search in the initial algorithm by the novel internal dynamic program, which ensures that the resulting approximation scheme becomes QPTAS for an arbitrary fixed doubling dimension $d > 1$, at least for $q = \text{polylog}(n)$.

4.6 Open questions and Perspectives

Based on the related work and the obtained results of this thesis, we can propose the following research perspectives. First of all, we should mention the further study of the dynamic versions of GTSP and VRP as those variants correspond well to the changing environments of the real-world optimization applications. An example of such an environment is the traffic jams or car accidents on the roads, thus the overall travel time and fuel expenses could increase. Another potential example is the last minute preference of the customer to change the delivery time and/or their location, therefore our route should be restructured.

Dynamic GTSP and VRP are obviously more complicated than their static variants considered in this thesis due to the variation of number of nodes and the weights of the arcs of the graph G . The uncertainty of different parameters related to the real life applications, however, potentially could be handled by various estimation methods, such as fuzzy estimation or stochastic estimation. Nevertheless, the uncertain GTSP and VRP are viable research directions for future work.

We understand that our results for the CPP can be considered only as a proof-of-concept. It seems useful to incorporate industrial implementation of all the proposed reductions and numerical algorithms into state-of-the-art CAD/CAM systems as a plugin, which will simplify the job for tool operators, allowing them to assign an optimal (close to optimal) cutting path for a current cutting template in almost a real time. We believe that the implementation of such plugin as well as its further testing and maintenance can be a viable direction for future work.

Regarding to the approximation algorithms, it seems interesting to take into account asymmetric versions of the considered problems, relying on the recent breakthrough results by (Svensson et al. [2020]) and (Traub

and Vygen [2022]) for the Asymmetric Traveling Salesman Problem with triangle inequality.

Speaking of real-world applications, one could distinguish the problem related to the design of reliable production processes. The recent changes in the hyper-competitive marketplace environment call for a considerable effort to research and implementation of sustainable and resilient supply chains. It is possible, however, to take advantage of stochastic models, where possible disruptions of a transportation network are described in terms of given scenarios. However, such an approach deteriorates if the interruption in question was not anticipated. Therefore, in some cases, a more suitable solution would be the one that minimizes the risks and remain preferably deterministic. To the best of our knowledge, those assured risk mitigation techniques for the deterministic production processes remain rather weakly studied. Therefore, a proposal of a novel deterministic modeling framework aimed to protect the manufacturing system from various non-anticipated production or logistic failures is a valid direction for future research.

The final research direction is the question of whether the CVRP can be formulated in a metric space of an arbitrary fixed doubling dimension without any restriction on the capacity growth be approximated by QPTAS.

Bibliography

- Ittai Abraham, Yair Bartal, and Ofer Neiman. Advances in metric embedding theory. *Advances in Mathematics*, 228(6):3026 – 3126, 2011. ISSN 0001-8708. doi: 10.1016/j.aim.2011.08.003.
- Anna Adamaszek, Artur Czumaj, and Andrzej Lingas. PTAS for k-tour cover problem on the plane for moderately large values of k . *International Journal of Foundations of Computer Science*, 21(6):893–904, 2010. doi: 10.1142/S0129054110007623.
- Olasile Babatunde Adedoyin and Emrah Soykan. Covid-19 pandemic and online learning: the challenges and opportunities. *Interactive Learning Environments*, 31(2):863–875, 2023. doi: 10.1080/10494820.2020.1813180.
- Claudia Archetti and M. Grazia Speranza. A survey on metaheuristics for routing problems. *EURO Journal on Computational Optimization*, 2(4):223–246, 2014. doi: 10.1007/s13675-014-0030-7.
- Florian Arnold and Kenneth Sörensen. Knowledge-guided local search for the vehicle routing problem. *Computers & Operations Research*, 105:32–46, 2019. doi: 10.1016/j.cor.2019.01.002.
- Sanjeev Arora. Polynomial Time Approximation Schemes for Euclidean Traveling Salesman and other geometric problems. *Journal of the ACM*, 45:753–782, 1998.
- Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: a new characterization of NP. *Journal of the ACM*, 45:70–122, 1 1998. doi: 10.1145/273865.273901.
- Tetsuo Asano, Naoki Katoh, Hisao Tamaki, and Takeshi Tokuyama. Covering points in the plane by k-tours: Towards a polynomial time approximation scheme for general k . In *Proceedings of the Twenty-ninth Annual ACM Symposium on Theory of Computing*, STOC '97, pages 275–283, New York, NY, USA, 1997. ACM. ISBN 0-89791-888-6. doi: 10.1145/258533.258602.
- Norbert Ascheuer, Michael Jünger, and Gerhard Reinelt. A branch & cut algorithm for the asymmetric traveling salesman problem with precedence constraints. *Computational Optimization and Applications*, 17(1):61–84, 2000. doi: 10.1023/A:1008779125567.
- Sergey Avdoshin and Ekaterina Beresneva. Local search metaheuristics for capacitated vehicle routing problem: a comparative study. *Proceedings of the Institute for System Programming of RAS*, 31:121–138, 2019. doi: 10.15514/ISPRAS-2019-31(4)-8.
- Egon Balas and Neils Simonetti. Linear time dynamic-programming algorithms for new classes of restricted TSPs: A computational study. *INFORMS J. on Computing*, 13(1):56–75, February 2001. ISSN 1526-5528. doi: 10.1287/ijoc.13.1.56.9748.
- Egon Balas, Matteo Fischetti, and William Pulleyblank. The precedence-constraint asymmetric traveling salesman polytope. *Math. Program.*, 68:241–265, 03 1995. doi: 10.1007/BF01585767.

