International Symposium
on Combinatorial Optimization

ISCO 2026

9th International Symposium on Combinatorial Optimization

BOOK OF EXTENDED ABSTRACTS

Kuşadası, Türkiye

May 6–8, 2026

Spring School: May 4–5, 2026

Korumar Ephesus Beach & Spa Resort

Kuşadası · Aydın · Türkiye

Organizing Institutions

Karabük University
Karabük, Türkiye

Aydın Adnan Menderes University
Aydın, Türkiye

Yaşar University
Izmir, Türkiye

Paris Dauphine University
Paris, France

University of Malta,
Malta

Kuwait University
Kuwait

Springer LNCS
Lecture Notes in Computer Science

ISCO 2026 Committees

Conference Chairs

Simge Küçükyavuz — *Northwestern University, USA*
A. Ridha Mahjoub — *Kuwait University, Kuwait; Paris Dauphine University, France*
Volker Kaibel — *Otto-von-Guericke Universität, Germany*
Hakan Kutucu — *Karabük University, Türkiye*

Organizing Committee

Korhan Günel — *Aydın Adnan Menderes University, Türkiye*
Hakan Kutucu — *Karabük University, Türkiye*
Refet Polat — *Yaşar University, Türkiye*
A. Ridha Mahjoub — *Kuwait University, Kuwait; Paris Dauphine University, France*
Enisse Kara — *Karabük University, Türkiye*
Onur Baysal — *University of Malta, Malta*
Mariem Ben Salem — *Esprit School of Engineering, Tunisia*

Steering Committee

Mourad Baïou	*LIMOS, CNRS, University Blaise Pascal, Clermont-Ferrand*	France
Pierre Fouilhoux	*LIPN, Université Sorbonne Paris Nord*	France
Luis Gouveia	*University of Lisbon*	Portugal
Jon Lee	*University of Michigan*	USA
Ivana Ljubic	*ESSEC Business School, Paris*	France
Nelson Maculan	*Universidad Federal do Rio de Janeiro*	Brasil
A. Ridha Mahjoub	*Kuwait University; Paris-Dauphine*	France
Giovanni Rinaldi	*IASI, CNR*	Italy

Program Committee

Beste Basciftci	*University of Iowa*	USA
Andreas Bley	*Universität Kassel*	Germany
Flavia Bonomo	*University of Buenos Aires*	Argentina
Francesco Carrabs	*Università di Salerno*	Italy
Raffaele Cerulli	*Università di Salerno*	Italy
Bo Chen	*Warwick University*	UK
Jean-François Cordeau	*HEC Montreal*	Canada
Ibrahima Diarrassouba	*Le-Havre University*	France
Tınaz Ekim	*Boğaziçi University*	Türkiye
Marcia Fampa	*Universidade Federal do Rio de Janeiro*	Brazil
Ricardo Fukasawa	*University of Waterloo*	Canada
Juan José Salazar González	*Universidad de La Laguna*	Spain
Oktay Günlük	*Georgia Tech*	USA
Oya Ekin Karaşan	*Bilkent University*	Türkiye
Markus Leitner	*Vrije Universiteit Amsterdam*	Netherlands
Irene Loiseau	*Universidad de Buenos Aires*	Argentina
Marco Lübbecke	*RWTH Aachen University*	Germany
Eduardo Moreno	*Universidad Adolfo Ibáñez*	Chile
Urfat Nuriyev	*Karabakh University*	Azerbaijan
Selin Özpeynirci	*Izmir University of Economics*	Türkiye
Nancy Perrot	*Orange Company*	France
Frits Spieksma	*Eindhoven University of Technology*	Netherlands
Eduardo Uchoa	*Universidade Federal Fluminense*	Brazil
Stefan Voss	*Universität Hamburg*	Germany
Hande Yaman	*KU Leuven*	Belgium

Preface

Dear Participants,

welcome to ISCO 2026, the 9th International Symposium on Combinatorial Optimization!

After the previous editions held in Hammamet, Athens, Lisbon, Vietri sul Mare, Marrakesh, Montreal (online), Paris (online), and Tenerife, we are happy to welcome you to Kuşadası, on the Aegean coast of Türkiye, a region that has been a meeting point of cultures, ideas, and travellers for thousands of years.

ISCO has become a highly anticipated biennial meeting for the combinatorial optimization research community. We are grateful to all authors who contributed to our high-level scientific program. Overall, we received about 120 submissions from researchers in many different countries. Among them, about 64 full papers were submitted for the LNCS post-conference proceedings book, and 41 of them were accepted. We would like to also thank all the members of the Program Committee, the Organizing Committee, and the external reviewers for their excellent work within demanding time constraints.

Our 28 contributed session span many different topics of combinatorial optimization. Together with four invited keynote speakers by internationally renowned researchers, Hande Yaman Paternotte, Tınaz Ekim, Sebastian Pokutta, and Miguel F. Anjos, and a dedicated Spring School on Packing and Covering by Gerard Cornuéjols and Ahmad Abdi, we are confident that the program will be a source of insights and fruitful discussions for all participants.

We hope you will all enjoy the conference and your stay in Kuşadası!

Simge Küçükyavuz
A. Ridha Mahjoub
Volker Kaibel
Hakan Kutucu
ISCO 2026 Co-Chairs

Plenary Lectures

Partitioning a Graph into Connected Components

Hande Yaman Paternotte — Faculty of Economics and Business, KU Leuven, Belgium

Network Security and Emergency Preparedness: The Defensive Domination Model

Tınaz Ekim — Industrial Engineering, Boğaziçi University, Türkiye

When Algorithms Learn: Discrete Optimization Meets Machine Learning

Sebastian Pokutta — Institute for Mathematics, Technische Universität Berlin, Germany

Discrete Optimization Models for Electric Vehicle Charging Operations

Miguel F. Anjos — Operational Research, University of Edinburgh, United Kingdom

Partitioning a Graph into Connected Components

Hande Yaman Paternotte — Faculty of Economics and Business, KU Leuven, Belgium

Partitioning the vertex set of a graph into connected components subject to side constraints — such as cardinality, capacity, or service-level requirements — is a fundamental combinatorial structure underlying many applications in districting, clustering, and network design. In this talk we review modelling approaches and exact solution methods for such problems, with a focus on polyhedral results, valid inequalities, and decomposition algorithms that exploit the structure of connected partitions. We also discuss recent advances and open challenges in this rich area of combinatorial optimization.

Network Security and Emergency Preparedness: The Defensive Domination Model

Tınaz Ekim — Industrial Engineering, Boğaziçi University, Türkiye

Defensive domination provides a graph-theoretic model for placing limited defensive resources on a network so that any attack of a given size can be neutralized by relocating defenders along edges. After motivating the model from network security and emergency preparedness applications, we present structural results, complexity analyses, and exact algorithmic frameworks for several variants of the problem, including k-defensive and secure domination. We conclude with computational results on real and synthetic instances and outline directions for future research.

When Algorithms Learn: Discrete Optimization Meets Machine Learning

Sebastian Pokutta — Institute for Mathematics, Technische Universität Berlin, Germany

The interaction between discrete optimization and machine learning has deepened significantly in recent years, with each field offering tools that reshape the other. In this talk, we explore how learning-based components can accelerate classical combinatorial algorithms, how optimization principles underpin many machine-learning pipelines, and how hybrid approaches achieve results that neither paradigm reaches alone. We illustrate these themes with concrete examples from large-scale integer programming, structured prediction, and learning-augmented algorithms.

Discrete Optimization Models for Electric Vehicle Charging Operations

Miguel F. Anjos — Operational Research, University of Edinburgh, United Kingdom

The transition to electric mobility raises a wide range of operational challenges, from siting public chargers to scheduling fleet charging under grid and demand uncertainty. This talk surveys discrete optimization models developed to address such problems, including mixed-integer formulations, robust and stochastic variants, and decomposition-based solution methods. We discuss case studies in fleet charging and infrastructure planning, and highlight the role of combinatorial optimization in supporting an efficient, reliable, and equitable electric-vehicle ecosystem.

Spring School

Packing and Covering

May 4–5, 2026 · Kuşadası, Türkiye

The ISCO 2026 conference is preceded by a Spring School on May 4 and 5, 2026, offering an excellent educational opportunity for graduate students and early-career researchers interested in combinatorial optimization.

Packing and covering problems lie at the heart of combinatorial optimization. Through six two-hour lectures and two two-hour exercise sessions, the school introduces objects sitting at the heart of many min-max theorems, totally unimodular and balanced matrices, perfect graphs, and ideal clutters together with recent advances in the area, including the first constant-factor relaxation of Woodall's conjecture, a polyhedral proof of Lovász's matching lattice theorem, and exponential lower bounds for cube-ideal set systems.

Lecturers

Gerard Cornuéjols — *Carnegie Mellon University, USA*
Ahmad Abdi — *London School of Economics, UK*

Sponsors

MAIN SPONSOR

hexaly

SPONSORS

probim

pargesoft

İstanbul | London | Brussels

Eduline

PUBLICATION PARTNER

Lecture Notes in Computer Science

LNCS LNAI LNBI

The Organizing Committee gratefully acknowledges the generous support of all sponsors and partners, whose contributions have made ISCO 2026 possible.

ISCO 2026 - Conference Program

MAY 6, WEDNESDAY

OPENING 08:45-09:00 | Smyrna Room

PLENARY	Tınaz Ekim 09:00-09:50	Room: Smyrna
	Network Security and Emergency Preparedness: The Defensive Domination Model	
W1 | 10:00 – 11:15	***Parallel Sessions***	
Session	**Room Smyrna — Scheduling I: Job & Machine Scheduling**	**Chairperson: Alain Quilliot**
#23	**Minimizing the Total Weighted Completion Time using Restarts**	
	*Aflatoun Amouzandeh, **Rob van Stee***	
#7	**Job Scheduling under No-Idleness Constraints**	
	***Alain Quilliot**, Hélène Toussaint, Philippe Chretienne*	
#10	**Minimizing the Weighted Makespan**	
	*Aflatoun Amouzandeh, Klaus Jansen, **Lis Pirotton**, Rob van Stee, Corinna Wambsganz*	
Session	**Room Troy — Graph Theory & Tree Algorithms**	**Chairperson: Mahsa Dalirrooyfard**
#12	**On the co-κ-plex polytopes of trees**	
	Alexandre Dupont-Bouillard	
#13	**A practical FPT algorithm for Treedepth**	
	*Ernst Althaus, **Jonas Blochwitz***	
#17	**Cosigning Crossing Families and Outer-Planar Gadgets**	
	*Ahmad Abdi, **Mahsa Dalirrooyfard**, Meike Neuwohner*	
Session	**Room Lydia — Optical Networks & Flows**	**Chairperson: Francesca Giancola**
#4	**On lower bounds for the minimum spectrum width for routing and spectrum assignment**	
	***Victoria Kaial**, Annegret K. Wagler*	
#97	**A Two-stage Chance-Constrained Programming Approach to Wavelength Dimensioning of WDM Networks**	
	*Haoyuan Xue, **Merve Bodur**, Maryam Daryalal*	
#135	**Network Congestion Modeling under Demand Uncertainty via Wardrop Equilibria**	
	*Yasmine Beck, **Francesca Giancola**, Ivana Ljubic, Sara Mattia*	

COFFEE BREAK 11:15-11:45

W2 | 11:45 – 13:00	***Parallel Sessions***	
Session	**Room Smyrna — Scheduling II: Advanced Models**	**Chairperson: Giovanni Righini**
#6	**Parallel Machine Job Scheduling under Tree-Delay Precedence Constraints**	
	***Alain Quilliot**, Djamal Rebaine*	
#40	**Efficient and fair single-machine scheduling**	
	Giovanni Righini	
#56	**Branch and Price for Job-Shop Scheduling with Time-Dependent Costs and Cardinality Resource Constraints**	
	***Marouane Felloussi**, Mohammed Ghannam, João Dionísio, Paolo Gianessi, Xavier Delorme*	
Session	**Room Troy — Polyhedral Combinatorics I**	**Chairperson: Gérard Cornuéjols**
#42	**On disjunction convex hulls for generalized cross polytopes**	
	***Yushan Qu**, Jon Lee*	
#53	**The Chvátal Rank of 2-Dimensional Integer-Free Polyhedra**	
	*Vrishabh Patil, **Gérard Cornuéjols***	
#64	**On the Virtual Network Embedding polytope**	
	***Alexis Schneider**, Pierre Fouilhoux, Lucas Létocart, Nancy Perrot, Amal Benhamiche*	
Session	**Room Lydia — Matching & Assignment**	**Chairperson: Kanstantsin Pashkovich**
#14	**Approximation Algorithms for the b-Matching and List-Restricted Variants of MaxQAP**	
	***Jiratchaphat Nanta**, Vorapong Suppakitpaisarn, Piyashat Sripratak*	
#58	**The Meta-rotation Poset for Student Project Allocation**	
	***Peace Ayegba**, Sofiat Olaosebikan*	
#65	**Sequential Linear Contracts on Matroids**	
	***Kanstantsin Pashkovich**, Jacob Skitsko, Yun Xing*	

LUNCH 13:00-14:30

W3 | 14:30 – 15:45	***Parallel Sessions***	
Session	**Room Smyrna — Algorithms & Computational Complexity**	**Chairperson:Mercedes Landete**
#16	**The falsification problem: how hard is it to falsify heuristics?**	
	***Stefano Huber**, Monaldo Mastrolilli, Eleonora Vercesi*	
#25	**Translocation Distance in Unary Encodings**	
	*Maria Constantin, **Adrian Miclaus**, Alexandru Popa*	
#72	**Exact Block-Decomposition Algorithms for the Generalized Vertex Cover Problem**	
	*Juan Franciso Correcher, **Mercedes Landete**, Juanjo Peiró, Hande Yaman*	

Session	Room Troy — Vehicle Routing & Green Logistics	Chairperson: Elham Jelodari Mamaghani
#36	**Time-Dependent Carbon-Aware Electric Vehicle Routing for Last-Mile Delivery**	
	Aysenur Bal, *Ecem Yavuzdogan, Milad Elyasi*	
#104	**AI-Guided Multi-Objective Optimization for Green Centralized Collaborative Logistics**	
	Elham Jelodari Mamaghani	
#143	**An Integrated ALNS–LSP Approach for the Pickup and Delivery Problem with Two Cross-Docks and 3D Loading Constraints**	
	Mehmet Asaf Düzen, *Vildan Özkır, Umman Mahir Yıldırım*	
Session	**Room Lydia — Graph Coloring & Combinatorial Structures**	**Chairperson: Martina Cerulli**
#122	**Modeling the Colorful Components Problems via Representative Formulations**	
	Carmine Sorgente, ***Martina Cerulli***, *Claudia Archetti, Diego Delle Donne*	
#124	**On the Polyhedral Structure of the Split-Interval Coloring Problem**	
	Diego Delle Donne, *Javier Marenco*	
#155	**The weak k-metric dimension of Kn × Kn**	
	Mohammad Farhan, *Dorota Kuziak, Ismael G. Yero*	

COFFEE BREAK 15:45-16:30

W4 | 16:15 – 17:30	*Parallel Sessions*	
Session	**Room Smyrna — Polyhedral Combinatorics II**	**Chairperson: Zacharie Ales**
#92	**The Integrality Gap of the Traveling Salesman Problem is 4/3 if the LP Solution Has at Most n+6 Non-Zero Components**	
	Tullio Villa, *Eleonora Vercesi, Janos Barta, Monaldo Mastrolilli*	
#103	**Valid Equalities and Dimension of the Optimal Classification Tree Problem**	
	Zacharie Ales	
#133	**Improving rank inequalities for covering and packing problems**	
	Leonardo de Abreu, *Manoel Campêlo*	
Session	**Room Troy — Heuristics & Approximation**	**Chairperson: Hande Yaman**
#19	**Reformulations and Heuristics for the Shortest Positive Path Problem in Signed Digraphs**	
	João Sombra, ***Rafael Andrade***	
#88	**Approximation schemes for covering and packing mixed-integer programs with a fixed number of constraints**	
	Kobe Grobben, ***Phablo F. S. Moura***, *Hande Yaman*	
#170	**A MILP approach to Regular MaxSAT**	
	Jordi Coll, Marcelo Finger, ***Felip Manyà***, *Sandro Preto, Elifnaz Yangin*	
Session	**Room Lydia — Production Scheduling & Lot-Sizing**	**Chairperson: Görkem Yılmaz**
#136	**A Branch-and-Price Approach for Lot-sizing and Scheduling Problem in Co-Production Systems**	
	Eyüp Ensar Işık, *Nabil Absi, Semra Ağralı*	
#158	**Time-Dependent Quality Degradation in Harvest-to-Mill Scheduling: A Mixed-Integer Programming Approach**	
	Haluk Elis, *Ozan Pembe, Ayhan Özgür Toy*	
#167	**Simultaneous Lot Sizing and Scheduling for a Production Planning Problem Including Shift and Overtime Decisions**	
	Öykü Özel, ***Gorkem Yilmaz***	

WELCOME RECEPTION 17:30-18:30

MAY 7, THURSDAY

PLENARY	Miguel F. Anjos 09:00-09:50	Room: Smyrna
	Discrete Optimization Models for Electric Vehicle Charging Operations	
Th1 | 10:00 – 11:15	*Parallel Sessions*	
Session	**Room Smyrna — UAVs & Post-Disaster Operations**	**Chairperson: Selin Özpeynirci**
#57	**Use of Unmanned Aerial Vehicles for Post-Disaster Response**	
	Mert Kara, Selin Özpeynirci, ***Arya Sevgen Misiç***	
#129	**Optimized Deployment of UAV-Base Stations in Post-Earthquake Scenarios**	
	Safiye Aybala Kılıç, *Betül Çoban Yelken, Eda Yücel*	
#160	**Optimization of Shelter Placement in Post-Disaster Management**	
	Mehmet Melikşah Çalışkan, ***Cemal Bolat***, *Didem Gözüpek*	
Session	**Room Troy — Domination & Security Networks**	**Chairperson: Tınaz Ekim**
#94	**A Benders Decomposition Framework for the k-Defensive Domination Problem**	
	Bilge Varol, *Tınaz Ekim, Kübra Tanınmış*	
#98	**Exact and Heuristic Methods for Variants of the Secure Domination Problem**	
	Burcu Çeştan, *Tınaz Ekim*	
#114	**Minimizing Response Time in District-Based Security Networks via Integrated Policy Comparison**	
	Fatih Mehmet Yilmaz, *Tınaz Ekim*	

Session	Room Lydia — Supply Chain & Logistics Optimization	Chairperson: Bilge Bilgen
#132	**Neural Approximate Dynamic Programming for the Ultra-fast Order Dispatching**	
	Mucahit Cevik, *Merve Bodur, Arash Dehghan*	
#96	**A Flexible Weight Approach for the Logistics Performance Index Methodology**	
	Işıl Can, ***Özgür Özpeynirci***, *Sinem Tokcaer*	
#105	**Olive oil supply chain planning under uncertainty**	
	Bilge Bilgen	
Session	**Room Pergamon — Real-World Applications of Discrete Optimization**	**Chairperson: İpek Nur Yiyit Yılmaz**
#35	**Curriculum prioritization in disruptive educational contexts: A solution approach based on the Knapsack Problem with Scheduled Items**	
	Valentina Ugalde-Calderón, *Jessica Rodríguez-Pereira, Sebastian Dávila-Gálvez, Óscar C. Vásquez*	
#93	**A Feature Engineering Based Market Prediction for the Traveling Purchaser Problem**	
	Aykut Sen, *Ilker Kucukoglu*	
#154	**Scenario-Based Stochastic Last-Mile Pharmaceutical Delivery: MILP Formulation**	
	İpek Nur Yiyit Yılmaz, *Muhammed Emre Keskin*	