- Roberto Baldacci, Enrico Bartolini, and Gilbert Laporte. Some applications of the generalized vehicle routing problem. *Journal of the Operational Research Society*, 61(7):1072–1077, 2010. doi: 10.1057/jors.2009.51.
- Jørgen Bang-Jensen and Gregory Z. Gutin. *Digraphs: Theory, Algorithms and Applications*. Springer London, 2009. ISBN 978-1-84800-998-1. doi: 10.1007/978-1-84800-998-1_1.
- Yair Bartal, Lee-Ad Gottlieb, and Robert Krauthgamer. The Traveling Salesman Problem: Low-dimensionality implies a polynomial time approximation scheme. *SIAM Journal on Computing*, 45(4):1563–1581, 2016. doi: 10.1137/130913328.
- Amariah Becker, Philip N. Klein, and Aaron Schild. A PTAS for bounded-capacity vehicle routing in planar graphs. In Zachary Friggstad, Jörg-Rüdiger Sack, and Mohammad R Salavatipour, editors, *Algorithms and Data Structures*, pages 99–111, Cham, 2019. Springer International Publishing. doi: 10.1007/978-3-030-24766-9_8.
- Julia A. Bennell and Jose F. Oliveira. The geometry of nesting problems: A tutorial. *European Journal of Operational Research*, 184(2):397–415, 2008. doi: 10.1016/j.ejor.2006.11.038.
- Norman Biggs. *Algebraic Graph Theory*. Cambridge Tracts in Mathematics 67. Cambridge University Press, 2 edition, 1974. ISBN 052120335X,9780521203357.
- Marco Caserta and Stefan Voß. A hybrid algorithm for the dna sequencing problem. *Discrete Applied Mathematics*, 163:87–99, 2014. doi: doi.org/10.1016/j.dam.2012.08.025.
- Kenneth Castelino, Roshan D’Souza, and Paul K. Wright. Toolpath optimization for minimizing airtime during machining. *Journal of Manufacturing Systems*, 22(3):173 – 180, 2003. doi: 10.1016/S0278-6125(03)90018-5.
- T.-H. Hubert Chan and Shaofeng H.-C. Jiang. Reducing curse of dimensionality: Improved ptas for tsp (with neighborhoods) in doubling metrics. *ACM Trans. Algorithms*, 14(1):18, 2018. doi: 10.1145/3158232.
- Chen, Gui, Ding, and Zhou. Optimization of transportation routing problem for fresh food by improved ant colony algorithm based on tabu search. *Sustainability*, 11, 2019. doi: 10.3390/su11236584.
- Alexander Chentsov. Problem of successive megalopolis traversal with the precedence conditions. *Automation and Remote Control*, 75(4):728–744, 2014. doi: 10.1134/S0005117914040122.
- Alexander Chentsov, Michael Khachai, and Daniel Khachai. An exact algorithm with linear complexity for a problem of visiting megalopolises. *Proceedings of the Steklov Institute of Mathematics*, 295(1):38–46, 2016. ISSN 1531-8605. doi: 10.1134/S0081543816090054.
- Alexander G. Chentsov, Pavel A. Chentsov, Alexander A. Petunin, and Alexander N. Sesekin. Model of megalopolises in the tool path optimisation for CNC plate cutting machines. *International Journal of Production Research*, 56(14):4819–4830, 2018a. doi: 10.1080/00207543.2017.1421784.
- Alexander G. Chentsov, Pavel A. Chentsov, Alexander A. Petunin, and Alexander N. Sesekin. Model of megalopolises in the tool path optimisation for CNC plate cutting machines. *International Journal of Production Research*, 56(14):4819–4830, 2018b. doi: 10.1080/00207543.2017.1421784.

- Luiz Henrique Cherri, Maria Ant3nia Carravilla, Cristina Ribeiro, and Franklina Maria Bragion Toledo. Optimality in nesting problems: New constraint programming models and a new global constraint for non-overlap. *Operations Research Perspectives*, 6:100125, 2019. doi: 10.1016/j.orp.2019.100125.
- Tanya Chervenkova and Dmitry Ivanov. Adaptation strategies for building supply chain viability: A case study analysis of the global automotive industry re-purposing during the covid-19 pandemic. *Transportation Research Part E: Logistics and Transportation Review*, 177:103249, 2023. ISSN 1366-5545. doi: 10.1016/j.tre.2023.103249.
- Hannah Hyunah Cho, Vladimir Strezov, and Tim J. Evans. A review on global warming potential, challenges and opportunities of renewable hydrogen production technologies. *Sustainable Materials and Technologies*, 35:e00567, 2023. ISSN 2214-9937. doi: 10.1016/j.susmat.2023.e00567.
- Nicos Christofides. Worst-case analysis of a new heuristic for the Traveling Salesman Problem. In *Symposium on New Directions and Recent Results in Algorithms and Complexity*, page 441, 1976.
- Philippe Chr3tienne, Pierre Fouilhoux, Eric Gourdin, and Jean-Mathieu Segura. The location-dispatching problem: Polyhedral results and content delivery network design. *Discrete Applied Mathematics*, 164:68–85, 2014. doi: 10.1016/j.dam.2012.05.001.
- Andre A. Cire and Willem-Jan van Hoeve. Multivalued decision diagrams for sequencing problems. *Operations Research*, 61(6):1411–1428, 2013. doi: 10.1287/opre.2013.1221.
- Fran3ois Clautiaux, J3r3my Guillot, and Pierre Pesneau. Exact approaches for solving a covering problem with capacitated subtrees. *Computers & Operations Research*, 105:85–101, 2019. ISSN 0305-0548. doi: 10.1016/j.cor.2019.01.008.
- Nicol3s Clavijo-Buritica, Laura Triana-Sanchez, and John Willmer Escobar. A hybrid modeling approach for resilient agri-supply network design in emerging countries: Colombian coffee supply chain. *Socio-Economic Planning Sciences*, 85:101431, 2023. doi: 10.1016/j.seps.2022.101431.
- Rosario Cuda, Gianfranco Guastaroba, and Maria G. Speranza. A survey on two-echelon routing problems. *Computers and Operations Research*, 55:185–199, 2015. doi: 10.1016/j.cor.2014.06.008.
- Daniel Cuellar-Usaqu3n, Alejandro Palacio, Emilia Ospina, Marcelo Botero, and David 3lvarez Mart3nez. Modeling and solving the endpoint cutting problem. *International Transactions in Operational Research*, 30(2):800–830, 2023. doi: 10.1111/itor.13091.
- George Dantzig, Delber R. Fulkerson, and Selmer M. Johnson. Solution of a large-scale traveling-salesman problem. *Journal of the Operations Research Society of America*, 2(4):393–410, 1954. doi: 10.1287/opre.2.4.393.
- George B. Dantzig and John H. Ramser. The truck dispatching problem. *Management science*, 6(1):80–91, 1959.
- Aparna Das and Claire Mathieu. A quasipolynomial time approximation scheme for Euclidean capacitated vehicle routing. *Algorithmica*, 73:115–142, 2015. doi: 10.1007/s00453-014-9906-4.
- B. A. Davey and H. A. Priestley. *Introduction to Lattices and Order, Second Edition*. Cambridge University Press, 2 edition, 2002. ISBN 0521784514,9780521784511.