COFFEE BREAK 11:15-11:45

PLENARY	Hande Yaman Paternotte 11:45-12:35	Room: Smyrna
	Partitioning a Graph into Connected Components	

LUNCH 12:40-14:00

Th2 | 14:00 – 14:50	*Parallel Sessions*	
Session	**Room Smyrna — Optimization under Uncertainty**	**Chairperson: Frédéric Meunier**
#91	**Geoffrion's theorem beyond finiteness and rationality**	
	Santanu S. Dey, ***Frédéric Meunier***, *Diego A. Morán R.*	
#99	**Interval Optimization under Bounded Capacity Constraints**	
	Arthur Mazeyrat, *Ammar Oulamara, Tifenn Rault, Ameur Soukhal*	
Session	**Room Troy — Finance & Sustainable Investment**	**Chairperson: Emre Sefer**
#137	**The Effect of Carbon Emission on Production and Investment Planning**	
	İrem Sultan Ünver, *Çağrı Koç, Eda Yücel, Bahar Yetiş Kara*	
#152	**Robust Portfolio Optimization with Return Prediction**	
	Burak Kara, ***Emre Sefer***, *İhsan Yanıkoğlu*	
Session	**Room Lydia**	
	HEXALY MATHEMATICAL OPTIMIZATION PLATFORM	

Th3 | 15:00 – 15:50	*Parallel Sessions*	
Session	**Room Smyrna — Districting & Facility Networks**	**Chairperson: Alaittin Kırtışoğlu**
#26	**A Heuristic Approach for Balanced, Compact and Contiguous Districting**	
	Firdawss Bouamlat, *Cléa Martinez, Pr. Safa Bhar Layeb*	
#175	**FalCom: An MCMC Sampling Framework for Districting and Hierarchical Capacitated Facility Location Problems**	
	Hemanshu Kaul, ***Alaittin Kırtışoğlu***	
Session	**Room Troy —Fairness in Combinatorial Optimization**	**Chairperson: Kübra Tanınmış**
#134	**An Exact Method for Fair Influence Maximization**	
	Kübra Tanınmış, *Meryem Şırnak*	
#173	**Polyhedral Analysis of an Envy-Free Assignment Problem**	
	Pierre Fouilhoux, Eric Gourdin, Roland Grappe, ***Jules Nicolas-Thouvenin***, *Emilie Sirvent-Hien*	
Session	**Room Lydia — Drones: Data Collection & Routing**	**Chairperson: Fatiha Bendali**
#159	**A Drone-Based Multi-Warehouse Vehicle Routing Model for Post-Disaster Emergency Supply Distribution**	
	Onat Düzgün, *Selin Özpeynirci, Arya Sevgen Misiç, Ozan Baran Demirçivi*	
#164	**A mixed-integer linear programming model for data collection with a drone avoiding obstacles**	
	Yuankang Hu, *Fatiha Bendali, Jean Mailfert, Christophe Cariou, Laure Moiroux-Arvis*	
Session	**Room Pergamon — Pharmaceutical Logistics**	**Chairperson: İhsan Sadati**
#95	**A Pre-processing Approach for Integrated 3D Bin Packing Problem in Pharmaceuticals**	
	Seray Çakırgil, *Elif Atar, Eda Yücel*	
#120	**Specimen Collection Problem with Multiple Facilities and Batch Deliveries**	
	İhsan Sadati, *Halenur Şahin, Sibel Salman, Eda Yücel*	

Ephesus Tour and Temple of Artemis 16:00-19:30

GALA DINNER 20:00

MAY 8, FRIDAY		
PLENARY	**Sebastian Pokutta 09:00-09:50**	**Room: Smyrna**
	When Algorithms Learn: Discrete Optimization Meets Machine Learning	
F1 | 10:00 – 11:15	***Parallel Sessions***	
Session	**Room Smyrna — Spanning Trees & Graph Dynamics**	**Chairperson: Massinissa Merabet**
#30	**Cycle Basis-Driven ILP Bounds for the Minimum Branch Vertices Spanning Tree Problem**	
	Massinissa Merabet, *Matthieu Desprez*	
#108	**A Simple Backtracking Algorithm for Solving the Minimum Spanning Tree Problem with Disjunctive Conflict Constraints**	
	Murat Umut İzer, *Temel Öncan, İ. Kuban Altınel*	
#139	**Effects of Density Changes on a Novel Graph Burning Algorithm**	
	Arianne Nantel, *Arda Utku, Erin Meger*	
Session	**Room Troy — Transport & Healthcare Scheduling**	**Chairperson: Sven Mallach**
#39	**Electric Bus Scheduling and Charging Problem with Parking Constraints: Model and Heuristic**	
	Amira Bendjebla, *Ammar Oulamara, Lhassane Idoumghar, Michel Basset*	
#43	**Integrating crew scheduling and crew rostering for rail freight with train delays**	
	Héloïse Gachet, *Frédéric Meunier, Juliette Pouzet*	
#51	**A new modeling approach to elective surgery scheduling**	
	Sven Mallach	
Session	**Room Lydia — Network Analytics**	**Chairperson: Carmine Sorgente**
#125	**Anonymizing networks via mathematical programming**	
	Raffaele Cerulli, Ivana Ljubić, ***Carmine Sorgente***	
#22	**Minmax sink location problem on path networks with capacitated sinks**	
	Jannatul Maowa, *Robert Benkoczi*	
#138	**Robustness of Centrality-Based Influence Maximization under Structural Noise**	
	Melike Karatay, *Yasemin Demirel, Fidan Nuriyeva, Onur Uğurlu*	

COFFEE BREAK 11:15-11:45		
F2 | 11:45 – 13:00	***Parallel Sessions***	
Session	**Room Smyrna — MIP, Decomposition & Structured Optimization**	**Chairperson: Juanjo Peiró**
#70	**Solving convex quadratic optimization with indicators over structured graphs**	
	Aaresh Bhathena, ***Salar Fattahi***, *Andrés Gómez, Simge Küçükyavuz*	
#90	**Some formulations for the capacitated dispersion problem**	
	Juanjo Peiró, *Mercedes Landete, Hande Yaman*	
#163	**Subproblem Solving in Benders Decomposition for Affine Potential-Based Flow Problems with Topology Switching and Robustness Scenarios**	
	Tim Donkiewicz, ***Oliver Gaul***	
Session	**Room Troy — Assignment & Preference-Based Optimization**	**Chairperson: Malek Mouhoub**
#61	**GPU-Accelerated Harris Hawks Optimization with Robust Tabu Search for Solving Quadratic Assignment Problems**	
	Cem Serimer, *Deniz Cantürk*	
#156	**Preference-Based Combinatorial Optimization**	
	Malek Mouhoub	
#63	**Expected Cost of Greedy Online Facility Assignment on Regular Polygons**	
	Rawha Siddiqi Riad, Tanzeem Rahat, ***Manzurul Hasan***	
Session	**Room Lydia — AI, LLMs & Machine Learning in OR**	**Chairperson: Sebastian Pokutta**
#32	**Global Optimization for Combinatorial Geometry Problems Revisited in the Era of LLMs**	
	Timo Berthold, ***Dominik Kamp***, *Gioni Mexi, Sebastian Pokutta, Imre Polik*	
#37	**Automatic Value Optimization and Semantic Option Matching in Sports Betting Markets using Vector-Based Logical Containment Framework**	
	Osman Dogukan Kefeli, *Dogugun Özkaya, Serdar Eric, Gokhan Ince*	
#171	**Large Language Models in Job Shop Scheduling: A Structured Analysis of Current Approaches**	
	Hasan Demir, *Deniz DEMİRKAN, Kürşat Mustafa Karaoğlan, Hakan Kutucu*	

CLOSING 13:00-13:20 | Smyrna Room
LUNCH 13:20

Content

Content

Content

Solving Convex Quadratic Optimization with Indicators Over Structured Graphs

Aaresh Bhathena[1], Salar Fattahi[1], Andrés Gómez[2], and Simge Küçükyavuz[3]

[1] University of Michigan, Ann Arbor, MI, USA
aareshfb@umich.edu, fattahi@umich.edu
[2] University of Southern California, Los Angeles, CA, USA, gomezand@usc.edu
[3] Northwestern University, Evanston, IL, USA, simge@northwestern.edu

Abstract. We propose an exact parametric dynamic programming algorithm for convex quadratic problems with indicator variables that exploits graph structure to achieve near-linear complexity under mild conditions.

We consider the following mixed-integer quadratic program (MIQP), defined by a symmetric and positive definite matrix $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$ and vectors $\boldsymbol{\lambda}, \boldsymbol{c} \in \mathbb{R}^n$:

$$\min_{\boldsymbol{x}\in\mathbb{R}^n, \boldsymbol{z}\in\{0,1\}^n} \quad \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c}^\top \boldsymbol{x} + \boldsymbol{\lambda}^\top \boldsymbol{z} \tag{1a}$$

$$\text{s.t.} \quad \boldsymbol{x}_i(1-\boldsymbol{z}_i) = 0 \qquad i = 1, 2, \ldots, n. \tag{1b}$$

In this problem, the binary vector $\boldsymbol{z} \in \{0,1\}^n$ encodes the support of the continuous vector $\boldsymbol{x} \in \mathbb{R}^n$. Specifically the constraint $\boldsymbol{x}_i(1-\boldsymbol{z}_i) = 0$ enforces that $\boldsymbol{x}_i = 0$ whenever $\boldsymbol{z}_i = 0$, and $\boldsymbol{z}_i = 1$ allows $\boldsymbol{x}_i \in \mathbb{R}$ to be unconstrained. The vector $\boldsymbol{\lambda} \in \mathbb{R}^n$ acts as the component-wise regularization parameter that promotes sparsity in $\boldsymbol{x}$. We assume throughout that $\boldsymbol{\lambda}_i > 0$ for every $i = 1, \ldots, n$ as $\boldsymbol{\lambda}_i \leq 0$ implies that $z_i = 1$ at optimality.

This work focuses on instances of Problem (1) in which the sparsity pattern of the Hessian matrix $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$ coincides with the adjacency matrix of a graph—hereafter referred to as the *support graph*—with certain sparsity structures. Specifically, we consider support graphs characterized by a *bounded treewidth* and a *polynomial volume growth* property. The former captures the extent to which the graph resembles a tree, while the latter imposes an upper bound on the number of nodes contained within any fixed graph distance from a given node. While treewidth is a classical concept in graph theory, polynomial volume growth is a less common but equally meaningful structural property. Both notions are defined below.

Definition 1 (Tree decomposition and treewidth). *The tree decomposition of a graph $G = (\mathcal{V}_G, \mathcal{E}_G)$ is a pair $(T, \mathcal{B})$, where $T = (\mathcal{V}_T, \mathcal{E}_T)$ is a tree and $\mathcal{B} = \{\mathcal{B}_u : u \in \mathcal{V}_T\}$ is a family of subsets (bags) of $\mathcal{V}_G$, satisfying the following conditions:*

1. *$\mathcal{V}_G = \bigcup_{u\in\mathcal{V}_T} \mathcal{B}_u$. That is, every node of G appears in at least one bag.*
2. *For every edge $(i, j) \in \mathcal{E}_G$, there exists a bag $\mathcal{B}_u$ containing both i and j.*
3. *For every node $i \in \mathcal{V}_G$, the set $\{u \in \mathcal{V}_T : i \in \mathcal{B}_u\}$ induces a connected subtree of T. That is, bags containing i form a connected subtree of T.*

The width of a tree decomposition is given by $\max_{u \in \mathcal{V}_T} \{|\mathcal{B}_u| - 1\}$. *The* treewidth*, denoted by* ω*, is defined as the minimum width among all tree decompositions of* G

Intuitively, treewidth measures how close a graph is to being a tree. For example, trees have treewidth 1. Series-parallel graphs have treewidth 2. On the other hand, fully dense graphs—being far from tree-like—have a maximum treewidth $|\mathcal{V}_\mathsf{G}| - 1$.

Graphs of bounded treewidth, introduced by [3] and popularized by [4], play a central role in algorithmic graph theory, as they characterize classes of problems that admit efficient dynamic programming algorithms. Many otherwise intractable combinatorial problems become polynomial-time solvable when restricted to such graphs.

Problem (1), however, involves a mixed-integer structure with both binary and continuous variables. Tree decompositions have also been studied in this broader optimization setting; in particular, [2] consider related quadratic problems with indicator variables and show that, when $\boldsymbol{Q}$ has bounded treewidth, an FPTAS is possible. In contrast to our work, their approach provides only approximate guarantees and, to our knowledge, lacks a practical implementation.

Definition 2 (Polynomial volume growth). *Given a graph* $\mathsf{G} = (\mathcal{V}_\mathsf{G}, \mathcal{E}_\mathsf{G})$ *and an integer* $m \geq 1$*, define* $\Delta_m := \max_{v \in \mathcal{V}_\mathsf{G}} \big|\{u \in \mathcal{V}_\mathsf{G} : \mathrm{dist}_\mathsf{G}(u, v) \leq m\}\big|$*, where* $\mathrm{dist}_\mathsf{G}(u, v)$ *denotes the graph distance between nodes* u *and* v *in* G*. We say that* G *has* polynomial volume growth *if there exist constants* $\gamma, \delta > 0$ *such that* $\Delta_m \leq \delta\, m^\gamma$ *for all* $m \geq 1$.

Definition 2 holds for many well-studied graph families. Examples include trees with $O(1)$ leaves, cycles, grid graphs, and support graphs of banded matrices. In contrast, graphs whose maximum degree scales with n, as well as trees with $O(n)$ leaves, such as binary trees and star graphs, do not satisfy this property.

Contributions

At the core of our proposed method lies a *pruning* technique that eliminates suboptimal choices of the binary vector $\boldsymbol{z} \in \{0, 1\}^n$, thereby reducing the search space. Consider the following equivalent formulation of Problem (1):

$$\begin{aligned} &\min_{\boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{z} \in \{0,1\}^n, \boldsymbol{x} \circ (1 - \boldsymbol{z}) = 0} \left\{ \frac{1}{2} \boldsymbol{x}^\top \boldsymbol{Q} \boldsymbol{x} + \boldsymbol{c}^\top \boldsymbol{x} + \boldsymbol{\lambda}^\top \boldsymbol{z} \right\} \\ &= \min_{\boldsymbol{x} \in \mathbb{R}^n} \left\{ \min_{\boldsymbol{z} \in \{0,1\}^n} \left\{ \frac{1}{2} \boldsymbol{x}^\top \left(\boldsymbol{Q} \circ \boldsymbol{z}\boldsymbol{z}^\top \right) \boldsymbol{x} + (\boldsymbol{c} \circ \boldsymbol{z})^\top \boldsymbol{x} + \boldsymbol{\lambda}^\top \boldsymbol{z} \right\} \right\}, \end{aligned}$$

where $\circ$ denotes the entry-wise product. Upon defining a convex quadratic function $p_{\boldsymbol{z}}(\boldsymbol{x}) := \frac{1}{2} \boldsymbol{x}^\top \left(\boldsymbol{Q} \circ \boldsymbol{z}\boldsymbol{z}^\top \right) \boldsymbol{x} + (\boldsymbol{c} \circ \boldsymbol{z})^\top \boldsymbol{x} + \boldsymbol{\lambda}^\top \boldsymbol{z}$ for every fixed $\boldsymbol{z} \in \{0, 1\}^n$, the above problem reduces to the following two-stage optimization problem:

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} f(\boldsymbol{x}), \quad \text{where} \quad f(\boldsymbol{x}) = \min_{\boldsymbol{z} \in \{0,1\}^n} \; p_{\boldsymbol{z}}(\boldsymbol{x}). \tag{2}$$

The formulation (1) suggests solving the problem by first characterizing the *parametric cost* f obtained by eliminating the binary variables $\boldsymbol{z}$, and then minimizing f over $\boldsymbol{x}$.

The main challenge lies in the first stage, since f is a piecewise function with up to 2^n convex quadratic pieces. Evidently, this offers no advantage over enumerating all z.

We show that this complexity can be drastically reduced by exploiting the structure of the Hessian matrix $\boldsymbol{Q}$. First, since $\boldsymbol{Q}$ is positive definite, any optimizer $\boldsymbol{x}^\star$ satisfies $\|\boldsymbol{x}^\star\|_\infty \le U$ for some constant U, and thus it suffices to characterize f over the bounded domain $\mathcal{D} = \{\boldsymbol{x} : \|\boldsymbol{x}\|_\infty \le U\}$. Second, we show that when two sparsity patterns z_1, z_2 have sufficiently large overlap, the difference $p_{z_1} - p_{z_2}$ has all its roots outside $\mathcal{D}$, implying that one quadratic piece uniformly dominates the other on $\mathcal{D}$ and can therefore be safely pruned. Leveraging these insights, we prove that under bounded treewidth and polynomial volume growth assumptions, the number of quadratic pieces that remain after pruning scales only linearly with n (but possibly exponentially with the treewidth).

Theorem 1 (Informal). *Suppose that the support graph of $\boldsymbol{Q}$ has bounded treewidth ω and polynomial volume growth. Under additional technical conditions, after pruning, the total number of quadratic pieces can be reduced to $O\left(n \cdot 2^{\mathrm{polylog}(\omega)}\right)$.*

While the existence of an efficient representation of the parametric cost f is encouraging, it does not by itself guarantee an efficient procedure for constructing it. To this end, we develop an efficient algorithm, called the *parametric algorithm*, for characterizing f in settings where the matrix $\boldsymbol{Q}$ admits a tree decomposition of small width. Our proposed parametric algorithm constructs f efficiently by dynamic programming (DP) operating over the tree decomposition of $\boldsymbol{Q}$.

Theorem 2 (Informal). *Suppose that the support graph of $\boldsymbol{Q}$ has bounded treewidth ω and polynomial volume growth. Under additional technical conditions, there exists an algorithm that solves* (1) *in $O\left(n\omega^2 \cdot 4^{\mathrm{polylog}(\omega)}\right)$ time and $O\left(n\omega^2 \cdot 2^{\mathrm{polylog}(\omega)}\right)$ memory.*

Application: Outlier detection in time-series

Accurate forecasting and outlier detection are often treated separately, yet classical forecasting methods are not robust to anomalous observations. We adopt a unified view and focus on extending *Simple Exponential Smoothing* (SES), a standard method for time-series forecasting. Given observations $\{\boldsymbol{y}_t\}_{t=1}^T$, SES constructs a smoothed signal $\{\boldsymbol{x}_t\}_{t=1}^T$ via $\boldsymbol{x}_t = \beta \boldsymbol{y}_t + (1-\beta)\boldsymbol{x}_{t-1}, t = 2, \ldots, T$, with $\boldsymbol{x}_1 = \boldsymbol{y}_1$ and $\beta \in (0,1)$. While SES is simple and effective, it exhibits a fundamental trade-off: large β yields fast adaptation but high sensitivity to outliers, whereas small β suppresses outliers at the cost of significant lag. Hence, SES cannot simultaneously achieve robustness and responsiveness.