- Emrah Demir, Katy Huckle, Aris Syntetos, Andrew Lahy, and Mike Wilson. *Vehicle Routing Problem: Past and Future*, pages 97–117. Springer International Publishing, Cham, 2019. ISBN 978-3-030-14493-7. doi: 10.1007/978-3-030-14493-7{_}7.
- Martin Desrochers and Gilbert Laporte. Improvements and extensions to the miller-tucker-zemlin subtour elimination constraints. *Operations Research Letters*, 10(1):27–36, 1991. ISSN 0167-6377. doi: 10.1016/0167-6377(91)90083-2.
- Reginald Dewil, Pieter Vansteenwegen, and Dirk Cattrysse. Construction heuristics for generating tool paths for laser cutters. *International Journal of Production Research*, 52(20): 5965–5984, 2014. doi: 10.1080/00207543.2014.895064.
- Reginald Dewil, Pieter Vansteenwegen, Dirk Cattrysse, Manuel Laguna, and Thomas Vossen. An improvement heuristic framework for the laser cutting tool path problem. *International Journal of Production Research*, 53(6):1761–1776, 2015. doi: 10.1080/00207543.2014.959268.
- Reginald Dewil, Pieter Vansteenwegen, and Dirk Cattrysse. A review of cutting path algorithms for laser cutters. *The International Journal of Advanced Manufacturing Technology*, 87(5): 1865–1884, 2016. doi: 10.1007/s00170-016-8609-1.
- Harald Dyckhoff. A typology of cutting and packing problems. *European Journal of Operational Research*, 44(2):145–159, 1990. ISSN 0377-2217. doi: [https://doi.org/10.1016/0377-2217\(90\)90350-K](https://doi.org/10.1016/0377-2217(90)90350-K). Cutting and Packing.
- Hasan Dünder, Mine Ömürgönülgen, and Mehmet Soysal. A review on sustainable urban vehicle routing. *Journal of Cleaner Production*, 285:125444, 2021. doi: 10.1016/j.jclepro.2020.125444.
- Laureano F. Escudero. An inexact algorithm for the sequential ordering problem. *European Journal of Operational Research*, 37(2):236–249, 1988. doi: 10.1016/0377-2217(88)90333-5.
- Matteo Fischetti, Juan José Salazar González, and Paolo Toth. The symmetric generalized traveling salesman polytope. *Networks*, 26:113–123, 09 1995. doi: 10.1002/net.3230260206.
- Matteo Fischetti, Juan José Salazar González, and Paolo Toth. A branch-and-cut algorithm for the symmetric Generalized Traveling Salesman Problem. *Operations Research*, 45(3): 378–394, 1997. doi: 10.1287/opre.45.3.378.
- Sana Frifita and Malek Masmoudi. VNS methods for home care routing and scheduling problem with temporal dependencies, and multiple structures and specialties. *International Transactions in Operational Research*, 27(1):291–313, 2020. doi: 10.1111/itor.12604.
- Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979. ISBN 0716710447.
- Michel Gendreau and Jean-Yves Potvin. *Handbook of Metaheuristics*. International Series in Operations Research & Management Science 272. Springer International Publishing, 3rd edition, 2019. ISBN 978-3-319-91085-7; 978-3-319-91086-4.
- Paul C. Gilmore and Ralph E. Gomory. A linear programming approach to the cutting-stock problem. *Operations Research*, 9(6):849–859, 1961. doi: 10.1287/opre.9.6.849.
- Fred Glover, Gregory Gutin, Anders Yeo, and Alexey Zverovich. Construction heuristics for the asymmetric tsp. *European Journal of Operational Research*, 129(3):555–568, 2001. doi: doi.org/10.1016/S0377-2217(99)00468-3.