To overcome this limitation, we propose *Exponential Smoothing with Outlier Correction* (ESOC), which explicitly models anomalies. We introduce an outlier vector $\boldsymbol{o} \in \mathbb{R}^T$ and binary variables $z \in \{0,1\}^T$ indicating corrupted samples, and consider a relaxed formulation that penalizes deviations from the SES dynamics:

$$\min_{\boldsymbol{x},\boldsymbol{o}\in\mathbb{R}^T} \sum_{t=1}^{T} (\boldsymbol{y}_t - \boldsymbol{x}_t - \boldsymbol{o}_t)^2 + \sum_{t=1}^{T} \boldsymbol{\lambda}_t z_t + \mu_1 \sum_{t=2}^{T} (\beta(\boldsymbol{y}_t - \boldsymbol{o}_t) + (1-\beta)\boldsymbol{x}_{t-1} - \boldsymbol{x}_t)^2 + \mu_2 \|\boldsymbol{o}\|_2^2 \tag{ESOC}$$

$$\text{s.t.} \quad \boldsymbol{o}_t(1 - z_t) = 0 \qquad \text{for } t = 1, \ldots, T,$$

When $z_t = 0$, this reduces to standard SES; when $z_t = 1$, o_t absorbs the corruption in y_t. This formulation preserves the simplicity and interpretability of exponential smoothing while enabling robust and principled outlier detection. Structurally, it leads to a mixed-integer quadratic program whose Hessian has treewidth equal to two and linear volume growth, making it an ideal application for the graph-based dynamic programming algorithms proposed in this work.

We evaluate the scalability of our algorithm on (ESOC) using five real-world time series from the Numenta Anomaly Benchmark [1], including AWS CloudWatch metrics and traffic speed data, with $\beta \in \{0.05, 0.2, 0.5\}$. For each series, we vary the problem size T by truncation and solve the resulting instances using both Gurobi and our algorithm. In total, 15 instances are solved for each T (five signals and three β values). As shown in Figure 1, our method solves all instances to optimality with an average runtime of 54 seconds at $T = 1000$, while Gurobi scales poorly and exceeds the one-hour limit for $T \geq 500$.

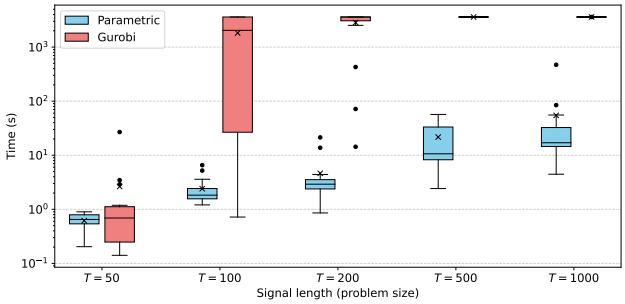

Fig. 1. Runtime comparison between the proposed algorithm and Gurobi on five real-world time series from the NAB dataset, with smoothing parameters. Signal lengths are varied by truncation. For each signal, smoothing parameters are set to $\beta \in \{0.05, 0.2, 0.5\}$. "×" denotes the mean runtime. Gurobi runs are terminated after one hour, resulting in flat values at larger signal lengths.

References

1. Ahmad, S., Lavin, A.: The numenta anomaly benchmark. https://github.com/numenta/NAB (2015)
2. Bienstock, D., Chen, T.: Solving convex QPs with structured sparsity under indicator conditions. arXiv preprint arXiv:2411.11722 (2024)
3. Halin, R.: S-functions for graphs. Journal of Geometry **8**(1-2), 171–186 (1976)
4. Robertson, N., Seymour, P.D.: Graph minors. II. Algorithmic aspects of tree-width. Journal of algorithms **7**(3), 309–322 (1986)

Exact Block-Decomposition Algorithms for the Generalized Vertex Cover Problem

Juan Francisco Correcher[1][0000-0002-0597-1741], Mercedes Landete[2][0000-0002-5201-0476], Juanjo Peiró[1][0000-0002-2944-599X], and Hande Yaman[3][0000-0002-3392-1127]

[1] Departament d'Estadística i Investigació Operativa, Facultat de Ciències Matemàtiques, Universitat de València, Spain
[2] Departamento de Estadística, Matemáticas e Informática, Instituto Centro de Investigación Operativa, Universidad Miguel Hernández de Elche, Spain
[3] Research Centre for Operations Research and Business Statistics (ORSTAT), Faculty of Economics and Business, KU Leuven, Belgium

Abstract. The Generalized Vertex Cover (GVC) problem extends the classical Vertex Cover by including penalties for uncovered edges, and is known to be $\mathcal{NP}$-hard. Given a graph with vertex costs and edge penalties, the goal is to select a subset of vertices minimizing the sum of selected vertex costs plus penalties.
We present an exact block-decomposition algorithm that exploits graph structures, specifically articulation points and block-tree decomposition, to reduce problem size while preserving optimality. By identifying leaf blocks and applying variable fixation rules based on cost bounds, the method decomposes the graph into smaller subproblems, which are solved iteratively.
Computational experiments on benchmark instances show that our approach significantly reduces the size of the optimization problem and accelerates solution times compared to standard integer linear programming formulations. Structured graphs with multiple articulation points benefit the most, highlighting the advantages of structural decomposition in exact algorithms for the GVC problem.

Keywords: Generalized Vertex Cover · Combinatorial Optimization · Graph Decomposition · Exact Algorithms · Block-Tree Structures

1 Introduction

Let $G = (V, E)$ be a graph. Let $c : V \to Q$ and $p : E \to Q_{>0}$ denote the objective-function coefficients associated with vertices and edges, respectively. The problem GVC(G) can be formulated as a standard quadratic unconstrained binary optimization (QUBO) problem ([2]).

A QUBO formulation for GVC(G) uses a binary decision variable x_i for each vertex $i \in V$, where $x_i = 1$ indicates that vertex i is selected in the cover, and

$x_i = 0$ otherwise. The model can be written as

$$\begin{aligned} &\min \sum_{i \in V} c_i x_i + \sum_{\{i,j\} \in E} p_{ij}(1 - x_i)(1 - x_j), \\ &\text{s.t.: } x_i \in \{0, 1\}, \quad i \in V. \end{aligned}$$

A classical linearization of this formulation introduces additional variables y_{ij}, one for each edge $\{i, j\} \in E$. The variable y_{ij} takes value 1 if at least one of its incident vertices belongs to the cover, and 0 otherwise. Using these variables, the problem can be expressed as the following linear mixed-integer formulation:

$$\min \sum_{i \in V} c_i x_i + \sum_{\{i,j\} \in E} p_{ij}(1 - y_{ij}), \tag{1}$$

$$\text{s.t.: } y_{ij} \leq x_i + x_j, \qquad \{i, j\} \in E, \tag{2}$$

$$x_i \leq y_{ij}, \quad x_j \leq y_{ij}, \qquad \{i, j\} \in E, \tag{3}$$

$$x_i \in \{0, 1\}, \qquad i \in V, \tag{4}$$

$$y_{ij} \in \{0, 1\}, \qquad \{i, j\} \in E. \tag{5}$$

Moreover, constraints (3) can be omitted, and the integrality constraints (5) can be relaxed to $0 \leq y_{ij} \leq 1$ for all $\{i, j\} \in E$, without affecting the optimal value of the formulation.

2 Methodology

A vertex $i \in V$ is an *articulation* (or *cut*) vertex of G if $G - \{i\}$ is disconnected. A *block* of G is a maximal 2-connected subgraph, and the *block–cut tree* $\mathcal{T}(G)$ is the tree whose nodes correspond to the blocks and articulation vertices of G, with an edge joining a block to an articulation vertex whenever the latter belongs to the former. A block that is a leaf of $\mathcal{T}(G)$ is called a *leaf block*.

The procedure starts from an empty solution x^*. For a given articulation vertex i, we analyze its incident leaf blocks to determine whether the value of x_i can be fixed. Each leaf block is solved independently, first testing whether x_i can be fixed to one and, if not, whether it can be fixed to zero. If either case applies, the corresponding variable values are fixed and the graph is reduced by removing the affected vertices and edges. Otherwise, the solutions for both scenarios are stored, all leaf blocks except vertex i are removed, the cost vector is updated accordingly, and the articulation vertex remains undecided. Whenever the value of x_i is eventually fixed, the solutions associated with its leaf blocks are recovered.

As each reduction of G may create new leaf blocks, the procedure is repeated until no articulation vertex remains unexplored.

3 Results

Regarding the experimental setup, we generated a collection of random graph instances with $n \in \{100, 200, 300\}$ and edge densities in $\{0.05, 0.10, 0.15, 0.20, 0.50, 1.00\}$.

Vertex costs c were drawn independently from a uniform distribution $U[1, 100]$, while edge penalties p were sampled from the intervals $\{[1, 10], [1, 25], [1, 50], [1, 100]\}$. Following the approach of [1], we applied a linking procedure to generate sets of instances with articulation points. Overall, this procedure resulted in a total of 420 instances.

Across all instances, approximately 10–11% of the edges remain uncovered in the optimal solutions, indicating that paying penalties is often more cost-efficient than enforcing full coverage. On average, 59% of the articulation vertices were fixed to one by the proposed reduction, while 28% were fixed to zero, leaving only 12% of the articulation vertices undecided. Furthermore, the articulation-based reduction and the decomposition algorithm based on the block–cut tree solve more than 82% of the instances faster than solving the linear model on the original, non-decomposed graph, achieving an average speedup of over 43% in those cases.

References

1. Landete, M., Marín, A., Sainz-Pardo, J.L.: Decomposition methods based on articulation vertices for degree-dependent spanning tree problems. *Computational Optimization and Applications* **68**, 749–773 (2017)
2. Punnen, A.P. (ed.): The Quadratic Unconstrained Binary Optimization Problem. Springer, Cham (2022)

Approximation schemes for covering and packing mixed-integer programs with a fixed number of constraints*

Kobe Grobben[0009−0008−1784−576X], Phablo F. S. Moura[0000−0002−8176−0874], and Hande Yaman[0000−0002−3392−1127]

Research Center for Operations Research & Statistics, KU Leuven, Belgium
{kobe.grobben,phablo.moura,hande.yaman}@kuleuven.be

1 Introduction

This paper presents an algorithmic study of mixed-integer programming problems, denoted by P, which are defined as:

$$\min \sum_{i\in[n]}\sum_{j\in[m]} v_{ij}x_{ij} + \sum_{i\in[n]} f_i y_i \tag{1}$$

$$\text{s.t.} \sum_{i\in[n]} x_{ij} \geq d_j \qquad \forall j \in [m], \tag{2}$$

$$\ell_{ij} y_i \leq x_{ij} \leq c_{ij} y_i \qquad \forall i \in [n],\ j \in [m], \tag{3}$$

$$y_i \in \{0,1\} \qquad \forall i \in [n]. \tag{4}$$

where m and n are positive integers, $v, \ell, c \in \mathbb{Z}^{nm}_{\geq}$, $d \in \mathbb{Z}^{m}_{\geq}$ and $f \in \mathbb{Z}^{n}_{\geq}$ denote the nonnegative integer parameters and $[a] = \{1, \ldots, a\}$ for a positive integer a. Throughout this paper, it is assumed, without loss of generality, that $c_{ij} \geq \ell_{ij}$ for all $i \in [n]$ and $j \in [m]$. For completeness, we also consider the packing variant of P, which is defined analogously.

The covering integer programs (CIP) are closely related to P. Given a matrix $A \in \mathbb{R}^{m\times n}_{\geq}$, vectors $a \in \mathbb{R}^{m}_{\geq}$, and $b, h \in \mathbb{R}^{n}_{\geq}$, CIP is defined as $\min\{h^T z : Az \geq a, z \leq b, z \in \mathbb{Z}^{n}_{\geq}\}$. Carr et al. [1] propose a Δ_1-approximation for this problem, where Δ_1 is the maximum number of non-zero coefficients in any row of A. Kolliopoulos and Young [7] design a $\mathcal{O}(\log m)$-approximation for CIP, and observe that this is asymptotically the best possible unless $\mathsf{P} = \mathsf{NP}$. This inapproximability threshold follows from the classic set cover problem, which does not admit a $o(\log m)$-approximation unless $\mathsf{P} = \mathsf{NP}$ [9]. The set cover problem is a special case of CIP where z is binary, and binary CIP is a particular case of P where $\ell_{ij} = c_{ij} = A_{ij}$, $v_{ij} = 0$, and $f_i = h_i$ for all $i \in [n]$ and $j \in [m]$. As a consequence, P does not admit a $o(\log m)$-approximation unless $\mathsf{P} = \mathsf{NP}$. Although there is little hope of designing better approximations for P in general, the case where the number of constraints is fixed remains largely unexplored. Observe

* Research supported by KU Leuven (C14/22/026) and CNPq-Brazil (404779/2025-5).

that P with fixed m encompasses classic cover problems, including multidimensional knapsack, facility location and supplier selection problems. Therefore, P remains NP-hard even when $m = 1$. For any fixed $m \geq 2$, P with m constraints does not admit a fully polynomial-time approximation scheme (FPTAS) unless $\mathsf{P} = \mathsf{NP}$. On the positive side, Frieze and Clarke [4] design a polynomial-time approximation scheme (PTAS) to the multidimensional knapsack cover problem. Kulik and Shachnai [8] show that no efficient PTAS (EPTAS) exists for this problem unless $\mathsf{W}[1] = \mathsf{FPT}$.

This work provides a theoretical study of P through the lens of approximation algorithms and perfect linear programming formulations. We show properties of the vertices of the polytope associated with P which are then used to decompose P into instances of a generalization of the multidimensional knapsack cover problem with one continuous variable per dimension. The proposed decomposition of P is used to design a PTAS for P with a fixed number of constraints. To the best of our knowledge, this is the first approximation scheme for such a general class of covering mixed-integer programs. Furthermore, in a sense, it is the best possible approximation for this problem since P admits no FPTAS, as previously observed. Using a similar strategy, we also design a PTAS for an assignment variant of P which includes general assignment problems (with fixed number of bins). A byproduct of this result is a PTAS for the knapsack cover with a single continuous variable, thereby improving upon the previously best-known 2-approximation algorithm by Zhao and Li [10].

Finally, we show a perfect compact formulation for the one-dimensional P with uniform bounds, that is, $\ell_i = \ell$ and $c_i = c$ for all $i \in [n]$. Although one-dimensional uniform P is a special case (with positive demand only in the last period) of a single-item lot-sizing problem with piecewise linear costs, which is known to be polynomial-time solvable [6], no tight formulation is known for this problem. Hence the proposed formulation contributes in this direction.

The details and proofs of the results mentioned above are available in [5].

2 Approximation scheme

We next present properties of an optimal solution of P. These allow us to demonstrate how to decompose P into a collection of instances of the following generalization of the multidimensional knapsack cover problem.

Multidimensional knapsack cover problem with continuous variables.
Input: $\eta, \mu \in \mathbb{Z}_{>}$, $\bar{f} \in \mathbb{Z}^{\eta}_{\geq}$, $\bar{v} \in \mathbb{Z}^{\mu}_{\geq}$, $\bar{c} \in \mathbb{Z}^{\mu}_{>}$, $\bar{w} \in \mathbb{Z}^{\eta\mu}_{\geq}$, and $\bar{d} \in \mathbb{Z}^{\mu}_{\geq}$.
Output: Subset $S \subseteq [\eta]$, and nonnegative $\alpha_j \leq \bar{c}_j$ for all $j \in [\mu]$ such that $\sum_{i \in S} \bar{w}_{ij} \geq d_j - \alpha_j$ for every $j \in [\mu]$.
Objective: Minimize $\sum_{i \in S} \bar{f}_i + \sum_{j \in [\mu]} \bar{v}_j \alpha_j$.

Note that an instance of MKC is feasible if and only if $\sum_{i \in [\eta]} \bar{w}_{ij} \geq d_j - \bar{c}_j$ for every $j \in [\mu]$. We assume henceforth that all considered instances of MKC are feasible. Let $j \in [m]$ and $k \in [n]$. We define $L_{kj} = \{i \in [n] \setminus \{k\} : v_{ij} > v_{kj}\} \cup \{i \in [n] \setminus \{k\} : v_{ij} = v_{kj} \text{ and } i < k\}$ and $C_{kj} = \{i \in [n] \setminus \{k\} : v_{ij} < v_{kj}\} \cup \{i \in$

$[n] \setminus \{k\} : v_{ij} = v_{kj}$ and $i > k\}$. Note that $\{L_{kj}, C_{kj}\}$ is a partition of $[n] \setminus \{k\}$ which only depends on the input instance, and can be computed in linear time.

Theorem 1. *Let I be an instance of* P. *There exists a function $g\colon [m] \to [n]$ such that I admits an optimal solution $(x, y) \in \mathbb{R}^{nm} \times \{0,1\}^n$ where, for each $j \in [m]$, $y_{g(j)} = 1$, $x_{ij} = \ell_{ij} y_i$ for all $i \in L_{g(j),j}$ and $x_{ij} = c_{ij} y_i$ for all $i \in C_{g(j),j}$.*

It follows from Theorem 1 that problem P boils down to finding such a function g and solving the MKC instance associated with g. The following corollary is used in the proposed algorithms.

Corollary 1. *Let I be an instance of* P. *There exists a collection $\mathcal{I}$ of $\mathcal{O}(n^m)$ instances of* MKC *such that* $\mathrm{OPT}(I) = \min_{I' \in \mathcal{I}} \mathrm{OPT}_{\text{MKC}}(I')$.

We design a polynomial-time approximation scheme for P (with m fixed), which is based on the proposed decomposition into MKC instances. Frieze and Clarke [4] devise a PTAS for the Multidimensional Knapsack Cover problem without continuous variables. We extend their PTAS to include exactly one continuous variable per dimension.

Theorem 2. *There exists a polynomial-time approximation scheme for* P *when m is fixed.*

Using a similar strategy, we also devise a PTAS for the packing variant of P. Moreover, we extend the approach previously described to design a PTAS for a related class of problems of the following form, which we denote by Q:

$$
\begin{aligned}
\max \quad & \sum_{i \in [n]} \sum_{j \in [m]} v_{ij} x_{ij} + \sum_{i \in [n]} \sum_{j \in [m]} f_{ij} y_{ij} && \\
\text{s.t.} \quad & \sum_{i \in [n]} x_{ij} \leq d_j && \forall j \in [m], \\
& \sum_{j \in [m]} y_{ij} \leq 1 && \forall i \in [n], \\
& \ell_{ij} y_{ij} \leq x_{ij} \leq c_{ij} y_{ij} && \forall i \in [n],\ j \in [m], \\
& y_{ij} \in \{0,1\} && \forall i \in [n],\ j \in [m],
\end{aligned}
$$

where m and n are positive integers, $v, \ell, c, f \in \mathbb{Z}^{nm}_{\geq}$ and $d \in \mathbb{Z}^{m}_{\geq}$ denote the nonnegative integer parameters. It is assumed, without loss of generality, that $d_j \geq c_{ij} \geq \ell_{ij}$ for all $i \in [n]$ and $j \in [m]$. Note that Q includes relevant problems as the Generalized Assignment Problem (GAP). In particular, when $v_{ij} = 0$ and $c_{ij} = \ell_{ij}$ for all $i \in [n]$ and $j \in [m]$, Q is equivalent to GAP, which is known to be APX-hard [2]. On the positive side, for any $\epsilon > 0$, there is an approximation for GAP with ratio $(e/(e-1) + \epsilon) \approx (1.582 + \epsilon)$ using an LP-rounding approach, which is currently the best known approximation for this problem [3].

3 Perfect formulation for the uniform case

A perfect formulation for $\mathrm{P_u}$ is obtained by integrating the decomposition into knapsack cover problems with one continuous variable and the union-of-polyhedra approach to the convex hull of sets.

Theorem 3. *P_u admits a perfect formulation with $\mathcal{O}(n^3)$ variables and $\mathcal{O}(n^3)$ constraints.*

References

[1] Carr, R.D., Fleischer, L.K., Leung, V.J., Phillips, C.A.: Strengthening integrality gaps for capacitated network design and covering problems. In: Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms, p. 106115, SODA '00, USA (2000)

[2] Chekuri, C., Khanna, S.: A polynomial time approximation scheme for the multiple knapsack problem. SIAM Journal on Computing **35**(3), 713–728 (2005)

[3] Fleischer, L., Goemans, M.X., Mirrokni, V.S., Sviridenko, M.: Tight approximation algorithms for maximum general assignment problems. In: Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithm - SODA '06, pp. 611–620, ACM Press, Miami, Florida (2006), ISBN 978-0-89871-605-4, https://doi.org/10.1145/1109557.1109624

[4] Frieze, A., Clarke, M.: Approximation algorithms for the m-dimensional 0–1 knapsack problem: Worst-case and probabilistic analyses. European Journal of Operational Research **15**(1), 100–109 (Jan 1984)

[5] Grobben, K., Moura, P.F.S., Yaman, H.: Approximation schemes for covering and packing mixed-integer programs with a fixed number of constraints (2025), URL https://arxiv.org/abs/2512.02571

[6] Hellion, B., Mangione, F., Penz, B.: A polynomial time algorithm to solve the single-item capacitated lot sizing problem with minimum order quantities and concave costs. European Journal of Operational Research **222**(1), 10–16 (Oct 2012)

[7] Kolliopoulos, S.G., Young, N.E.: Approximation algorithms for covering/packing integer programs. Journal of Computer and System Sciences **71**(4), 495–505 (2005)

[8] Kulik, A., Shachnai, H.: There is no EPTAS for two-dimensional knapsack. Information Processing Letters **110**(16), 707–710 (Jul 2010)

[9] Raz, R., Safra, S.: A sub-constant error-probability low-degree test, and a sub-constant error-probability pcp characterization of NP. In: Proceedings of the twenty-ninth annual ACM Symposium on Theory of Computing, pp. 475–484 (1997)

[10] Zhao, C., Li, X.: Approximation algorithms on 0–1 linear knapsack problem with a single continuous variable. Journal of Combinatorial Optimization **28**(4), 910–916 (Nov 2014)

Some formulations for the capacitated dispersion problem

Mercedes Landete[1][0000−0002−5201−0476], Juanjo Peiró[2][0000−0002−2944−599X], and Hande Yaman[3][0000−0002−3392−1127]

[1] Departamento de Estadística, Matemáticas e Informática. Instituto Centro de Investigación Operativa, Universidad Miguel Hernández de Elche, Spain. landete@umh.es

[2] Departament d'Estadística i Investigació Operativa. Facultat de Ciències Matemàtiques, Universitat de València, Spain juanjo.peiro@uv.es

[3] Research Centre for Operations Research and Business Statistics (ORSTAT). Faculty of Economics and Business. KU Leuven, Belgium hande.yaman@kuleuven.be

Abstract. This talk focuses on the capacitated dispersion problem for which we study several mathematical formulations using variables associated with different characteristics of its underlying graph. The dual bounds obtained from stronger but larger formulations are used to improve the strength of weaker but smaller formulations. Several sets of computational experiments are conducted to illustrate the usefulness of the findings, as well as the aptness of the formulations for different types of instances.

Keywords: Location science · Dispersion problem · Diversity problem.

1 Introduction

Dispersion problems are a family of $\mathcal{NP}$-hard optimization problems that consist in selecting a subset of elements from a given set in such a way that a distance measure between the selected elements is maximized. These problems are found in different application fields, specially when diversity is a constituent factor: biology, biodiversity and ecology, genetics, ethnicity and gender diversity, heterogeneous formation of work teams and committees, academic curriculum design, public facility location, market planning, financial portfolio design, etcetera.

In the context of dispersion models, the work [1] expanded the horizons of dispersion problems with the inclusion of capacity and cost constraints, originally motivated by applications in Location Science.

We will focus on a version of the constrained dispersion problem known as the Capacitated Dispersion Problem (CDP). In it, given a set V of n elements that we call nodes or facilities, a positive capacity c_i for each node i, a nonnegative distance d_{ij} between any pair of distinct nodes i and j, and a positive demand B to cover, we would like to find a subset V' of nodes such that their sum of capacities is large enough to cover the demand, i.e., $\sum_{i \in V' \subseteq V} c_i \geq B$, and the

nodes in V' are as distant as possible from one another, i.e., $\min_{i,j \in V' \subseteq V : i \neq j} d_{ij}$ is maximum. This version was originally proposed in [1] and coined as the MAX-DIST/CAP problem.

In [2], the CDP was first formulated as the following non-linear integer program:

$$\begin{aligned} \max \quad & \min_{i,j \in V \,:\, i<j} \; d_{ij} x_i x_j \\ \text{s.t.:} \quad & \sum_{i \in V} c_i x_i \geq B, \\ & x_i \in \{0,1\}, \quad \forall i \in V, \end{aligned}$$

where variable x_i takes value 1 if node $i \in V$ is selected (facility i is open), and 0 otherwise.

In this talk, we will present some integer linear formulations for the CDP and see how they computationally behave. We will also outline an approach that makes use of stronger and larger formulations to improve the performance of weaker and smaller formulations.

All the results we will present are based on the ones that we introduced in [3].

Acknowledgments. This work was supported by Gobierno de España through Ministerio de Ciencia e Innovación and by ERDF—A way of making Europe—under projects PID2021-122344NB-I00, PID2021-124975OB-I00 and RED2018-102363-T, by Generalitat Valenciana under projects Prometeo CIPROM/2024/34 and Emergent CIGE/2024/132. Research of the third author was supported by Internal Funds KU Leuven (Grant C14/22/026).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this extended abstract.

References

1. Rosenkrantz, D., Tayi, K., Ravi, S. S.: Facility dispersion problems under capacity and cost constraints. Journal of Combinatorial Optimization **4**, 7–33 (2000).
2. Peiró, J., Jiménez, I., Laguardia, J., Martí, R.: Heuristics for the capacitated dispersion problem. International Transactions in Operational Research **28**(1), 119–141 (2021).
3. Landete, M., Peiró, J., Yaman, H.: Formulations and valid inequalities for the capacitated dispersion problem. Networks **81**(2) 294–315 (2023).

Geoffrion's theorem beyond finiteness and rationality

Santanu S. Dey[1], Frédéric Meunier[2,3], and Diego A. Morán R.[3]

[1] H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology santanu.dey@isye.gatech.edu
[2] CERMICS, ENPC, Institut Polytechnique de Paris frederic.meunier@enpc.fr
[3] Industrial and Systems Engineering Department, Rensselaer Polytechnic Institute morand@rpi.edu

Abstract. Geoffrion's theorem is a fundamental result from mathematical programming assessing the quality of Lagrangian relaxation, a standard technique to get bounds for integer programs. An often implicit condition is that the set of feasible solutions is finite or described by rational linear constraints. However, we show through concrete examples that the conclusion of Geoffrion's theorem does not necessarily hold when this condition is dropped. We then provide sufficient conditions ensuring the validity of the result even when the feasible set is not finite and cannot be described using finitely-many rational linear constraints.

Keywords: Lagrangian relaxation · Geoffrion's theorem · Non-linear integer programming.

1 Lagrangian relaxation and Geoffrion's theorem

The exact resolution of integer programs relies in a crucial way on bounds that help assess the quality of feasible solutions. A popular systematic and efficient way to produce lower bounds on integer programs formulated as minimization problems, and more generally dual bounds, is *Lagrangian relaxation*, introduced in 1970 by Held and Karp for the resolution of the traveling salesman problem [3,4], which works as follows.

Consider an integer program

$$\begin{aligned} \text{minimize } & c \cdot x \\ \text{s.t. } & Ax \geqslant b \\ & x \in X\,, \end{aligned} \tag{1}$$

where $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, and $X \subseteq \mathbb{Z}^n$. The function

$$\mathcal{G}\colon \lambda \in \mathbb{R}^m_+ \longmapsto \inf\{c \cdot x + \lambda \cdot (b - Ax)\colon x \in X\} \in \overline{\mathbb{R}}$$

is a *dual function* of (1). As is well known, $\mathcal{G}(\lambda)$ provides a lower bound on the optimal value of (1). *Lagrangian relaxation* involves using such lower bounds

to solve (1). The method often aims to identify the tightest possible bound achievable in this manner, namely the quantity $v^L := \sup\{\mathcal{G}(\lambda) \colon \lambda \in \mathbb{R}^m_+\}$.

One of the results that best explains the strength of Lagrangian relaxation is Geoffrion's theorem [1, Theorem 1]. It considers the following optimization problem:

$$\begin{aligned} \text{minimize } & c \cdot x \\ \text{s.t. } & Ax \geqslant b \\ & x \in \operatorname{conv}(X)\,, \end{aligned} \tag{2}$$

in the space of the original variables. Denote by $v^\star$ the optimal value of (2) which is defined as the infimum of $c \cdot x$ over the feasible region. Geoffrion's theorem considers the case where X is finite or the set of integer points contained in a rational polyhedron, and states that if (1) is feasible, then $v^L = v^\star$. (We remind the reader that a *rational polyhedron* is the intersection of finitely many half-spaces, each of which being described using rational data.)

The proof of Geoffrion's theorem relies crucially on $\operatorname{conv}(X)$ being polyhedral, which is the case when X is finite or when X is the set of integer points within a rational polyhedron (a consequence of Meyer's theorem [5]). The theorem does not say anything about the case when X is the set of integer points of an unbounded non-rational polyhedron. The only place in the literature we have been able to find where this fact is explicitly mentioned is a sentence in a paper by Guignard [2, p.157]:

> "... this result may not be true if the constraint matrices are not rational, or more precisely for non-rational polyhedra that are not equal to the convex hull of their extreme points."

Our contributions are twofold:

- Concrete examples where the conclusion of Geoffrion's theorem does not hold because X is precisely the collection of integer points contained in a polyhedron whose defining inequalities have irrational coefficients.
- Sufficient conditions ensuring that the conclusion of Geoffrion's theorem holds, including cases where X is an infinite set of integer points that cannot be described as the integer points in a polyhedron, i.e., X may be the set of integer points in a general convex set.

When X is finite or can be described as the integer points in a rational polyhedron, its convex hull is closed (also a consequence of Meyer's theorem). This is not necessarily the case when X is the set of integer points in an unbounded non-rational polyhedron. Thus, to investigate Geoffrion's theorem "beyond finiteness and rationality," considering the following optimization problem could also be relevant, which will be confirmed by Example 1 in the next subsection:

$$\begin{aligned} \text{minimize } & c \cdot x \\ \text{s.t. } & Ax \geqslant b \\ & x \in \overline{\operatorname{conv}}(X)\,, \end{aligned} \tag{3}$$

where $\overline{\text{conv}}(X)$ is the closure of $\text{conv}(X)$. Denote by $\bar{v}^\star$ the optimal value of (3) which is again defined to be the infimum of $c \cdot x$ over the feasible region. For sake of completeness, we state Geoffrion's theorem for the optimization problems (2) and (3) simultaneously.

Theorem 1 (Geoffrion's theorem [1]). *Suppose that X is finite or formed by the integer points of a rational polyhedron. If (1) is feasible, then $v^L = \bar{v}^\star = v^\star$.*

2 Two examples showing the failure of Geoffrion's theorem beyond finiteness and rationality

The first example shows that the vanilla version of Geoffrion's theorem—about the optimization problem (2)—does not hold in full generality when X is the set of integer points of a non-rational polyhedron.

Example 1. Consider the integer program

$$\begin{aligned} \text{minimize } & -x \\ \text{s.t. } & -\sqrt{2}x + y \geqslant 0 \\ & -\sqrt{2}x + y \leqslant 0 \\ & x, y \in \mathbb{Z}_+ \,. \end{aligned} \tag{4}$$

Set $X = \{(x, y) \in \mathbb{Z}^2_+ \colon -\sqrt{2}x + y \leqslant 0\}$. It can be checked that, on the one hand, $v^L = -\infty$, and, on the other hand, $v^\star = 0$. The program (4) is therefore such that $v^L < v^\star$. ◇

In Example 1, the closure of $\text{conv}(X)$ is actually $\{(x, y) \in \mathbb{R}^2_+ \colon -\sqrt{2}x + y \leqslant 0\}$. Taking in (2) the closure of $\text{conv}(X)$ instead would ensure the validity of Geoffrion's theorem for this example. However, even the equality $v^L = \bar{v}^\star$ does not hold for some cases where X is the set of integer points of an unbounded non-rational polyhedron.

Example 2. Let $P \coloneqq \text{conv}(P^1 \cup P^2)$ where $P^1 \coloneqq \big\{(x, y, z) \in \mathbb{R}^3 \colon \sqrt{2}x - y \geqslant 0,\ y \geqslant 0,\ z = 0\big\}$ and $P^2 \coloneqq \big\{(x, y, z) \in \mathbb{R}^3 \colon \sqrt{2}x - y \geqslant 0,\ y \geqslant 0,\ x \geqslant 1/2,\ z = 1\big\}$. Consider the integer program

$$\begin{aligned} \text{minimize } & -z \\ \text{s.t. } & -\sqrt{2}x + y \geqslant 0 \\ & (x, y, z) \in X \,, \end{aligned} \tag{5}$$

where $X \coloneqq P \cap \mathbb{Z}^3$. It can be checked that, on the one hand, $v^L = -1$, and, on the other hand, $\bar{v}^\star = 0$. The program (5) is therefore such that $v^L < \bar{v}^\star$. ◇

3 Sufficient conditions for Geoffrion's theorem to hold.

The examples of Section 2 are "degenerate" in the sense that the set of feasible solutions of (3) is located on the boundary of $\text{conv}(X)$. This is not coincidental. The relative interior of $\text{conv}(X)$ having a non-empty intersection with $\{x \in \mathbb{R}^n : Ax \geqslant b\}$ is nothing else than Slater's condition, and basic results from convex programming ensure that Geoffrion's theorem holds with $v^L = \bar{v}^\star = v^\star$. The challenge for extending Geoffrion's theorem beyond finiteness and rationality is thus to understand integer programs whose feasibility set lies on the boundary of $\text{conv}(X)$. Our three main results in this regard are presented next.

When X is the set of integer points of a non-rational polyhedron, $\overline{\text{conv}}(X)$ is not necessarily a polyhedron. The first result deals with the case when $\overline{\text{conv}}(X)$ is *locally polyhedral*, namely, that its intersection with every box is a polytope.

Theorem 2. *Suppose that $\overline{\text{conv}}(X)$ is locally polyhedral. If (3) admits an optimal solution, then $v^L = \bar{v}^\star$.*

Note that while our original aim was to explore the case where X is the set of integer points in an unbounded polyhedron which is not necessarily rational, the above result provides sufficient conditions even is cases where X is the set of integer points in a non-polyhedral set. In fact, our proof of Theorem 2 is quite general and does not even assume that X is a set of integer points.

The second result states that when the feasible region is included in the plane, then Geoffrion's theorem holds for programs of the form (3).

Theorem 3. *Suppose that $n \leqslant 2$. If (1) is feasible, then $v^L = \bar{v}^\star$.*

Example 1 shows that under the conditions of Theorems 2 and 3, while $v^L = \bar{v}^\star$ must hold, the equality $v^L = v^\star$ may not necessarily hold. (Note that however we always have $\bar{v}^\star \leqslant v^\star$.)

The third result somehow shows a reverse phenomenon with regards to the standard assumptions for Geoffrion's theorem to hold: instead of requiring the rationality of the constraints defining X, we require rationality for the other constraints.

Theorem 4. *Suppose that the system $Ax \geqslant b$ is a rational single-row system. If $\inf\{c \cdot x : x \in X\}$ is finite, then $v^L = \bar{v}^\star = v^\star$.*

References

1. Geoffrion, A.M.: Lagrangian relaxation for integer programming. Mathematical Programming Study **2**, 82–114 (1974)
2. Guignard, M.: Efficient cuts in Lagrangean 'relax-and-cut' schemes. European Journal of Operational Research **105**(1), 216–223 (1998)
3. Held, M., Karp, R.M.: The traveling-salesman problem and minimum spanning trees. Operations Research **18**(6), 1138–1162 (1970)
4. Held, M., Karp, R.M.: The traveling-salesman problem and minimum spanning trees: Part II. Mathematical Programming **1**(1), 6–25 (1971)
5. Meyer, R.R.: On the existence of optimal solutions to integer and mixed-integer programming problems. Mathematical Programming **7**(1), 223–235 (1974)

A Feature Engineering-Based Market Prediction for the Traveling Purchaser Problem

Aykut Sen[1][0009-0004-7077-3209] and Ilker Kucukoglu[1][0000-0002-5075-0876]

[1] Bursa Uludag University, Gorukle Campus 16059 Bursa, Turkey
aykutsen@uludag.edu.tr, ikucukoglu@uludag.edu.tr

Keywords: Feature Engineering, Machine Learning, Traveling Purchaser Problem.

1 Problem Definition

The Traveling Purchaser Problem (TPP) is a well-known comprehensive generalization of the classic Traveling Salesman Problem (TSP) that integrates purchasing decisions and is classified as an NP-hard problem [1]. TPP involves selecting markets that balance price and stock availability for the demand products while at the same time determining an optimal visiting sequence that minimizes transportation costs [1]. Early TPP studies mainly relied on exact solution methods, which guarantee optimality but become computationally impractical as the problem size grows. To address this, heuristic/meta-heuristic approaches have been used [1]. While heuristic methods can reach solutions faster than exact methods, they often fall into the trap of getting stuck at local optimum results [2].