- Lisa Baidu Gobio-Thomas, Muhamed Darwish, and Valentina Stojceska. Environmental impacts of solar thermal power plants used in industrial supply chains. *Thermal Science and Engineering Progress*, 38:101670, 2023. doi: 10.1016/j.tsep.2023.101670.
- Luis Gouveia and Pierre Pesneau. On extended formulations for the precedence constrained asymmetric traveling salesman problem. *Networks*, 48(2):77–89, 2006. doi: 10.1002/net.20122.
- Luis Gouveia and Jose M. Pires. The asymmetric travelling salesman problem and a reformulation of the miller–tucker–zemlin constraints. *European Journal of Operational Research*, 112(1):134–146, 1999. doi: 10.1016/S0377-2217(97)00358-5.
- Luis Gouveia and Jose M. Pires. The asymmetric travelling salesman problem: on generalizations of disaggregated miller–tucker–zemlin constraints. *Discrete Applied Mathematics*, 112(1):129–145, 2001. doi: 10.1016/S0166-218X(00)00313-9.
- Luis Gouveia and Mario Ruthmair. Load-dependent and precedence-based models for pickup and delivery problems. *Computers & Operations Research*, 63:56–71, 2015. ISSN 0305-0548. doi: 10.1016/j.cor.2015.04.008.
- Luis Gouveia, Pierre Pesneau, Mario Ruthmair, and Daniel Santos. Combining and projecting flow models for the (precedence constrained) asymmetric traveling salesman problem. *Networks*, 71(4):451–465, 2018. doi: 10.1002/net.21765.
- Anupam Gupta, Robert Krauthgamer, and James R. Lee. Bounded geometries, fractals, and low-distortion embeddings. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pages 534–543, 2003. doi: 10.1109/SFCS.2003.1238226.
- Gregory Gutin and Daniel Karapetyan. A memetic algorithm for the generalized traveling salesman problem. *Natural Computing*, 9(1):47–60, 2010. doi: 10.1007/s11047-009-9111-6.
- Gregory Gutin and Abraham P. Punnen. *The Traveling Salesman Problem and Its Variations*. Springer US, Boston, MA, 2007. ISBN 978-0-387-44459-8.
- Taner Güney. Solar energy, governance and co2 emissions. *Renewable Energy*, 184:791–798, 2022. doi: 10.1016/j.renene.2021.11.124.
- Mordecai Haimovich and Alexander H. G. Rinnooy Kan. Bounds and heuristics for capacitated routing problems. *Mathematics of Operations Research*, 10(4):527–542, 1985. doi: 10.1287/moor.10.4.527.
- Guk-chan Han and Suck-joo Na. A study on torch path planning in laser cutting processes part 2: Cutting path optimization using simulated annealing. *Journal of Manufacturing Systems*, 18(2, Suppl. 1):62–70, 1999. ISSN 0278-6125. doi: 10.1016/S0278-6125(99)80027-2.
- Michael Held and Richard M. Karp. A dynamic programming approach to sequencing problems. *Journal of the Society for Industrial and Applied Mathematics*, 10(1):196–210, 1962. ISSN 03684245.
- Keld Helsgaun. An effective implementation of the lin–kernighan traveling salesman heuristic. *European Journal of Operational Research*, 126(1):106–130, 2000. doi: 10.1016/S0377-2217(99)00284-2.
- Keld Helsgaun. Solving the equality Generalized Traveling Salesman Problem using the Lin–Kernighan–Helsgaun algorithm. *Mathematical Programming Computation*, pages 1–19, 2015.

- Jeffrey W. Herrmann and David R. Delalio. Algorithms for sheet metal nesting. *IEEE Transactions on Robotics and Automation*, 17(2):183–90, 2001. doi: 10.1109/70.928563.
- Jeffrey Hoefft and Udatta S. Palekar. Heuristics for the plate-cutting traveling salesman problem. *IIE Transactions*, 29(9):719–731, 1997. doi: 10.1080/07408179708966382.
- Pedro Hokama, Flávio K. Miyazawa, and Eduardo C. Xavier. A branch-and-cut approach for the vehicle routing problem with loading constraints. *Expert Systems with Applications*, 47: 1–13, 2016. ISSN 0957-4174. doi: <https://doi.org/10.1016/j.eswa.2015.10.013>.
- Paul Hong, Sandeep Jagani, Phuoc Pham, and Euisung Jung. Globalization orientation, business practices and performance outcomes: an empirical investigation of b2b manufacturing firms. *Journal of Business & Industrial Marketing*, ahead-of-print(ahead-of-print), 2023. doi: 10.1108/JBIM-02-2021-0098.
- Qirut Hu, Zhiwei Lin, and Jianzhong Fu. A robust fast bridging algorithm for laser cutting. *The International Journal of Advanced Manufacturing Technology*, 121(3):2083–2094, 2022. doi: 10.1007/s00170-022-09465-w.
- Yi Huang, Chen Lin, Sibao Liu, and Heiwai Tang. Trade networks and firm value: Evidence from the u.s.-china trade war. *Journal of International Economics*, page 103811, 2023. doi: 10.1016/j.jinteco.2023.103811.
- Shinji Imahori, Motoki Kushiya, Takeru Nakashima, and Kokichi Sugihara. Generation of cutter paths for hard material in wire edm. *Journal of Materials Processing Technology*, 206(1):453–461, 2008. doi: 10.1016/j.jmatprotec.2007.12.039.
- Badar Alam Iqbal, Nida Rahman, and Jonathan Elimimian. The future of global trade in the presence of the sino-us trade war. *Economic and Political Studies*, 7(2):217–231, 2019. doi: 10.1080/20954816.2019.1595324.
- Arpan Kumar Kar, Shweta Kumari Choudhary, and Vinay Kumar Singh. How can artificial intelligence impact sustainability: A systematic literature review. *Journal of Cleaner Production*, 376:134120, 2022. doi: 10.1016/j.jclepro.2022.134120.
- Daniel Karapetyan and Gregory Gutin. Efficient local search algorithms for known and new neighborhoods for the generalized traveling salesman problem. *European Journal of Operational Research*, 219(2):234–251, 2012. ISSN 0377-2217. doi: 10.1016/j.ejor.2012.01.011.
- Daniil Khachai, Ruslan Sadykov, Olga Battaia, and Michael Khachay. Precedence constrained generalized traveling salesman problem: Polyhedral study, formulations, and branch-and-cut algorithm. *European Journal of Operational Research*, 309(2):488–505, 2023. doi: 10.1016/j.ejor.2023.01.039.
- Michael Y. Khachai and Yuri Y. Ogorodnikov. Haimovich - Rinnooy Kan Polynomial-Time Approximation Scheme for the CVRP in metric spaces of a fixed doubling dimension. *Trudy instituta matematiki i mekhaniki UrO RAN*, 25(4):235–248, 2019a. doi: 10.21538/0134-4889-2019-25-4-235-248.
- Michael Yu. Khachai and Roman D. Dubinin. Approximability of the vehicle routing problem in finite-dimensional euclidean spaces. *Proceedings of the Steklov Institute of Mathematics*, 297(1):117–128, 2017. ISSN 1531-8605. doi: 10.1134/S0081543817050133.

- Michael Yu. Khachai and Ekaterina D. Neznakhina. Approximation schemes for the generalized traveling salesman problem. *Proceedings of the Steklov Institute of Mathematics*, 299(1): 97–105, 2017. ISSN 1531-8605. doi: 10.1134/S0081543817090127.
- Michael Yu. Khachai and Yuri Ogorodnikov. Polynomial-time approximation scheme for the capacitated vehicle routing problem with time windows. *Proc. of the Steklov institute of Mathematics*, 307(suppl. 1):S51–S63, 2019b. doi: 10.1134/S0081543819070058.
- Michael Khachay and Roman Dubinin. *PTAS for the Euclidean Capacitated Vehicle Routing Problem in R^d* , volume 9869 of *LNCS*, pages 193–205. Springer International Publishing, Cham, 2016. ISBN 978-3-319-44914-2. doi: 10.1007/978-3-319-44914-2_16.
- Michael Khachay and Katherine Neznakhina. Complexity and approximability of the Euclidean Generalized Traveling Salesman Problem in grid clusters. *Annals of Mathematics and Artificial Intelligence*, 88(1):53–69, 2020. doi: 10.1007/s10472-019-09626-w.
- Michael Khachay and Yuri Ogorodnikov. Efficient PTAS for the Euclidean CVRP with Time Windows. In *Analysis of Images, Social Networks and Texts - 7th International Conference, AIST 2018*, volume 11179 of *LNCS*, pages 318–328, Cham, 2018. Springer International Publishing. ISBN 978-3-030-11026-0. doi: 10.1007/978-3-030-11027-7_30.
- Michael Khachay and Yuri Ogorodnikov. Approximation scheme for the capacitated vehicle routing problem with time windows and non-uniform demand. In *Mathematical Optimization Theory and Operations Research – 18th International conference (MOTOR 2019). Proceedings*, volume 11548 of *LNCS*, pages 309–327. Springer International Publishing, 2019. doi: 10.1007/978-3-030-22629-9_22.
- Michael Khachay and Helen Zaytseva. *Polynomial Time Approximation Scheme for Single-Depot Euclidean Capacitated Vehicle Routing Problem*, volume 9486 of *LNCS*, pages 178–190. Springer International Publishing, Cham, 2015. ISBN 978-3-319-26626-8. doi: 10.1007/978-3-319-26626-8_14.
- Michael Khachay, Andrei Kudriavtsev, and Alexander Petunin. PCGLNS: A heuristic solver for the Precedence Constrained Generalized Traveling Salesman Problem. *LNCS*, 12422: 196–208, 2020a. doi: 10.1007/978-3-030-62867-3_15.
- Michael Khachay, Yuri Ogorodnikov, and Daniel Khachay. An extension of the Das and Mathieu QPTAS to the case of polylog capacity constrained CVRP in metric spaces of a fixed doubling dimension (accepted). In *Mathematical Optimization Theory and Operations Research – 19th International conference (MOTOR 2020). Proceedings*, volume 12095 of *LNCS*. Springer International Publishing, 2020b.
- Michael Khachay, Yuri Ogorodnikov, and Daniel Khachay. Efficient approximation of the metric CVRP in spaces of fixed doubling dimension. *Journal of Global Optimization*, 80(3):679–710, 2021a. doi: 10.1007/s10898-020-00990-.
- Michael Khachay, Stanislav Ukolov, and Alexander Petunin. Problem-specific Branch-and-Bound algorithms for the Precedence Constrained Generalized Traveling Salesman Problem. *LNCS*, pages 136–148, 2021b. doi: 10.1007/978-3-030-91059-4_10.
- Tonni Agustiono Kurniawan, Mohd Hafiz Dzarfan Othman, Xue Liang, Hui Hwang Goh, Petros Gikas, Kok-Keong Chong, and Kit Wayne Chew. Challenges and opportunities for biochar to promote circular economy and carbon neutrality. *Journal of Environmental Management*, 332:117429, 2023. doi: 10.1016/j.jenvman.2023.117429.