Despite the increase in Machine Learning (ML) applications in solving combinatorial optimization problems, the potential of candidate reduction of the search space for TPP has not been explored. Based on this motivation, this study introduces a feature-engineering-based market prediction approach to reduce the search space for the TPP. In this context, five features are proposed to rank markets according to different priority metrics. To evaluate the performance of the proposed approach, a well-known TPP benchmark problem set is used. The similarity between the proposed rankings and the optimal market sets was evaluated. Results show that the markets ranked by the proposed solution approach largely overlapped with the markets that yielded optimal results.

2 Methodology

The TPP involves a set of markets M and a set of products K, where the total demand for each product d_k must be precisely met from the markets. Each market i in M has a limited capacity q_{ik} for product k and offers it at a unit price p_{ik}. In order to complete the procurement operation, the purchaser starts its tour from a depot node (0), visits a number of markets in M, and returns to the depot node again. The TPP is defined on a complete directed graph $G = (V, A)$ where $V = (M \cup \{0\})$ is the node set and $A =$

$\{(i,j): i,j \in V, i \neq j\}$ is the arc set. Each arc in the problem is associated with a traveling cost. The goal is to satisfy all product demands while minimizing the combined purchase and travel costs. For a formal mathematical formulation of the routing and procurement model, we refer to the review by Manerba et al. [1].

Based on the given definition above, the TPP includes three decision layers: selecting the markets to be visited, the procurement plan to satisfy the product demand, and the route plan of the purchaser. In this study, we implemented reproducible feature engineering to score and rank markets for reducing the search space of the TPP. In case the markets to be visited in the TPP are known, the problem transforms into the classical TSP. Consequently, solving the problem by an exact approach becomes easier. Based on market coordinates, product demand, and offering data, five features (F1, …, F5) were constructed to capture potential purchase savings, price competitiveness, travel-effort-based value, supplier substitution, and demand-weighted savings.

Feature 1 (F1): TSP (Total Saving Potential): It measures the theoretical profit that can be obtained by comparing market prices to the average of other markets. Equation 1 presents the TSP for each market, in which $\bar{p}_k$ represents the average price of product k in the problem.

$$\text{TSP} = \sum_{k \in K_i} \max(0, \bar{p}_k - p_{ik}) \, . \, d_k \qquad i \in M \tag{1}$$

Feature 2 (F2): CPR (Competitive Price Ratio): The proportion of products for which a supplier provides the lowest available price among all competitors. $p_{Min,k}$ presents the minimum price of product k across all markets. Π Indicator function; equals 1 if market i offers the minimum price for product k, 0 otherwise.

$$\text{CPR} = \frac{\sum_{k \in K_i} \Pi(p_{ik} = p_{Min,k})}{|K_i|} \tag{2}$$

Feature 3 (F3): NVD (Net Value Density): The net economic benefit generated by a supplier after accounting for the travel cost required to visit it. c_{0i} and c_{i0} denote the costs from and to the depot.

$$NVD = \frac{TSP - (c_{i0} + c_{0i})}{K_i} \tag{3}$$

Feature 4 (F4): RS (Replaceability Score): A measure of how easily a supplier can be substituted, based on the availability of alternative suppliers for the same products. M_k presents the set of markets that offer product k.

$$\text{RS} = \frac{1}{|K_i|} \sum_{k \in K_i} (|M_k| - 1) \tag{4}$$

Feature 5 (F5): DWS (Demand-Weighted Savings): The aggregate savings achieved by a supplier, emphasizing price advantages on high-demand products.

$$DWS = \sum_{k \in K_i} (\bar{p}_k - p_{ik}) . d_k \tag{5}$$

All features were normalized within each instance to ensure comparability, then aggregated into a single overall score defined as 0.25.TSP + 0.20.CPR + 0.25.NVD + 0.10.(1-RS) + 0.20.DWS. The weights of each feature are determined with respect to the preliminary studies.

3 Computational Results

The computational experiments were conducted on the benchmark test problems proposed by Laporte et al. [3]. Each instance is defined by 50 products and 50 markets, referred to as 50x50 configuration. In computational studies, the markets in each problem are ranked regarding the F1, F2, F3, F4, F5, and overall score. To evaluate the performance of the rankings, the problems are solved by using the GUROBI solver. The rankings are compared to the set of markets visited in the optimal solution.

Table 1 summarizes the computational results and prediction accuracy of features under different values of λ, which controls the maximum number of markets allowed in a feasible solution, with higher values making the selection process more restrictive. |M*| indicates the size of the optimal market set and Acc%(F1)–Acc%(F5) present the prediction accuracy of each feature. The accuracy is computed as the percentage of markets correctly predicted by the feature relative to the number of markets in the optimal solution. Acc%(Total) reports the prediction accuracy of the overall feature score. The values in bold highlight which feature is ranked 1st, and the First Place Count shows how often each feature ranks first.

The results show that the proposed scoring framework produces stable and interpretable rankings. As shown in Table 1, while overall accuracy decreases with increasing λ, some features continue to provide high prediction accuracy, demonstrating robustness and potential for further improvement. Especially when the overall feature score is 0%, F3 still achieves a relatively high prediction accuracy of around 50%, highlighting its individual discriminative power despite stricter selection conditions.

Table 1. Computational Results

λ	Size	|M*|	Acc%(F1)	Acc%(F2)	Acc%(F3)	Acc%(F4)	Acc%(F5)	Acc%(Total)
0.1	50x50	49	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
0.1	50x50	49	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
0.1	50x50	49	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
0.1	50x50	49	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
0.1	50x50	46	**95.65**	93.48	**95.65**	93.48	93.48	**95.65**
0.5	50x50	45	**95.56**	91.11	**95.56**	91.11	93.33	**95.56**
0.5	50x50	47	97.87	95.74	**100.00**	95.74	97.87	**100.00**
0.5	50x50	45	**93.33**	**93.33**	**93.33**	91.11	91.11	**93.33**
0.5	50x50	47	**97.87**	95.74	**97.87**	95.74	95.74	**97.87**
0.5	50x50	44	93.18	88.64	**95.45**	88.64	90.91	93.18
0.7	50x50	39	**94.87**	89.74	**94.87**	74.36	87.18	92.31
0.7	50x50	43	**90.70**	86.05	88.37	**90.70**	88.37	88.37
0.7	50x50	41	**90.24**	**90.24**	**90.24**	82.93	87.80	**90.24**
0.7	50x50	38	**92.11**	81.58	89.47	78.95	81.58	**92.11**
0.7	50x50	37	83.78	83.78	**86.49**	83.78	75.68	83.78
0.8	50x50	34	73.53	73.53	**76.47**	67.65	76.47	**76.47**
0.8	50x50	35	**77.14**	74.29	71.43	74.29	71.43	68.57
0.8	50x50	30	**73.33**	70.00	70.00	70.00	66.67	70.00
0.8	50x50	28	75.00	60.71	**82.14**	53.57	75.00	78.57

Table 1. Computational Results (Continue)

0.8	50x50	32	68.75	68.75	**71.88**	**71.88**	62.50	**71.88**
0.9	50x50	23	52.17	56.52	**60.87**	30.43	52.17	56.52
0.9	50x50	25	**60.00**	56.00	**60.00**	**60.00**	56.00	56.00
0.9	50x50	21	52.38	57.14	38.10	42.86	**66.67**	**66.67**
0.9	50x50	20	50.00	60.00	**80.00**	40.00	45.00	65.00
0.9	50x50	15	**60.00**	33.33	26.67	26.67	46.67	40.00
0.95	50x50	17	35.29	35.29	29.41	29.41	**47.06**	35.29
0.95	50x50	16	50.00	37.50	**75.00**	18.75	43.75	50.00
0.95	50x50	14	**57.14**	35.71	50.00	28.57	50.00	**57.14**
0.95	50x50	15	33.33	26.67	**46.67**	13.33	33.33	40.00
0.95	50x50	14	42.86	**57.14**	35.71	28.57	35.71	42.86
0.99	50x50	7	14.29	14.29	**42.86**	0.00	28.57	28.57
0.99	50x50	13	38.46	30.77	**69.23**	15.38	30.77	53.85
0.99	50x50	10	10.00	20.00	**50.00**	10.00	10.00	0.00
0.99	50x50	12	16.67	**25.00**	16.67	16.67	16.67	16.67
0.99	50x50	11	**27.27**	**27.27**	18.18	18.18	18.18	**36.36**
First Place Count			18	9	24	7	6	16

Out of 35 problems, F3 ranks first in 24 cases (68.6%), highlights the importance of evaluating markets not only based on the prices of their products but also by considering the cost of travel to that market. F1 ranking first in 18 cases (51.4%), confirms that although travel costs are critical, markets with significant price advantages remain highly competitive in the selection process. F2, ranking first in 25.7% of the instances, shows that consistently offering the lowest price for individual products provides a competitive but limited advantage. F4 (20.0%) highlights the relevance of market substitutability, while F5 (17.1%) indicates that demand-weighted price savings are beneficial in specific cases. This distribution indicates that F3 and F1 play a leading role across the problem set, making them focal points for further analysis. Accordingly, for future studies, the proposed solution approach can be extended by developing up to 25 features and by adopting more systematic methods for determining feature weights when creating the overall score may improve the performance of the solution approach.

Disclosure of Interests. Both authors have no competing interests to declare that are relevant to the content of this article.

Acknowledgments. This study was supported by Bursa Uludag University Research Projects Coordination Office under the Grant Number FYL-2026-2659. The authors thank to BUU BAP Unit for their supports.

References

1. Manerba. D.. Mansini. R.. Riera-Ledesma. J.: The Traveling Purchaser Problem and its variants. Eur. J. Oper. Res. 259. 1–18 (2017). https://doi.org/10.1016/J.EJOR.2016.12.017.
2. Blum. C.. Roli. A.: Metaheuristics in combinatorial optimization. ACM Computing Surveys (CSUR). 35. 268–308 (2003). https://doi.org/10.1145/937503.937505.
3. Laporte, G., Riera-Ledesma, J., Salazar-González, J.-J.: A Branch-and-Cut Algorithm for the Undirected Traveling Purchaser Problem. Oper. Res. 51, 940–951 (2003). https://doi.org/10.1287/opre.51.6.940.24921.

A Benders Decomposition Framework for the k-Defensive Domination Problem

Bilge Varol[1,2], Tınaz Ekim[2], and Kübra Tanınmış[3]

[1] Boğaziçi University, Istanbul, Türkiye tinaz.ekim@bogazici.edu.tr
[2] Turkish-German University, Istanbul, Türkiye bilge.varol@tau.edu.tr
[3] Koç University, Istanbul, Türkiye ktaninmis@ku.edu.tr

Abstract. The k-defensive domination problem seeks a minimum defender set in a graph that can defend any simultaneous attack on up to k vertices through feasible defender–attacker matchings. This study presents the first exact solution approach for general graphs, based on an integer programming formulation within a Benders decomposition framework. A key contribution is a purely combinatorial cut separation method that identifies Hall violators directly from the graph structure, thereby replacing LP-based subproblem solves with efficient graph operations. When combined with clique cover–based warm-start heuristics, the proposed framework outperforms both the compact IP formulation and classical Benders decomposition methods.

Keywords: Defensive domination · Benders decomposition · Combinatorial cuts · Network security.

1 Introduction

Domination in graphs is an extensively studied problem [6]. The defensive domination problem, introduced by Farley and Proskurowski [4], offers advanced modeling power for applications in network security and disaster management. It addresses situations where threats may occur simultaneously at any $k > 1$ unknown nodes, requiring the ability to counter all possible attack scenarios. Since a defender at a node can only assist one node under attack, an assignment between defenders and attacked nodes must exist for every possible scenario.

There are few studies on defensive domination. Ekim et al. [3] showed that k-defensive domination is NP-complete in split graphs for fixed k; if k is not fixed, the problem is unlikely to be in co-NP. Efficient algorithms exist only for restricted classes: threshold and co-chain graphs [3], and proper interval graphs [2]. Recently, Henning et al. [7] proved NP-completeness for bipartite graphs with fixed $k > 0$, APX-completeness for bounded-degree graphs, and polynomial solvability for complete bipartite graphs.

Let $G = (V, E)$ be a graph with vertex set V and edge set E. The open neighborhood of v is $N(v) = \{u \in V : (u, v) \in E\}$, and the closed neighborhood is $N[v] = N(v) \cup \{v\}$. A *k-attack* is a set $A = \{a_1, \ldots, a_k\} \subseteq V$ of vertices simultaneously under attack. Attack A can be *countered* by defenders $X \subseteq V$

if there exists an injective $f : A \to X$ with $f(a_i) = a_i$ or $(a_i, f(a_i)) \in E, \quad \forall 1 \leq i \leq k$. A set $D \subseteq V$ is a *k-defensive dominating set* if D counters any k-attack. The k-defensive domination problem seeks minimum cardinality D. Moreover, [3] proved that given G, integer k, and $D \subseteq V(G)$, deciding whether D is k-defensive is co-NP-hard. This implies that even verifying feasibility is hard, making heuristics challenging. No exact approach existed for general graphs prior to this work.

As a contribution to the literature, we develop (i) the first IP formulation for general graphs, (ii) a Benders decomposition with branch-and-benders-cut (BB), (iii) a combinatorial cut separation avoiding LP subproblems, and (iv) clique cover-based warm-start heuristics.

2 Problem Formulation and Solution Approaches

Let $\mathcal{A}$ denote all k-attack scenarios, $N[v]$ the closed neighborhood of v, and $N[A] = \bigcup_{j \in A} N[j]$. Define binary variables $x_i = 1$ if $i \in D$, and $y_{ij}^A = 1$ if defender i counters attacked vertex j in scenario A. The compact integer programming (IP) formulation is presented.

$$\text{(IP)} \quad \min \ \sum_{i \in V} x_i \tag{1}$$

$$\text{s.t.} \ \sum_{i \in N[j]} y_{ij}^A = 1 \qquad \forall j \in A, A \in \mathcal{A} \tag{2}$$

$$\sum_{j \in A \cap N[i]} y_{ij}^A \leq x_i \qquad \forall i \in N[A], A \in \mathcal{A} \tag{3}$$

$$x_i, y_{ij}^A \in \{0, 1\} \tag{4}$$

Constraint (2) ensures each attacked vertex is defended exactly once; (3) links assignments to defender selection. Crucially, when defender variables x are fixed to binary values $\hat{x}$, the subproblem over y variables becomes a bipartite matching problem between attacked vertices A and available defenders $\{i : \hat{x}_i = 1\}$. The constraint matrix of this subproblem is totally unimodular, guaranteeing that the LP relaxation yields integer optimal solutions. Therefore, y variables can be relaxed to continuous ($y_{ij}^A \geq 0$) without loss of optimality, which we exploit in our Benders decomposition.

Benders Decomposition: We partition variables: master problem (MP) selects defenders x, while the subproblem $\text{SP}(\hat{x})$ verifies feasibility for candidate $\hat{x}$. Since attack scenarios are independent, SP decomposes: $\text{SP}(\hat{x}) = \sum_{A \in \mathcal{A}} \text{SP}_A(\hat{x})$. Taking the dual of SP_A:

$$(\mathbf{DSP}_A(\hat{x})) \quad \max \ \sum_{j \in A} \alpha_j - \sum_{i \in N[A]} \beta_i \hat{x}_i \tag{5}$$

$$\text{s.t.} \ \ \alpha_j - \beta_i \leq 0 \quad \forall i \in N[j], j \in A; \quad \beta_i \geq 0 \tag{6}$$

When $\text{SP}_A(\hat{x})$ is infeasible, the dual is unbounded with ray $(\hat{\alpha}, \hat{\beta})$, yielding the Benders feasibility cut $\sum_{j \in A} \hat{\alpha}_j + \sum_{i \in N[A]} \hat{\beta}_i x_i \leq 0$. We integrate cut generation within branch-and-benders-cut via lazy constraint callbacks.

Combinatorial Cut Separation: Theorem 1 establishes that Hall violators define valid improving rays for $\mathbf{DSP}_A(\hat{x})$, enabling a combinatorial separation

routine that bypasses the need for solving an LP. To maximize cut strength, the expansion of S toward larger violators is prioritized. Specifically, if $S \subset S'$ and $|S' \setminus S| \geq |N[S'] \setminus N[S]|$, the cut generated from S' dominates the cut from S.

Theorem 1. *Let $\hat{x}$ be a defender configuration, $A \in \mathcal{A}$ an attack scenario, and $S \subseteq A$ a partial attack s.t. $|\{i \in N[S] : \hat{x}_i = 1\}| < |S|$. The vector $(\hat{\alpha}, \hat{\beta})$ defined by $\hat{\alpha}_S = \mathbf{1}$, $\hat{\beta}_{N[S]} = -\mathbf{1}$, and zero otherwise, is an improving ray of $\boldsymbol{DSP}_A(\hat{x})$.*

Cut Generation Algorithm: For fixed candidate $\hat{x}$ and attack A, we find Hall violators via recursive search. Starting from each $v \in A$, we explore subsets $S \subseteq A$ such that $|\{i \in N[S] : \hat{x}_i = 1\}| < |S|$. A depth-first search is employed to prioritize expanding S to find larger violators yielding stronger cuts based on Proposition 1. We maintain a buffer of $C_{\max}$ cuts, ranked by cut violation at $\hat{x}$. Weaker cuts are replaced when stronger ones are found. Within each callback, up to $C_{\max}$ cuts are added (multi-cut strategy).

Warm-Start Heuristics: To obtain a clique cover $\mathcal{C}$ for the graph G, a graph coloring problem is solved on the complement $\bar{G}$ using the DSATUR heuristic [1] or, alternatively, an optimal cover is derived by computing a perfect elimination ordering (PEO) via Maximum Cardinality Search (MCS) [5]. Rather than a baseline selection of $\min(k, |C|)$ vertices per clique, a matching-based greedy refinement is employed to minimize redundancy. Cliques are processed in decreasing order of size; for each clique $C \in \mathcal{C}$, a matching M is computed between the current defender set D and the vertices of C. Only $\min(k, |C| - |M|)$ additional vertices from the unmatched portion of C are added to D. This procedure maximizes defender reuse across cliques, resulting in a significantly tighter initial upper bound for the defensive set.

3 Computational Results and Discussions

All experiments use C++ with Gurobi 12.0 on an M2 Mac (16GB), 600s time limit. Table 1 reports results on Erdős–Rényi graphs and Table 2 on chordal graphs. Across all instances, the clique-based warm-start heuristics yield high-quality initial upper bounds, substantially improving over the trivial $UB{=}n$ initialization. Moreover, the combinatorial cut separation with warm starts achieves the best overall computational time and optimality gap.

Parameter Sensitivity: We investigated the effect of cut buffer size $C_{\max}$ and after careful considerations, it is recommended that selecting $C_{\max} = [20, 50]$ as robust defaults.

Future work will expand experiments to Barabási–Albert graphs and investigate bilevel extensions of the proposed framework.

Acknowledgments. Supported by the European Commission's Horizon Europe Research and Innovation programme through the Marie Skłodowska-Curie Actions Staff Exchanges (MSCA-SE) under Grant Agreement No. 101182819 (COVER); Bilge Varol was supported by TÜBİTAK BİDEB 2211.

Disclosure of Interests. The authors declare that they have no competing interests.