- Gilbert Laporte. Fifty years of vehicle routing. *Transportation Science*, 43:408–416, 2009. doi: 10.1287/trsc.1090.0301.
- Gilbert Laporte and Frédéric Semet. Computational evaluation of a transformation procedure for the symmetric generalized traveling salesman problem. *INFOR: Information Systems and Operational Research*, 37(2):114–120, 1999.
- Eugene L. Lawler, Jan Karel Lenstra, and Alexander H. G. Rinnooy Kan. *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. Wiley Series in Discrete Mathematics & Optimization. Wiley, 1985. ISBN 0471904139,9780471904137.
- Moon-Kyu Lee and Ki-Bum Kwon. Cutting path optimization in CNC cutting processes using a two-step genetic algorithm. *International Journal of Production Research*, 44(24):5307–5326, 2006. doi: 10.1080/00207540600579615.
- Jiewu Leng, Weinan Sha, Baicun Wang, Pai Zheng, Cunbo Zhuang, Qiang Liu, Thorsten Wuest, Dimitris Mourtzis, and Lihui Wang. Industry 5.0: Prospect and retrospect. *Journal of Manufacturing Systems*, 65:279–295, 2022. doi: 10.1016/j.jmsy.2022.09.017.
- Jiewu Leng, Yuanwei Zhong, Zisheng Lin, Kailin Xu, Dimitris Mourtzis, Xueliang Zhou, Pai Zheng, Qiang Liu, J. Leon Zhao, and Weiming Shen. Towards resilience in industry 5.0: A decentralized autonomous manufacturing paradigm. *Journal of Manufacturing Systems*, 71: 95–114, 2023. doi: 10.1016/j.jmsy.2023.08.023.
- Ning Li, Liang Cheng, Chen Ji, Hui Chen, WanXuan Geng, and WeiMing Yang. Airport detection in remote sensing real-open world using deep learning. *Engineering Applications of Artificial Intelligence*, 122:106083, 2023a. doi: 10.1016/j.engappai.2023.106083.
- Wei Li, Hui Sun, Hao Dong, Yuqin Gan, and Lenny Koh. Outsourcing decision-making in global remanufacturing supply chains: The impact of tax and tariff regulations. *European Journal of Operational Research*, 304(3):997–1010, 2023b. doi: 10.1016/j.ejor.2022.05.016.
- Yi-Jing Liang and Zhi-Xing Luo. A survey of truck–drone routing problem: Literature review and research prospects. *Journal of the Operations Research Society of China*, 10(2):343–377, 2022. doi: 10.1007/s40305-021-00383-4.
- Giusy Macrina, Luigi Di Puglia Pugliese, Francesca Guerriero, and Gilbert Laporte. Drone-aided routing: A literature review. *Transportation Research Part C: Emerging Technologies*, 120: 102762, 2020. doi: 10.1016/j.trc.2020.102762.
- Tatiana A. Makarovskikh, Anatoly V. Panyukov, and Egor A. Savitskiy. Mathematical models and routing algorithms for economical cutting tool paths. *International Journal of Production Research*, 56(3):1171–1188, 2018. doi: 10.1080/00207543.2017.1401746.
- Udi Manber and Sharat Israni. Pierce point minimization and optimal torch path determination in flame cutting. *Journal of Manufacturing Systems*, 3(1):81–89, 1984. doi: 10.1016/0278-6125(84)90024-4.
- Karl Menger. *Ergebnisse eines Mathematischen Kolloquiums*, pages IX, 470. Springer Vienna, 1932. ISBN 978-3-7091-6470-9. doi: 10.1007/978-3-7091-6470-9.
- Clair E. Miller, Albert W. Tucker, and Richard A. Zemlin. Integer programming formulation of traveling salesman problems. *J. ACM*, 7(4):326–329, 1960. doi: 10.1145/321043.321046.

- Joseph S.B. Mitchell. Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric tsp, k-mst, and related problems. *SIAM Journal on Computing*, 28(4):1298–1309, 1999. doi: 10.1137/S0097539796309764.
- Jairo R. Montoya-Torres, Andrés Muñoz-Villamizar, and Christopher Mejia-Argueta. Mapping research in logistics and supply chain management during covid-19 pandemic. *International Journal of Logistics Research and Applications*, 26(4):421–441, 2023. doi: 10.1080/13675567.2021.1958768.
- Luís M. Moreira, José F. Oliveira, A. Miguel Gomes, and J. Soeiro Ferreira. Heuristics for a dynamic rural postman problem. *Computers & Operations Research*, 34(11):3281–3294, 2007. doi: 10.1016/j.cor.2005.12.008.
- Thomas L. Morin and Roy E. Marsten. Branch-and-bound strategies for dynamic programming. *Operations Research*, 24(4):611–627, 1976.
- Mohsen Mosayebi, Manbir Sodhi, and Thomas A. Wettergren. The traveling salesman problem with job-times (tspj). *Computers & Operations Research*, 129:105226, 2021. doi: doi.org/10.1016/j.cor.2021.105226.
- Jakub Nalepa and Miroslaw Blocho. Adaptive memetic algorithm for minimizing distance in the Vehicle Routing Problem with Time Windows. *Soft Computing*, 20(6):2309–2327, 2016. ISSN 1433-7479. doi: 10.1007/s00500-015-1642-4.
- Sina Nayeri, Zeinab Sazvar, and Jafar Heydari. Designing an iot-enabled supply chain network considering the perspective of the fifth industrial revolution: Application in the medical devices industry. *Engineering Applications of Artificial Intelligence*, 122:106113, 2023. doi: 10.1016/j.engappai.2023.106113.
- Mohammadreza Nazari, Afshin Oroojlooy, Martin Takáč, and Lawrence V. Snyder. Reinforcement learning for solving the vehicle routing problem. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS’18, page 9861–9871, Red Hook, NY, USA, 2018. Curran Associates Inc.
- Racula Necula, Mihaela Breaban, and Madalina Raschip. Tackling Dynamic Vehicle Routing Problem with Time Windows by means of ant colony system. In *2017 IEEE Congress on Evolutionary Computation (CEC)*, pages 2480–2487, 2017. doi: 10.1109/CEC.2017.7969606.
- Charles E. Noon and James C. Bean. An efficient transformation of the generalized traveling salesman problem. *INFOR: Information Systems and Operational Research*, 31(1):39–44, 1993. doi: 10.1080/03155986.1993.11732212.
- Larissa Tebaldi Oliveira, Everton Fernandes Silva, José Fernando Oliveira, and Franklina Maria Bragion Toledo. Integrating irregular strip packing and cutting path determination problems: A discrete exact approach. *Computers and Industrial Engineering*, 149:106757, 2020. doi: 10.1016/j.cie.2020.106757.
- Alena Otto, Niels Agatz, James Campbell, Bruce Golden, and Erwin Pesch. Optimization approaches for civil applications of unmanned aerial vehicles (uavs) or aerial drones: A survey. *Networks*, 72(4):411–458, 2018. doi: 10.1002/net.21818.
- Christos Papadimitriou. Euclidean TSP is NP-complete. *Theoret. Comput. Sci.*, 4:237–244, 1977.