Table 1. Comparison of all solution approaches on Erdős–Rényi graphs.

k	n	ρ	IP Time	IP Gap	IP Op.	Benders Time	Benders Gap	Benders Op.	BBA Time	BBA Gap	BBA Op.	BBC Time	BBC Gap	BBC Op.	BBMC Time	BBMC Gap	BBMC Op.	BBMC+H Time	BBMC+H Gap	BBMC+H Op.	UB_0	Opt. val.
2	50	0.2	21	0	5	28	0	5	9	0	5	1	0	5	0	0	5	0	0	5	12.2	9.1
		0.5	123	0	5	43	0	5	32	0	5	0	0	5	0	0	5	0	0	5	8.2	4.8
		0.8	147	0	5	7	0	5	3	0	5	0	0	5	0	0	5	0	0	5	8.0	3
	100	0.2	–	49	0	–	40	0	–	35	0	381	5	3	279	0	5	270	0	5	14.4	10.6
		0.5	–	57	0	591	35	1	590	25	1	20	0	5	10	0	5	7	0	5	9.0	5.5
		0.8	545	20	1	580	0	5	514	0	5	1	0	5	0	0	5	0	0	5	11.6	3.1
	150	0.2	–	62	0	–	50	0	–	32	0	–	30	0/5	–	28	0	–	27	0	17.0	12.6
		0.5	–	98	0	–	43	0	–	31	0	591	13	1	433	7	3	407	7	3	11.4	6.2
		0.8	–	98	0	–	25	0	580	0	5	16	0	5	3	0	5	2	0	5	13.8	3.2
3	50	0.2	–	51	0	–	30	0	–	19	0	13	0	5	5	0	5	4	0	5	12.8	10.6
		0.5	–	65	0	444	0	5	481	0	5	8	0	5	2	0	5	1	0	5	8.2	6.4
		0.8	–	29	0	584	0	5	570	0	5	1	0	5	1	0	5	0	0	5	9.2	4.4
	100	0.2	–	–	0	–	–	0	–	–	0	–	35	0	–	24	0	–	26	0	14.6	11.2
		0.5	–	–	0	–	–	0	–	–	0	–	19	0	281	3	4	137	0	5	9.0	5.1
		0.8	–	–	0	–	–	0	–	–	0	61	0	5	8	0	5	4	0	5	12.8	3.2

Time: Runtime (s), **Gap:** Optimality gap (%), **Op.:** Number of instances solved to optimality (–/5), **IP:** IP formulation, **Benders:** Classical Benders, **BBA:** BB with aggregated cuts, **BBC:** BB with decomposed cuts (1 cut/iter), **BBMC:** BB with decomposed cuts, multiple cuts per iteration with buffer limit (50 cuts), **H:** Heuristic warm start(DSATUR coloring and matching-based reduction), **Opt. val.:**Optimal objective.

Table 2. Comparison of solution approaches on Chordal graphs.

k	n	**Time | Gap | Op.(–/5)** BBC ρ=0.2	BBC ρ=0.5	BBC ρ=0.8	BBMC ρ=0.2	BBMC ρ=0.5	BBMC ρ=0.8	**Time | Gap | Op.(–/5)** BBMC+H ρ=0.2	BBMC+H ρ=0.5	BBMC+H ρ=0.8	UB_0
2	500	21|0|5	38|0|5	51|0|5	4|0|5	5|0|5	5|0|5	1|0|5|63	2|0|5|29	2|0|5|11	
	750	104|0|5	146|0|5	155|0|5	9|0|5	11|0|5	13|0|5	2|0|5|78	7|0|5|39	7|0|5|15	
	1000	335|0|5	483|0|5	368|0|5	17|0|5	28|0|5	24|0|5	7|0|5|91	20|0|5|44	18|0|5|18	
3	500	–|–|0	–|–|0	–|–|0	–|34|0	–|15|0	–|0|5	263|0|5|73	428|0|5|39	229|0|5|17	
	750	–|–|0	–|–|0	–|–|0	–|98|0	–|99|0	–|99|0	–|74|0|93	–|79|0|50	–|74|0|22	
	1000	–|–|0	–|–|0	–|–|0	–|99|0	–|99|0	–|99|0	–|62|0|112	–|81|0|58	–|77|0|26	

BBMC: (50 cuts); **H:** PEO-derived clique cover and matching-based reduction (50 cuts)

References

1. Brélaz, D.: New methods to color the vertices of a graph. Communications of the ACM **22**(4), 251–256 (1979). https://doi.org/10.1145/359094.359101
2. Ekim, T., Farley, A., Proskurowski, A., Shalom, M.: Defensive domination in proper interval graphs. Discrete Applied Mathematics **331**, 59–69 (2023)
3. Ekim, T., Farley, A.M., Proskurowski, A.: The complexity of the defensive domination problem in special graph classes. Discrete Mathematics **343**(2), 111665 (2020)
4. Farley, A.M., Proskurowski, A.: Defensive domination. Congressus Numerantium **168**, 97 (2004)
5. Gavril, F.: Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph. SIAM Journal on Computing **1**(2), 180–187 (1972). https://doi.org/10.1137/0201013
6. Haynes, T.W., Hedetniemi, S., Slater, P.: Fundamentals of domination in graphs. CRC Press (2013)
7. Henning, M.A., Pandey, A., Tripathi, V.: More on the complexity of defensive domination in graphs. Discrete Applied Mathematics **362**, 167–179 (2025)

A Pre-processing Approach for Integrated 3D Bin Packing Problem in Pharmaceuticals

Seray Çakırgil[1][0000-0002-4068-5839], Elif Atar[2][0009-0009-2095-3099], and Eda Yücel[1][0000-0002-3448-1522]

[1] TOBB University of Economics and Technology, Söğütözü Ankara 06510, Türkiye
[2] Optipro Software, Söğütözü Ankara 06560, Türkiye
seraycakirgil@etu.edu.tr

Abstract. Pharmaceutical logistics requires a high degree of operational efficiency due to complex, multi-stage packaging hierarchies involving bundling, boxing, and palletization. This paper addresses the resulting multi-level three-dimensional bin-packing problem (3D-BPP), characterized by interdependent constraints and a large combinatorial search space. We propose an integrated preprocessing algorithm that evaluates spatial orientations and row-refinement heuristics to identify efficient packaging configurations across all packaging stages. The preprocessing algorithm generates and filters feasible bundle–box–pallet combinations, thereby transforming the original multi-stage 3D-BPP into a reduced assignment problem. Tested on an industrial dataset of 18 pharmaceutical products and 26 orders, the algorithm achieved a sub-second execution time and an average improvement in pallet utilization of 13% compared to current industrial practices. These results demonstrate the potential of the preprocessing approach to enhance volumetric efficiency and operational speed in pharmaceutical distribution.

Keywords: Pharmaceutical industry · Preprocessing algorithm · Bin packing · Pallet loading

1 Introduction

Pharmaceutical logistics requires high precision in packaging operations due to product sensitivity and the complexity of global supply chains. Unlike conventional distribution systems, pharmaceutical shipments follow a rigid multi-stage hierarchy in which primary products are grouped into *bundles*, packed into *cartons*, and loaded onto *pallets*. This hierarchical structure directly affects warehouse operations, since customer orders often involve multiple products and quantities smaller than a bundle, carton, or pallet, leading to additional picking, consolidation, and handling activities. Although the three-dimensional bin-packing problem (3D-BPP) provides the theoretical basis for modeling such systems [1, 2], pharmaceutical applications involve distinctive constraints, including strict orientation rules, regulatory SKU differentiation, structural stability

requirements, and increased handling effort for multi-product orders [3, 4]. The core challenge arises from the sequential dependency among packaging layers: even minor changes in bundle, carton, or pallet configurations can significantly affect pallet utilization and handling costs. Addressing this gap, this study proposes an integrated modeling framework that treats bundling, cartoning, and palletization as a unified combinatorial problem, formalizes bundling as an explicit hierarchical layer, and reduces the resulting search space through a preprocessing algorithm that captures interactions between product characteristics, order sizes, and logistics-related handling costs.

The study is organized as follows: Section 2 reviews related work on pharmaceutical supply chains and bin-packing optimization. Section 3 presents the problem formulation and the proposed multi-stage preprocessing algorithm. Section 4 reports the computational results, and Section 5 concludes the paper with future research directions.

2 Related Work

Optimization studies in pharmaceutical logistics span strategic supply chain design, three-dimensional bin packing (3D-BPP), and pallet loading problems (PLP). At the network level, existing research emphasizes regulatory coordination, resilience, and sustainability [5, 6]; however, these studies typically abstract away from operational packing constraints.

The 3D-BPP provides the theoretical foundation for modeling packing efficiency at the operational level [1]. In pharmaceutical contexts, these models are uniquely constrained by dosage-form orientation, structural stability, and SKU-level traceability [3]. Due to the NP-hardness of the problem, heuristic and metaheuristic approaches are commonly employed for industrial-scale instances where exact methods remain computationally impractical [7].

While pallet loading problems (PLP) have been extended to integrate upstream packaging variables [8], bundling remains a neglected decision layer. Most existing frameworks treat bundling, boxing, and palletization as independent or sequential processes, overlooking their critical interdependencies. This study contributes to the literature by proposing a unified optimization framework that explicitly incorporates bundling as a decision layer and jointly addresses all three stages within a pharmaceutical logistics context.

3 Pre-processing Algorithm

The pharmaceutical packing problem is modeled as a multi-stage three-dimensional bin-packing problem (3D-BPP). Let P be the set of products, where each product $p \in P$ is defined by dimensions (d_p^1, d_p^2, d_p^3). The logistics hierarchy consists of three packaging layers: bundles, cartons, and pallets. While bundling decisions may be product-specific, the sets of available carton types and pallet types, denoted by K and L, respectively, are shared across all products. Binary parameters b_p and k_p indicate whether product p requires bundling or cartonization.

Order size distributions are specified by pairs $\langle s_{op}, r_{op} \rangle$, where r_{op} denotes the percentage of total orders of product p that are of size s_{op}. Let O_p be the set of all possible order sizes for product p; the distribution satisfies $\sum_{o \in O_p} r_{op} = 100\%$.

A packing configuration is defined by the number of units placed along each spatial axis: $(\alpha_p^b, \beta_p^b, \gamma_p^b)$ for bundles, $(\alpha_p^k, \beta_p^k, \gamma_p^k)$ for boxes, and $(\alpha_p^l, \beta_p^l, \gamma_p^l)$ for pallets. These values determine the total number of product units at each packaging stage: n_{p_b}, n_{p_k}, and n_{p_l}.

Since the classical 3D-BPP is NP-hard [9, 10], the introduction of multiple hierarchical layers further enlarges the combinatorial search space. To address this challenge, we propose a preprocessing algorithm (Algorithm 1) that leverages the standardized packaging catalogs commonly used in pharmaceutical operations. The algorithm systematically enumerates feasible bundle, carton, and pallet combinations by evaluating six spatial orientations and applying row-refinement heuristics that adjust partially filled layers to improve space utilization. Each candidate configuration is validated against pallet footprint tolerances τ_{over} and τ_{under}, prioritizing feasibility and subsequently selecting configurations that maximize volumetric utilization while reducing material handling effort arising from order-size and bundle- or carton-size mismatches. For each product $p \in P$, the algorithm returns the best n_p candidate configurations according to each criterion separately. This preprocessing step transforms the original multi-stage 3D-BPP into a tractable assignment problem.

Algorithm 1 Integrated Packaging and Pallet Selection Pre-processing Algorithm

Require: Products P, Cartons K, Pallets L, tolerances $(\tau_{over}, \tau_{under})$, requirements $\{b_p, k_p\}_{p \in P}$, order size distributionss $\langle s_{op}, r_{op} \rangle_{o \in O_p, p \in P}$, number of candidate configurations n_p for product $p \in P$ and criterion.

Ensure: Candidate configuration list $\langle \mathcal{C}_p = \{b, k, l, \text{util}\} \rangle$ for all $p \in P$.

for all $p \in P$ **do**
  Define candidate sets: B_p (bundles), K_p (boxes), and L_p (pallets).
  Initialize feasible configuration set $\mathcal{F}_p \leftarrow \emptyset$
  for all $(b, k, l) \in B_p \times K_p \times L_p$ **do**
    Internal Packing ($b \to k$):
    Find max bundle count n_{bk} via 6 rotations and row refinements $r \in \{0, 1, 2\}$.
    External Loading ($k \to l$):
    Find max box count n_{kl} via 6 rotations and row refinements $r \in \{0, 1, 2\}$.
    Validate pallet footprint against tolerances $[\tau_{under}, \tau_{over}]$.
    if configuration is feasible **then**
      Compute volumetric utilization: $util = \text{Volume}(p \times n_{bk} \times n_{kl})/\text{Volume}(l)$
      Add configuration $(b, k, l, util)$ to $\mathcal{F}_p$: $\mathcal{F}_p \leftarrow \mathcal{F}_p \cup \{(b, k, l, util)\}$
  Rank configurations in $\mathcal{F}_p$ based on volumetric utilization and material handling effort, separately.
  Select top n_p configurations with highest utilization and set $\mathcal{C}_p \leftarrow$ selected configurations for each criterion.
return $\{\mathcal{C}_p\}_{p \in P}$

4 Computational Study

To evaluate the performance of the proposed pre-processing algorithm, a numerical study was conducted using an industrial dataset obtained from a pharmaceutical company. The test instance comprises 26 orders involving 18 distinct pharmaceutical products, 2 pallet types, and 15 carton types. Within this product catalog, 9 products require bundling (i.e., $b_p = 1$). The problem is bounded by a 1200 mm maximum pallet height, with footprint tolerances of $\tau_{over} = 5$ mm and $\tau_{under} = 100$ mm. At the packaging level, bundles are restricted to 30 units total and a maximum of 7 units per linear dimension. Internal carton stacking is capped at 6 units per dimension to ensure structural integrity. These thresholds define the feasibility of the spatial orientations and row-refinement heuristics in the pre-processing phase. The algorithm was implemented in C# within the Visual Studio environment. The implementation processed the entire dataset in less than one second on a standard workstation. Sample outputs comparing the best of algorithm-generated configurations with current industrial assignments are reported in Table 1.

Table 1. Comparison of current assignment and algorithm-based assignment

Product	Type	Bundling	Pallet	Box	Current assignment	Algorithm assignment
P1	Solid	No	L1	B1	1 pallet = 80 cartons = 8000 units	1 pallet = 90 cartons = 9000 units
P1	Solid	Yes	L1	B1	1 pallet = 36 cartons = 432 bundles = 10800 units	1 pallet = 90 cartons = 360 bundles = 10800 units
P2	Semi-solid	No	L1	B2	1 pallet = 24 cartons = 2400 units	1 pallet = 27 cartons = 5400 units
P3	Liquid	No	L2	B3	1 pallet = 24 cartons = 1200 units	1 pallet = 32 cartons = 1600 units

The results from the numerical study demonstrate a consistent improvement in volumetric efficiency across the product catalog. Except for the specific configuration of product P1, where bundling requirements constrained the achievable density, the number of units loaded per pallet increased in all evaluated cases. Across the 26 customer orders, the proposed approach achieved an average increase in pallet utilization of approximately 13%. This improvement not only maximizes the density of the logistics flow but also suggests a potential reduction in the total number of pallets required for distribution, thereby enhancing both cost-efficiency and sustainability within the pharmaceutical supply chain.

5 Conclusion

This study proposed a preprocessing algorithm framework for pharmaceutical packaging and palletization. By modeling bundling as an explicit decision layer, we developed an that captures interdependencies between packaging stages and

warehouse handling. Numerical results show an average 13% improvement in pallet utilization over current practices, with sub-second computation times suitable for large-scale operations. Future work will extend the framework using mixed-integer linear programming to incorporate machine availability, capacity constraints, and multi-objective trade-offs between utilization and operational cost.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. G. Wäscher, H. Haußner, and H. Schumann, An improved typology of cutting and packing problems, *European Journal of Operational Research*, 183(3), 1109–1130, 2007.
2. A. Lodi, S. Martello, and D. Vigo, Approximation algorithms for the oriented two-dimensional bin packing problem, *European Journal of Operational Research*, 141(2), 439–452, 2002.
3. C. Paquay, M. Schyns, and S. Limbourg, Three-dimensional bin packing and mixed load pallet building: Problems and mathematical formulations, *International Transactions in Operational Research*, 25(5), 1607–1644, 2018.
4. S. Çakırgil and E. Yücel, Optimizing blister packaging design for solid-form pharmaceuticals, *Computers & Industrial Engineering*, 111478, 2025.
5. N. Shah, Pharmaceutical supply chains: Key issues and strategies for optimisation, *Computers & Chemical Engineering*, 28(6–7), 929–941, 2004.
6. P. K. Ariningsih, C. A. Irawan, A. Paulraj, and J. Dai, A pharmaceutical distribution network considering supply cycles, waste, and inequity, *Computers & Operations Research*, 176, 106943, 2025.
7. S. Martello, D. Pisinger, and D. Vigo, Three-dimensional bin packing in the strong sense, *European Journal of Operational Research*, 144(2), 368–379, 2003.
8. E. Yücel, F. S. Salman, and G. Erdoğan, Optimizing two-dimensional vehicle loading and dispatching decisions in freight logistics, *European Journal of Operational Research*, 302(3), 954–969, 2022.
9. S. Martello, D. Pisinger, and D. Vigo, The three-dimensional bin packing problem, *Operations Research*, 48(2), 256–267, 2000.
10. W. Zhu, S. Chen, M. Dai, and J. Tao, Solving a 3D bin packing problem with stacking constraints, *Computers & Industrial Engineering*, 188, 109814, 2024.

A Flexible Weight Approach for the Logistics Performance Index Methodology

Işıl Can[1][0009−0001−2215−4516], Özgür Özpeynirci[1][0000−0002−3695−6587], and Sinem Tokcaer[1][/0000−0001−8842−3574]

Izmir University of Economics, Dept. of Logistics Management, Izmir, Turkey
{isil.can, ozgur.ozpeynirci, sinem.tokcaer}@ieu.edu.tr

Abstract. The Logistics Performance Index (LPI) is a benchmarking tool developed by the World Bank that uses a common weight set for all countries. However, this fixed weight approach may not reflect the diverse economic strengths and comparative advantages of different nations. In this study, we propose a mixed-integer linear programming (MILP) model that introduces weight flexibility within a reasonable range of the original LPI weights. We further refine the results using a lexicographic approach to minimize deviations. Finally, we apply k-means clustering to the optimal weights using centered log-ratio transformation to identify regional and economic patterns in logistics priorities.

Keywords: LPI · Weight Flexibility · Mixed Integer Programming · Lexicographic Optimization · k-means Clustering.

1 Introduction and Problem Definition

The Logistics Performance Index (LPI) is a benchmarking tool developed by the World Bank to provide countries with insights into their performance in international trade [1]. The index is composed of six fundamental indicators: efficiency of customs, quality of infrastructure, ease of arranging competitively priced international shipments, competence and quality of logistics services, ability to track and trace consignments, and timeliness of shipments [2].