- Panos Pardalos, Ding-Zhu Du, and Ronald Graham. *Handbook of Combinatorial Optimization*. Springer, 2013. ISBN 978-1-4419-7996-4.
- Wanxi Peng, Nyuk Ling Ma, Dangquan Zhang, Quan Zhou, Xiaochen Yue, Shing Ching Khoo, Han Yang, Ruirui Guan, Huiling Chen, Xiaofan Zhang, Yacheng Wang, Zihan Wei, Chaofan Suo, Yuhao Peng, Yafeng Yang, Su Shiung Lam, and Christian Sonne. A review of historical and recent locust outbreaks: Links to global warming, food security and mitigation strategies. *Environmental Research*, 191:110046, 2020. doi: 10.1016/j.envres.2020.110046.
- Artur Pessoa, Ruslan Sadykov, Eduardo Uchoa, and François Vanderbeck. A generic exact solver for vehicle routing and related problems. *Mathematical Programming*, 183:483–523, 09 2020. doi: 10.1007/s10107-020-01523-z.
- Artur Alves Pessoa, Ruslan Sadykov, and Eduardo Uchoa. Enhanced branch-cut-and-price algorithm for heterogeneous fleet vehicle routing problems. *European Journal of Operational Research*, 270(2):530–543, 2018. doi: 10.1016/j.ejor.2018.04.009.
- Alexander Petunin. General model of tool path problem for the cnc sheet cutting machines. *IFAC-PapersOnLine*, 52(13):2662–2667, 2019. doi: 10.1016/j.ifacol.2019.11.609.
- Alexander Petunin, Michael Khachay, Stanislav Ukolov, and Pavel Chentsov. Using pcgtsp algorithm for solving generalized segment continuous cutting problem. *IFAC-PapersOnLine*, 55(10):578–583, 2022. doi: doi.org/10.1016/j.ifacol.2022.09.456.
- Alexander A. Petunin and Chrysostomos Stylios. Optimization models of tool path problem for cnc sheet metal cutting machines. *IFAC-PapersOnLine*, 49(12):23–28, 2016. doi: 10.1016/j.ifacol.2016.07.544.
- Olcay Polat. A parallel variable neighborhood search for the vehicle routing problem with divisible deliveries and pickups. *Computers & Operations Research*, 85:71–86, 2017. doi: 10.1016/j.cor.2017.03.009.
- Petrică C. Pop, Ovidiu Cosma, Cosmin Sabo, and Corina Pop Sitar. A comprehensive survey on the generalized traveling salesman problem. *European Journal of Operational Research*, 2023. ISSN 0377-2217. doi: 10.1016/j.ejor.2023.07.022.
- Meng Qiu, Zhuo Fu, Richard Eglese, and Qiong Tang. A tabu search algorithm for the vehicle routing problem with discrete split deliveries and pickups. *Computers & Operations Research*, 100:102–116, 2018. doi: 10.1016/j.cor.2018.07.021.
- Jefferson A. Riera, Ricardo M. Lima, and Omar M. Knio. A review of hydrogen production and supply chain modeling and optimization. *International Journal of Hydrogen Energy*, 48(37):13731–13755, 2023. doi: 10.1016/j.ijhydene.2022.12.242.
- Roberto Roberti and Paolo Toth. Models and algorithms for the asymmetric traveling salesman problem: an experimental comparison. *EURO Journal on Transportation and Logistics*, 1(1): 113–133, 2012. ISSN 2192-4376. doi: 10.1007/s13676-012-0010-0.
- Ana Maria Rodrigues and José Soeiro Ferreira. Cutting path as a rural postman problem: solutions by memetic algorithms. *Int. J. Comb. Optim. Probl. Informatics*, 3(1):31–46, 2012.
- Diego Gabriel Rossit, Daniele Vigo, Fernando Tohmé, and Mariano Frutos. Visual attractiveness in routing problems: A review. *Computers and Operations Research*, 103:13–34, 2019. doi: 10.1016/j.cor.2018.10.012.