The World Bank calculates the final LPI score by assigning fixed weights to each indicator, implying equal importance across all countries. However, economic environments and legislative features differ significantly between nations [3]. A country with a distinct economic structure may prioritize certain indicators over others based on its comparative advantage. This study emphasizes that a fixed weight methodology may disfavor some countries and proposes a mathematical framework that introduces controlled weight flexibility to better reveal their underlying logistics performance.

2 Literature Review and Common Weight Discussions

The multicriteria structure of LPI makes it suitable for Multi-Criteria Decision Making (MCDM) methods. While the World Bank uses common weights,

several researchers have used methods like the Best-Worst Method [4] or Data Envelopment Analysis (DEA) [5] to reassess rankings.

A critical discussion in the literature concerns the "common weight" approach. Using the same weight set for every participant can disadvantage those who excel in specific areas [6]. In the 2023 LPI report, the World Bank rounded results to the first decimal, which effectively transforms the ranking into a sorting problem. Our research aligns with the "flexible ranking" concept, suggesting that in cases of subjective or imprecise data, sorting is more appropriate than precise ranking [7].

3 Methodology

We develop a lexicographic Mixed-Integer Linear Programming (MILP) model to identify the best possible rank for each country by adjusting the criteria weights. We solve the model for each country separately and for different levels of weight flexibility. The model follows a hierarchical objective structure: it first optimizes the ranking of the focal country; subsequently, it minimizes the maximum and total deviations to ensure that the selected weight vectors remain as close as possible to the original LPI standards.

Let $K = \{1, 2, \ldots, n\}$ denote the set of countries and $J = \{1, 2, \ldots, q\}$ represent the set of criteria. The parameter u_{jk} is the score of country k for criterion j, while p_j corresponds to the original LPI weight. In the model, w_j represents the weight assigned to criterion j. The total utility of country k is denoted by ut_k, and $uint_k$ is an auxiliary integer variable used to ensure that the final utility score is rounded to one decimal place, consistent with World Bank reporting standards. The binary variable y_k equals 1 if the utility of country k is not higher than that of the focus country i, and 0 otherwise. This lexicographic approach is essential because multiple weight sets can often yield the same best rank (alternative optimal solutions). By using (PL^i_α), we ensure that all countries determine their weights using the same systematic logic: first achieving the best possible ranking, then minimizing the maximum deviation in any single weight (dev_{max}), and finally minimizing the total deviation from the original weight vector. This structured selection of weights ensures comparability across countries and, as a result, substantially enhances the interpretability of the subsequent clustering analysis.

The objective function (1) lexicographically minimizes the ranking of the country of focus, the maximum and the total deviation from original LPI weights. Constraint (2) ensures that the sum of weights of all criteria equals 1. Constraint set (3) introduces weight flexibility α to the model. Constraint set (4) defines the lower and upper limits of $uint_k$ used for rounding the scores to the first decimal. Constraint set (5) relates $uint_k$ to ut_k. Constraint set (6) determines the value of the binary variable y_k by comparing utility levels. Constraint sets (7) and (8) computes the deviation in the weights. Constraint set (9) computes the highest deviation. Constraint sets (10)-(15) define the decision variables.

Model $\left(PL_{\alpha}^{i}\right)$

$$\min \quad n - \sum_{k \in K, k \neq i} y_k + 0.1 \times dev_{max} + 0.001 \times \sum_{j=1}^{q} dev_j \tag{1}$$

Subject to:

$$\sum_{j=1}^{q} w_j = 1 \tag{2}$$

$$(1-\alpha)p_j \leq w_j \leq (1+\alpha)p_j, \qquad \forall j \tag{3}$$

$$-0.5 \leq uint_k - 10 \times \left(\sum_{j=1}^{q} u_{jk} w_j \right) \leq 0.5, \qquad \forall k \tag{4}$$

$$uint_k = ut_k, \qquad \forall k \tag{5}$$

$$ut_i \geq ut_k - M(1-y_k), \qquad \forall k \neq i \tag{6}$$

$$dev_j \geq w_j - p_j, \qquad \forall j \tag{7}$$

$$dev_j \geq p_j - q_j, \qquad \forall j \tag{8}$$

$$dev_{max} \geq dev_j, \qquad \forall j \tag{9}$$

$$dev_j \geq 0, \qquad \forall j \tag{10}$$

$$dev_{max} \geq 0, \tag{11}$$

$$y_k \in \{0,1\}, \qquad \forall k \tag{12}$$

$$uint_k \in \mathbb{Z}^{+}, \qquad \forall k \tag{13}$$

$$ut_k \geq 0, \qquad \forall k \tag{14}$$

$$w_j \geq 0, \qquad \forall j \tag{15}$$

After obtaining optimal weights $(w_{ji})_{\alpha}^{*}$ for each country, we apply k-means clustering to identify common behavioral patterns in weight selection. Since weight vectors are compositional data (i.e., they sum to one), before applying k-means clustering, we employ the centered log-ratio (CLR) transformation to avoid misleading results in distance-based measures. The optimal number of clusters is determined using Silhouette analysis, which yields $k = 8$.

4 Computational Results and Discussion

The model was tested for $\alpha \in \{0.05, 0.10, 0.15, 0.20\}$. Spearman's rank correlation tests indicate a high level of consistency with the original rankings, confirming that weight flexibility does not distort the overall ranking structure. Instead, flexibility primarily affects local rank positions, allowing countries to adjust indicator priorities while remaining within a dominance-consistent framework. Within this structure, three systematic patterns emerge:

- **Rank Improvements:** As α increases, growing number of countries improve their ranking. At $\alpha = 0.20$, 127 countries achieve higher rankings compared to the fixed weight model.
- **Developing vs. Developed Countries:** Majority of the countries showing substantial rank improvements are developing or least developed. In contrast, developed countries like Singapore and Finland tend to maintain their positions as they already benefit from the original weights.
- **Infrastructure Priority:** The clustering results indicate that less developed countries systematically shift to clusters characterized by higher average weights assigned to infrastructure relative to other LPI indicators.

Overall, the results indicate that weight flexibility enables meaningful differentiation among non-dominating countries, while preserving the relative positions of dominant performers. Minor deviations observed in the $\alpha = 0$ case are attributable to differences in rounding procedures rather than model inconsistency.

5 Conclusion

This study provides a mixed integer programming approach to introduce fairness into global logistics benchmarking. By allowing reasonable weight flexibility, we demonstrate that countries can better showcase their comparative advantages. For developing nations, infrastructure remains the most critical priority for improving logistics performance. Future work will focus on integrating longitudinal data to observe how optimal weight profiles evolve over time.

References

1. World Bank (2023). *Logistics Performance Index (LPI).* Available at: https://lpi.worldbank.org/ (Accessed: 01.10.2025).
2. Arvis, J. F., et al.: Connecting to Compete 2023: Trade Logistics in the Global Economy. World Bank (2023).
3. Hollweg, C. and Wong, M. (2009). *Measuring regulatory restrictions in logistics services.* Discussion Paper Series 2009–14, Economic Research Institute for ASEAN and East Asia (ERIA).
4. Rezaei, J., van Roekel, W. S., and Tavasszy, L. (2018). Measuring the relative importance of the logistics performance index indicators using Best Worst Method. *Transport Policy*, Vol. 68(2), pp. 158–169.
5. Martí, L., Puertas, R., and Garcia, L. (2014). The importance of the Logistics Performance Index in international trade. *Applied Economics*, Vol. 46(24), pp. 2982–2992.
6. Köksalan, M., Mousseau, V., Özpeynirci, Ö., and Özpeynirci, S. B. (2009). A new outranking-based approach for assigning alternatives to ordered classes. *Naval Research Logistics*, Vol. 56(1), pp. 74–85.
7. Köksalan, M., Büyükbaşaran, T., Özpeynirci, Ö., and Wallenius, J. (2010). A flexible approach to ranking with an application to MBA programs. *European Journal of Operational Research*, Vol. 201, pp. 470–476.

A Two-stage Chance-Constrained Programming Approach to Wavelength Dimensioning of WDM Networks

Haoyuan Xue[1], Merve Bodur[2], and Maryam Daryalal[3]

[1] Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, Ontario M5S 3G8, Canada, hxue@mie.utoronto.ca
[2] School of Mathematics and Maxwell Institute for Mathematical Sciences, University of Edinburgh, Edinburgh, UK, merve.bodur@ed.ac.uk
[3] Department of Decision Sciences, HEC Montréal, Montréal, Québec H3T 2A7, Canada, maryam.daryalal@hec.ca

Abstract. We study the core resource-allocation problem of wavelength dimensioning in WDM optical backbone networks. Departing from traditional deterministic traffic assumptions and uniform per-link provisioning, we propose a two-stage chance-constrained mixed-integer program. First-stage decisions choose a non-uniform wavelength structure on each link; service reliability is enforced through a joint chance constraint on the network's blocking rate. In recourse, we solve a routing and wavelength assignment (RWA) model to compute the minimum blocking, enabling a network-level assessment of robustness. To tackle the computational challenges, we propose a tailored branch-and-cut algorithm, embedded in a two-phase framework that (i) applies wavelength-aggregation cuts to rapidly tighten lower bounds and (ii) uses a targeted capacity-augmentation heuristic to construct high-quality feasible designs. On instances derived from real network topologies, our method achieves prescribed service levels with markedly less capacity: non-uniform designs save over 25% versus uniform provisioning, explicit RWA recourse saves over 10% versus shortest-path routing, and chance constraints reduce over-provisioning by more than 7.5% relative to robust single-scenario designs. Also, sensitivity analyses quantify various trade-offs.

Keywords: Wavelength dimensioning · Optical networks · Chance-constrained programming · Branch-and-cut algorithm.

1 Introduction

Telecommunication networks form the backbone of the global communication infrastructure, supporting an ever-expanding array of high-bandwidth applications from cloud computing to video streaming. As traffic volumes continue to surge, network operators face constant pressure to provision sufficient capacity while maintaining stringent quality-of-service (QoS) requirements. At the core of this infrastructure lies optical networking, where the dominant technology is wavelength-division multiplexing (WDM). In effect, WDM transforms

each fiber link into a bundle of parallel communication channels. This capability gives rise to a critical, high-stakes strategic planning challenge: *wavelength dimensioning*. This core resource-allocation problem involves determining the set of wavelengths to install on each fiber link so that the resulting network design meets service requirements at minimal cost. The decision entails a fundamental economic trade-off: the substantial capital expenditure of deploying network resources versus the need to ensure high QoS, typically measured by the probability of blocking (the chance that a connection request is rejected due to insufficient capacity). Striking the right balance is therefore essential for the efficient operation of large-scale optical backbones. The complexity of this decision is compounded by the inherent uncertainty in traffic demand; in particular, the volume and spatial distribution of connection requests are not fully predictable and can vary significantly over time and across the network, making cost-effective network design under service guarantees challenging.

Current approaches to wavelength dimensioning often rely on simplifying assumptions that limit practical relevance. Many models are deterministic, assuming traffic demands are known in advance, or they represent uncertainty with overly simplified processes. A common strategy is uniform dimensioning, where the same set of wavelengths is installed on every link—reducing the planning problem to choosing a single network-wide number of wavelengths. This can be highly inefficient since it fails to account for the heterogeneous traffic patterns observed in real networks. Finally, routing decisions for connection requests are frequently simplified by assuming that all connections follow predefined routes (e.g., shortest-path routing), which restricts operational flexibility and can lead to suboptimal network-wide performance. In this work, we address these critical shortcomings by proposing a two-stage chance-constrained mixed-integer optimization framework for wavelength dimensioning under traffic uncertainty.

2 Problem Formulation

The network is modelled as a directed graph $G = (\mathcal{V}, \mathcal{L})$ with available wavelengths Ω. Traffic uncertainty is represented by a finite set of scenarios Ξ. For each scenario $\xi \in \Xi$, let $N(\xi)$ be the total number of requests and $\bar{r}^{(s,d)}(\xi)$ be the demand for node pair of a realized request $(s,d) \in \mathcal{SD}(\xi)$. We define binary decision variables $U_{\ell,\omega}$ (1 if wavelength ω is installed on link ℓ) and z_ξ (1 if scenario ξ is discarded/failed). We also employ auxiliary variables W_ω for symmetry breaking. The problem is formulated as follows:

$$\min_{\substack{z\in\{0,1\}^{|\Xi|}, W\in\{0,1\}^{|\Omega|} \\ U\in\{0,1\}^{|\mathcal{L}|\times|\Omega|}}} \quad \sum_{\omega\in\Omega}\sum_{\ell\in\mathcal{L}} U_{\ell,\omega} \tag{1a}$$

$$\text{s.t.} \quad \sum_{\xi\in\Xi} z_\xi \le \lfloor \epsilon|\Xi| \rfloor \tag{1b}$$

$$z_\xi = 0 \implies Q(U,\xi) = 0 \quad \forall \xi \in \Xi \tag{1c}$$

$$\sum_{\ell\in\mathcal{L}} U_{\ell,\omega} \le |\mathcal{L}| \cdot W_\omega \qquad \forall \omega \in \Omega \tag{1d}$$

$$W_{\omega_1} \geq W_{\omega_2} \qquad \forall \omega_1, \omega_2 \in \Omega, \omega_1 > \omega_2 \qquad (1e)$$

The objective is to minimize total capacity. (1b) enforces the reliability requirement by limiting the number of discarded scenarios to an ϵ-fraction of the total. (1c) states that if a scenario is not discarded ($z_\xi = 0$), the recourse blocking $Q(U, \xi)$ must be zero. The recourse function evaluates feasibility by minimizing the excess blocked requests λ, beyond the $(1 - p)$ ratio:

$$Q(U, \xi) := \min \quad \lambda \qquad (2a)$$

$$\text{s.t.} \quad \sum_{(s,d) \in \mathcal{SD}(\xi)} x_{\ell,\omega}^{(s,d)} \leq U_{\ell,\omega} \qquad \forall \omega \in \Omega, \ell \in \mathcal{L} \qquad (2b)$$

$$\sum_{\omega \in \Omega} \sum_{\ell \in \delta^+(s)} x_{\ell,\omega}^{(s,d)} \leq \bar{r}^{(s,d)}(\xi) \qquad \forall (s,d) \in \mathcal{SD}(\xi) \qquad (2c)$$

$$\sum_{\omega \in \Omega} \sum_{(s,d) \in \mathcal{SD}(\xi)} \sum_{\ell \in \delta^+(s)} x_{\ell,\omega}^{(s,d)} + \lambda \geq \lceil (1-p) N(\xi) \rceil \qquad (2d)$$

$$\text{Flow conservation constraints on binary } x_{\ell,\omega}^{(s,d)} \qquad (2e)$$

This model seeks a Routing and Wavelength Assignment (RWA) that minimizes the blocking shortfall, described by variables $x_{\ell,\omega}^{(s,d)}$ representing the flow of request (s, d) on link ℓ using wavelength ω. (2b) and (2e) jointly ensure valid lightpath routing by enforcing capacity limits (wavelength conflict) and multi-commodity flow conservation (wavelength continuity). (2c) prevents routing flows in excess of specific demands, whereas (2d) quantifies λ if the aggregate flow fails to meet the $(1 - p)$ service-level threshold.

3 Solution Methodology

To solve the large-scale integer program, we develop a tailored branch-and-cut (B&C) algorithm. Given that the linear programming relaxation of the RWA problem is usually tight [1], in the solution process, we relax the integrality of the flow variables, which enables the efficient cut generation based on dual information. (In the solution evaluations for the experiments, however, we do consider binary flow variables.) During the B&C, the logical (implicitly defined) constraints (1c) are approximated via the following scenario-based Benders feasibility cuts [2], where the dual solution of (2) is denoted by $\hat{\pi}$:

$$\sum_{\ell \in \mathcal{L}, \omega \in \Omega} \hat{\pi}_{\ell,\omega}^{(2b)} U_{\ell,\omega} + (1 - z_\xi) \left[\sum_{(s,d) \in \mathcal{SD}(\xi)} \bar{r}^{(s,d)}(\xi) \hat{\pi}_{(s,d)}^{(2c)} + \hat{\pi}^{(2d)} \lceil (1-p) N(\xi) \rceil \right] \leq 0$$

To address the slow convergence (due to wavelength symmetry in the first-stage design variables, weak early lower bounds from sparse incumbents, and difficulty in finding feasible solutions), we embed the B&C in a *two-phase framework* (Fig. 1). The key idea is to decouple the two main computational bottlenecks:

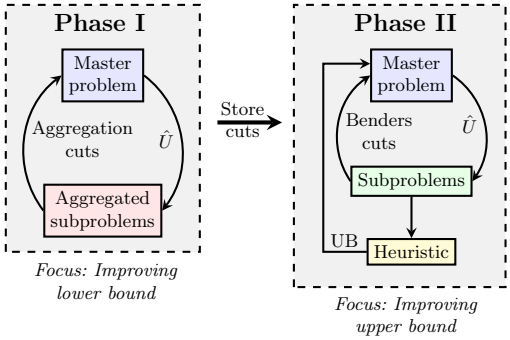

Fig. 1. The two-phase branch-and-cut solution framework.

Phase I *(lower-bound acceleration).* We separate *wavelength-aggregation cuts* from *aggregated* subproblems as in [1] that test feasibility against per-link total installed capacity $\sum_{\omega\in\Omega} U_{\ell,\omega}$, thereby reducing sensitivity to wavelength symmetry. These cuts quickly eliminate under-capacitated master solutions.

Phase II *(upper-bound acceleration).* We run the original B&C initialized with the Phase I cuts, separating Benders feasibility cuts from the original (disaggregated) subproblems whenever $Q(\hat{U},\xi) > 0$ for some non-discarded scenario ($\hat{z}_\xi = 0$). Moreover, we employ a *capacity-augmentation heuristic* that incrementally adds wavelengths on selected links (via auxiliary subproblems) until at least $|\Xi| - \lfloor\epsilon|\Xi|\rfloor$ scenarios satisfy (2d), yielding improved incumbents.

4 Numerical Results

We evaluate the proposed ideas on three real backbone topologies. The base setting uses $\epsilon = 0.05$ and $p = 0$; we test $|\Omega| \in \{50, 100, 200\}$ and measure reliability using 2000 test scenarios. Each scenario has $N \in \{30, 40, 50\}$ connection requests, built using a traffic-imbalance parameter $c \in [0, 1]$ that concentrates sources/destinations on traffic-center nodes. Across networks, non-uniform dimensioning achieves the target reliability while saving over 25% wavelength capacity versus uniform provisioning, explicit RWA recourse saves at least 10% versus shortest-path routing, and compared with robust single-scenario designs, chance-constrained designs reduce over-provisioning by more than 7.5%.

Disclosure Statement. The authors have no competing interests.

References

1. Daryalal, M. and Bodur, M. Stochastic RWA and lightpath rerouting in WDM networks. INFORMS Journal on Computing, **34**(5):2700–2719 (2022)
2. Luedtke, J. A branch-and-cut decomposition algorithm for solving chance-constrained mathematical programs with finite support. Mathematical Programming, **146**:219–244 (2014)

Exact and Heuristic Methods for Variants of the Secure Domination Problem

Burcu Çeştan[1,2] and Tınaz Ekim[1]

Boğaziçi University, Istanbul, Türkiye
Yeditepe University, Istanbul, Türkiye
burcu.cestan@yeditepe.edu.tr, tinaz.ekim@bogazici.edu.tr

Abstract. Secure domination is a graph protection approach in which guards respond to an attack by relocating while maintaining domination. We study four variants that combine different movement rules (single-guard vs. all-guards relocation) and capacity representations (binary vs. multiset placements): SD, m-SD, MSD, and m-MSD. We present integer programming (IP) formulations for all models and propose fast constructive heuristics for m-SD.