- Raad Salman, Johan S. Carlson, Fredrik Ekstedt, Domenico Spensieri, Johan Torstensson, and Rikard Söderberg. An industrially validated CMM inspection process with sequence constraints. *Procedia CIRP*, 44:138 – 143, 2016. doi: 10.1016/j.procir.2016.02.136.
- Raad Salman, Fredrik Ekstedt, and Peter Damaschke. Branch-and-bound for the Precedence Constrained Generalized Traveling Salesman Problem. *Operations Research Letters*, 48(2): 163–166, 2020. doi: 10.1016/j.orl.2020.01.009.
- Subhash C. Sarin, Hanif D. Sherali, and Ajay Bhootra. New tighter polynomial length formulations for the Asymmetric Traveling Salesman Problem with and without precedence constraints. *Oper. Res. Lett.*, 33(1):62–70, 2005. ISSN 0167-6377. doi: 10.1016/j.orl.2004.03.007.
- Anatoly I. Serdyukov. Some extremal bypasses in graphs (in russian). *Upravlyaemye Systemy = Controllable Systems*, (76–79), 1978.
- Zuojun Max Shen and Yiqi Sun. Strengthening supply chain resilience during covid-19: A case study of jd.com. *Journal of Operations Management*, 69(3):359–383, 2023. doi: 10.1002/joom.1161.
- Hanif D. Sherali and Patrick J. Driscoll. On tightening the relaxations of miller-tucker-zemlin formulations for asymmetric traveling salesman problems. *Operations Research*, 50(4):656–669, 2002. doi: 10.1287/opre.50.4.656.2865.
- S. Umar Sherif, Natarajan Jawahar, and M. Balamurali. Sequential optimization approach for nesting and cutting sequence in laser cutting. *Journal of Manufacturing Systems*, 33(4): 624–638, 2014. doi: 10.1016/j.jmsy.2014.05.011.
- Everton Fernandes Silva, Larissa Tebaldi Oliveira, José Fernando Oliveira, and Franklina Maria Bragion Toledo. Exact approaches for the cutting path determination problem. *Computers & Operations Research*, 112:104772, 2019. doi: 10.1016/j.cor.2019.104772.
- Michiel Smid. On some combinatorial problems in metricspaces of bounded doubling dimension, 2010. URL <https://people.scs.carleton.ca/~michiel/mst-ann-doubling.pdf>. Accessed Dec. 6, 2020.
- Stephen L. Smith and Frank Imeson. GLNS: An effective large neighborhood search heuristic for the Generalized Traveling Salesman Problem. *Computers & Operations Research*, 87: 1–19, 2017. doi: 10.1016/j.cor.2017.05.010.
- Ninja Soeffker, Marlin W. Ulmer, and Dirk C. Mattfeld. Stochastic dynamic vehicle routing in the light of prescriptive analytics: A review. *European Journal of Operational Research*, 298(3):801–820, 2022. doi: 10.1016/j.ejor.2021.07.014.
- Seema Srivastava, Santosh Kumar, R Garg, and P. Sen. Generalized Traveling Salesman Problem through n sets of nodes. *CORS journal*, 7(2):97–101, 1969.
- George Steiner. On the complexity of dynamic programming for sequencing problems with precedence constraints. *Annals of Oper. Res.*, 26(1):103–123, 1990.
- Yu Su-Ping and Mao Wei-Wei. An improved ant colony optimization for vrp with time windows. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 9: 327–334, 10 2016. doi: 10.14257/ijsp.2016.9.10.31.

- Ola Svensson, Jakub Tarnawski, and László A. Végh. A constant-factor approximation algorithm for the asymmetric traveling salesman problem. *J. ACM*, 67(6), 2020. ISSN 0004-5411. doi: 10.1145/3424306.
- Kunal Talwar. Bypassing the embedding: Algorithms for low dimensional metrics. In *Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing, STOC '04*, page 281–290, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 1581138520. doi: 10.1145/1007352.1007399.
- Paolo Toth and Daniele Vigo, editors. *The Vehicle Routing Problem*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2001. ISBN 0-89871-498-2.
- Paolo Toth and Daniele Vigo. *Vehicle Routing: Problems, Methods, and Applications, Second Edition*. MOS-Siam Series on Optimization. SIAM, 2 edition, 2014. ISBN 1611973589,978-1-611973-58-7.
- Vera Traub and Jens Vygen. An improved approximation algorithm for the asymmetric traveling salesman problem. *SIAM Journal on Computing*, 51(1):139–173, 2022. doi: 10.1137/20M1339313.
- Thibaut Vidal, Teodor Gabriel Crainic, Michel Gendreau, and Christian Prins. Heuristics for multi-attribute vehicle routing problems: A survey and synthesis. *European Journal of Operational Research*, 231(1):1–21, 2013a. doi: doi.org/10.1016/j.ejor.2013.02.053.
- Thibaut Vidal, Teodor Gabriel Crainic, Michel Gendreau, and Christian Prins. A hybrid genetic algorithm with adaptive diversity management for a large class of Vehicle Routing Problems with Time Windows. *Comput. Oper. Res.*, 40(1):475–489, 2013b. ISSN 0305-0548. doi: 10.1016/j.cor.2012.07.018.
- David P. Williamson and David B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, New York, NY, USA, 1st edition, 2011. ISBN 0521195276, 9780521195270.
- Gerhard Wäscher, Heike Haußner, and Holger Schumann. An improved typology of cutting and packing problems. *European Journal of Operational Research*, 183(3):1109–1130, 2007. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2005.12.047.
- Zhenci Xu, Yingjie Li, Sophia N. Chau, Thomas Dietz, Canbing Li, Luwen Wan, Jindong Zhang, Liwei Zhang, Yunkai Li, Min Gon Chung, and Jianguo Liu. Impacts of international trade on global sustainable development. *Nature Sustainability*, 3(11):964–971, 2020. doi: 10.1038/s41893-020-0572-z.
- Yaoqin Yu, Feng Lian, and Zhongzhen Yang. Pricing of parcel locker service in urban logistics by a tsp model of last-mile delivery. *Transport Policy*, 114:206–214, 2021. doi: doi.org/10.1016/j.tranpol.2021.10.002.
- Rong Yuan, João F.D. Rodrigues, Juan Wang, and Paul Behrens. The short-term impact of us-china trade war on global ghg emissions from the perspective of supply chain reallocation. *Environmental Impact Assessment Review*, 98:106980, 2023. doi: 10.1016/j.eiar.2022.106980.
- Yuan Yuan, Diego Cattaruzza, Maxime Ogier, and Frédéric Semet. A branch-and-cut algorithm for the generalized traveling salesman problem with time windows. *European Journal of Operational Research*, 286(3):849–866, 2020. ISSN 0377-2217. doi: 10.1016/j.ejor.2020.04.024.

-
- Tao Zhang. Global economic challenges and opportunities. *Business Economics*, 53(1):3–9, 2018.
- Yaojie Zhang, Jiaxin He, Mengxi He, and Shaofang Li. Geopolitical risk and stock market volatility: A global perspective. *Finance Research Letters*, 53:103620, 2023. doi: 10.1016/j.fr1.2022.103620.