Keywords: Secure domination · Integer programming · Heuristics

1 Introduction

Secure domination, introduced by Cockayne et al. [3], extends classical domination by requiring that every attack on a vertex can be defended by relocating guards while maintaining domination. Let $G = (V, E)$ be a simple undirected graph, and let $N(v)$ and $N[v] = N(v) \cup \{v\}$ denote the open and closed neighborhoods of $v \in V$. A set $S \subseteq V$ is dominating if $N[v] \cap S \neq \emptyset$ for all $v \in V$, and it is a secure dominating set if for every $v \in V \setminus S$, there exists $u \in S \cap N(v)$ such that $(S \setminus \{u\}) \cup \{v\}$ remains dominating.

We study four variants obtained by combining two modeling choices: (i) movement rules (single-guard move vs. all-guards move) and (ii) capacity representations (binary vs. multiset placements): Secure Domination (SD), m-Secure Domination (m-SD), Multiset Secure Domination (MSD), and Multiset m-Secure Domination (m-MSD), with optimum values $\gamma_{1,1}(G)$, $\gamma_{m,1}(G)$, $\gamma_{1,m}(G)$, and $\gamma_{m,m}(G)$, respectively. In SD and MSD, a single guard moves to defend each attack, whereas in m-SD and m-MSD multiple guards may relocate simultaneously. In MSD and m-MSD, guard placements are multisets D with $D(v) \in \mathbb{Z}_{\geq 0}$.

We investigate how these relaxations affect the minimum number of required guards, with each variant yielding an optimum value no larger than $\gamma_{1,1}(G)$. Moreover, within the multiset setting, m-MSD improves upon or matches MSD. We prove that the two all-guards-move variants are equivalent, i.e., $\gamma_{m,1}(G) = \gamma_{m,m}(G)$.

To address these questions, we propose integer programming formulations for all four models and develop fast constructive heuristics for m-SD. We evaluate the performance of our methods on random Erdős–Rényi graphs and chordal graphs.

2 Solution Approaches

This section presents our solution approaches, including IP formulations for the four SD variants and constructive heuristics for the m-SD model. The IP formulations are inspired by the binary programming approach of Burger et al. [1].

Integer programming formulations. Let x_i denote the number of guards at vertex $i \in V$ (binary in SD and m-SD, integer in MSD and m-MSD). All models minimize $\sum_{i \in V} x_i$ subject to domination constraints $\sum_{j \in N[i]} x_j \geq 1$ for all $i \in V$. The models differ in how the defense step is represented: SD and MSD allow a single guard move, whereas m-SD and m-MSD allow multiple guards to relocate simultaneously.

As a representative example, consider m-SD. For an attack at $\ell \in V$, let $y^{\ell}_{kr} \in \{0, 1\}$ indicate whether a guard initially located at k relocates to $r \in N[k]$. We enforce: (i) each guard chooses exactly one destination, i.e., $\sum_{r \in N[k]} y^{\ell}_{kr} = x_k$ for all $\ell \in V$ and $k \in V$; (ii) there exists a feasible defense for every possible attack, i.e., $\sum_{k \in N[\ell]} y^{\ell}_{k\ell} \geq 1$ for all $\ell \in V$; and (iii) the post-defense configuration is dominating, i.e., $\sum_{r \in N[i]} \sum_{k \in N[r]} y^{\ell}_{kr} \geq 1$ for all $\ell \in V$ and $i \in V$. To the best of our knowledge, IP formulations for the m-SD, MSD, and m-MSD problems have not been explicitly presented in the literature.

Heuristic methods for m-SD. We propose four constructive heuristics for m-SD. The first three heuristics follow a two-phase structure. Phase 1 builds an initial guard placement using either a maximal independent set (MIS) or a greedy dominating set (DS), while Phase 2 applies a greedy support augmentation step to ensure domination is preserved after the simultaneous relocations allowed in m-SD.

The *Static MIS* heuristic computes an MIS using a degree-descending vertex ordering, whereas the *Dynamic MIS* heuristic constructs an MIS iteratively by repeatedly selecting a maximum-degree vertex in the current induced subgraph. The *Greedy DS* heuristic adapts the greedy set cover method of Chvátal [2] by selecting vertices that cover the largest number of uncovered nodes (via closed neighborhoods). Each of these core solutions is then augmented with additional support vertices.

Finally, the *Parent–Leaf* heuristic builds a BFS tree and repeatedly selects a parent vertex together with one of its leaf children, removes them (and their leaf descendants), and continues until the tree becomes empty. This method is inspired by a secure domination heuristic from [4].

3 Computational Experiments

We report experiments on Erdős–Rényi (ER) and chordal graphs. ER instances were generated using our own Python/NetworkX implementation, and chordal instances were generated using publicly available code [5]. All IP formulations were solved with Gurobi (Python) under a 1200s time limit. For each (n, p), five

Table 1. IP results on our graphs.

	n	p	SD				m-SD				MSD				m-MSD			
			OPT	Value	time	gap	OPT	Value	time	gap	OPT	Value	time	gap	OPT	Value	time	gap
ER Graphs	25	0,25	5/5	7	0	0	5/5	6	0	0	5/5	7	0	0	5/5	5,6	0	0
	25	0,5	5/5	4	0	0	5/5	3	1	0	5/5	4	0	0	5/5	3,4	1	0
	25	0,75	5/5	2,8	0	0	5/5	3	0	0	5/5	3	0	0	5/5	2,6	0	0
	30	0,25	5/5	7	0	0	5/5	6	1	0	5/5	7	0	0	5/5	5,8	1	0
	30	0,5	5/5	4	1	0	5/5	3	1	0	5/5	4	0	0	5/5	3,4	2	0
	30	0,75	5/5	3	0	0	5/5	3	1	0	5/5	3	0	0	5/5	2,6	1	0
	40	0,25	5/5	7,6	1	0	5/5	6	26	0	5/5	8	1	0	5/5	6,2	27	0
	40	0,5	5/5	4,4	3	0	5/5	4	7	0	5/5	4	2	0	5/5	4	10	0
	40	0,75	5/5	3	6	0	5/5	3	2	0	5/5	3	0	0	5/5	3	4	0
	50	0,25	5/5	8	6	0	5/5	6	306	0	5/5	8	3	0	4/5	6	53	0,14
	50	0,5	5/5	4,6	15	0	5/5	4	174	0	5/5	4,6	3	0	4/5	4	65	0,25
	50	0,75	5/5	3	6	0	5/5	3	8	0	5/5	3	1	0	5/5	3	16	0
	60	0,25	5/5	8,6	212	0	1/5	6	722	0,18	5/5	8,6	23	0	0/5	-	-	0,15
	60	0,5	5/5	4,8	87	0	5/5	4	289	0	5/5	4,8	7	0	4/5	4	494	0,25
	60	0,75	5/5	3	35	0	5/5	3	25	0	5/5	3	1	0	5/5	3	54	0
	70	0,25	1/5	8	280	0,13	0/5	-	-	0,18	5/5	8,6	70	0	0/5	-	-	0,2
	70	0,5	5/5	5	216	0	4/5	4	591	0,25	5/5	5	15	0	4/5	4	626	0,25
	70	0,75	5/5	3	63	0	5/5	3	67	0	5/5	3	9	0	5/5	3	128	0
	75	0,25	0/5	-	-	0,20	0/5	-	-	0,25	5/5	9	223	0	0/5	-	-	0,25
	75	0,5	5/5	5	326	0	2/5	4	784	0,25	5/5	5	19	0	2/5	4	942	0,25
	75	0,75	5/5	3	157	0	5/5	3	78	0	5/5	3	12	0	5/5	3	154	0
Chordal Graphs	100	0,2	5/5	13,8	10	0	5/5	9,4	18	0	5/5	13,4	0	0	5/5	9,4	4	0
	100	0,5	5/5	6,4	41	0	5/5	4,6	64	0	5/5	6,4	1	0	5/5	4,6	16	0
	100	0,8	5/5	2,8	20	0	5/5	2,8	91	0	5/5	2,8	1	0	5/5	2,8	32	0
	150	0,2	5/5	14,2	55	0	5/5	9,4	132	0	5/5	14	1	0	5/5	9,4	36	0
	150	0,5	4/5	6	289	0,14	5/5	4,4	622	0	5/5	6,2	3	0	5/5	4,4	113	0
	150	0,8	5/5	3,6	311	0	3/5	3,3	746	0,75	5/5	3,6	4	0	5/5	3,2	260	0
	200	0,2	5/5	15,4	460	0	4/5	10	687	0,08	5/5	15,4	3	0	5/5	10,4	269	0
	200	0,5	1/5	6	1153	0,22	0/5	-	-	-	5/5	6,4	8	0	3/5	5,3	1015	-
	200	0,8	1/5	4	478	0,41	0/5	-	-	-	5/5	4,2	12	0	0/5	-	-	-

instances were generated; Table 1 reports averages. The column "OPT" indicates how many of the five runs were solved to optimality. For instances solved to optimality, we report average objective value and runtime; for time-limit runs, we report the average optimality gap. Experiments were run on a MacBook Pro (Apple M4, 16 GB RAM).

As expected, higher ER density leads to smaller solutions. The all-guards-move variants (m-SD and m-MSD) are considerably harder and often reach the time limit. Within the time limit, optimal solutions are obtained for larger chordal graphs, whereas for ER graphs optimality is typically reached only on smaller instances. On instances solved to optimality, m-SD and m-MSD yield identical objective values, consistent with $\gamma_{m,1}(G) = \gamma_{m,m}(G)$. Moreover, all-guards-move variants produce smaller solutions than their single-move counterparts; SD and MSD differ on chordal instances, while in our ER tests they often coincide.

Table 2 highlights a clear quality–speed trade-off: Greedy DS yields the smallest solutions, while Parent–Leaf is the fastest on large ER graphs. Relative to

Table 2. m-SD heuristic results on large ER graphs (avg. guards and runtime).

n	p	Static MIS		Dynamic MIS		Greedy DS		Parent–Leaf	
		g	t	g	t	g	t	g	t
2500	.25	24.2	.208	20.6	.333	16.0	.725	52.0	.004
2500	.50	11.6	.119	9.8	.295	8.0	.647	22.8	.004
2500	.75	6.6	.078	6.6	.224	5.0	.349	12.4	.003
5000	.25	28.0	1.016	22.2	1.507	17.8	3.804	54.8	.008
5000	.50	13.0	.554	10.6	1.249	9.0	3.057	25.2	.008
5000	.75	8.0	.396	6.2	.880	5.8	1.638	13.6	.428
7500	.25	28.2	2.313	24.6	3.671	19.0	8.830	57.6	.376
7500	.50	13.8	1.332	11.0	2.969	9.0	6.402	26.0	.374
7500	.75	7.6	.854	6.2	3.270	6.0	5.448	14.0	.022
10000	.25	29.2	111.07	25.2	6.418	20.2	68.43	60.8	.857
10000	.50	13.8	2.366	11.2	7.321	10.0	20.28	28.4	3.717
10000	.75	8.6	1.748	6.6	6.894	6.0	21.81	14.4	14.71

optimal m-SD IP solutions on smaller instances, Greedy DS attains the lowest average gaps, suggesting the strongest overall performance.

4 Conclusion

We studied four secure domination variants combining different movement rules and capacity representations. We proved that the two all-guards-move variants are equivalent. Our results show that all-guards-move models quickly become computationally challenging, whereas MSD scales better across the tested instances. Finally, m-SD heuristics provide efficient alternatives on large ER graphs, with Greedy DS yielding the best solution quality and Parent–Leaf the fastest runtimes.

Acknowledgments. Supported by the European Commission's Horizon Europe Research and Innovation programme through the Marie Skłodowska-Curie Actions Staff Exchanges (MSCA-SE) under Grant Agreement No. 101182819 (COVER).

Disclosure of Interests. The authors declare that they have no competing interests.

References

1. Burger, A., de Villiers, A., van Vuuren, J.: A binary programming approach towards achieving effective graph protection. In: Proceedings of the 2013 ORSSA Annual Conference. pp. 19–30 (2013)
2. Chvátal, V.: A greedy heuristic for the set-covering problem. Mathematics of Operations Research **4**(3), 233–235 (1979)
3. Cockayne, E.J., Grobler, P.J.P., Gründlingh, W.R., Munganga, J., Van Vuuren, J.H.: Protection of a graph. Utilitas Mathematica **67**, 19–32 (2005)
4. de Villiers, M.: Edge Criticality in Secure Graph Domination. Ph.D. thesis, University of Johannesburg (2014)
5. Şeker, O.: Chordal: Chordal graph generation code. `https://github.com/oylumseker/Chordal` (2025), accessed: 2025-12-05

Interval Optimization under Bounded Capacity Constraints

Arthur Mazeyrat[1], Ammar Oulamara[2], Tifenn Rault[1], and Ameur Soukhal[1]

[1] University of Tours, LIFAT EA 6300, 64 avenue Jean Portalis, 37200 Tours, France
arthur.mazeyrat@etu.univ-tours.fr
{tifenn.rault, ameur.soukhal}@univ-tours.fr
[2] University of Lorraine, LORIA - UMR 7503, Campus Scientifique - BP 239, 54506 Vandœuvre-les-Nancy, France
ammar.oulamara@loria.fr

Abstract. In this paper, we address the following question: given a combinatorial optimization problem (P) defined on interval graphs, where one selects a subset of intervals, can (P) be solved in polynomial time under an additional maximum clique constraint $\omega(O) \leq m$, for any fixed $m \in \mathbb{N}$? We show that if (P) admits a dynamic programming algorithm satisfying a few mild structural conditions, then this question can be answered positively. Beyond the theoretical result, the proposed approach also leads to algorithms that outperform state-of-the-art solvers on representative problem instances.

Keywords: Interval Graphs, Dynamic Programming, Parameterized Complexity

1 Introduction

Interval-based optimization is a central topic in operational research [3], covering essential application fields such as energy management, telecommunications and logistics. Many of these problems admit polynomial time solutions under infinite capacity assumptions. However, introducing finite resource limits (m) typically induces a considerable increase in computational complexity, often resulting in NP-hard variants. A prime example is the Electric Vehicle Charging Scheduling Problem (EVCSP), where the complexity for a fixed number of identical chargers has remained an unresolved challenge [4].

This paper presents a recursive expansion procedure that converts standard interval-based dynamic programming algorithms (DP) into variants supporting maximum clique constraints. Traditionally, bounded capacities have been addressed through problem-specific model redesigns, often leading to a combinatorial explosion of the state space. Our approach avoids this limitation by relying on an alternative state-space representation in which resource occupation is treated as a structural property. We exploit the topology of temporal overlaps to evaluate feasibility instead of relying on explicit item enumeration, we maintain optimality while ensuring a manageable state space.

2 Problem description

Let us consider a binary optimization problem (P) defined by:

- A set $\mathcal{J}$ consisting of n intervals defined by $J_i = [r_i, d_i)$.

- A set $\mathcal{F}$ of feasible subsets $O \subseteq \mathcal{J}$.
- A cost function $C : \mathcal{F} \to \mathbb{R}$ that assign to each feasible subset of $\mathcal{F}$ a cost.

Without loss of generality, (P) is a minimization problem:

$$\begin{aligned} (P) \quad \min \quad & C(O) \\ \text{s.t.} \quad & O \subseteq \mathcal{F} \end{aligned} \tag{1}$$

We now work under the hypothesis that our problem can be solved using a dynamic programming algorithm (DP) of the following fashion:

We want to find the optimal subset O of sets inside that lies in an interval containing every jobs. For any sub-interval $[a, b)$, the DP computes the best subset from sub-intervals $[a', b')$ with $a' \geq a$ and $b' \leq b$. The DP can also combines up to ℓ sub-intervals at any step.

Under a certain structural condition of these ℓ sub-intervals (which is true for a wide range of DP), let us define the extended problem as $P_{\omega(m)}$:

$$\begin{aligned} (P_{\omega(m)}) \quad \min \quad & C(O) \\ \text{s.t.} \quad & O \subseteq \mathcal{F} \\ & \omega(O) \leq m \end{aligned} \tag{2}$$

Theorem 1. *For any base problem P solvable in $O(\text{poly}(n))$, its version $P_{\omega(m)}$ is solvable in time $O(n^{2m\ell} \cdot \text{poly}(n))$, which is polynomial for any fixed m.*

3 Applications

We present three application cases to further illustrate the broad scope of our results:

1. The added constraint $\omega(O) \leq m$ can be generalized to a time-dependent constraint of the form $\omega_t(O) \leq f(t)$, where $f(t)$ is bounded by a constant, i.e., $f(t) \leq m$. This extension allows our results to be applied to industrial settings in which machines may be temporarily unavailable or resource capacities vary over time.
2. The proposed approach is also compatible with multidimensional resource constraints. In particular, it applies to the Resource-Constrained Interval Scheduling Problem (RCISP) [1, 2], where each job consumes an amount r_i^k of each resource $k \in \{1, \ldots, m\}$, and where, at any time, the total consumption of each resource may not exceed a given capacity M^k. The key reason this extension is possible is that the extended $\text{DP}_{\omega(m)}$ has the appropriate structure, allowing the theorem to be applied iteratively to additional families of clique constraints, one for each resource.
3. The extended $\text{DP}_{\omega(m)}$ remains computationally efficient for certain NP-hard problems that admit polynomial-time solutions in the absence of clique constraints. Many industrial applications fall into this category, where the number of machines or tracks is very limited, while the number of jobs is large. Assuming an unlimited number of machines typically trivializes the problem, as selecting all jobs becomes a feasible solution.
   A representative example is the Electric Vehicle Charging Scheduling Problem (EVCSP). In the single-machine case with parking constraints, the objective is to schedule preemptive charging on a machine with m tracks so as to maximize the number of accepted vehicles, each represented by an interval and an energy demand, under the requirement that any accepted vehicle remains parked throughout its interval. The complexity of this

variant had remained open [4]. We resolve this question by proving that the problem is NP-hard in general, while admitting a polynomial-time algorithm for any fixed m.
Beyond the theoretical guarantees of polynomiality, the effectiveness of our algorithmic methodology has been validated through numerical experiments on industrial-size instances. We tested the model on configuration involving 300 jobs ($n = 300$) and a capacity of 10 resources ($m = 10$), focusing on so-called agreeable intervals, a setting in which the EVCSP remains NP-hard.
The performance tests highlight several key results: our DP approach proves to be significantly faster than commercial MILP solvers such as Gurobi, while maintaining remarkable efficiency under high resource saturation, a context in which generic solvers presents a dramatic increase in computational time. By consistently delivering exact solutions within computational short time, our approach demonstrates strong industrial viability, showing that exploiting the topology of overlaps enables the effective treatment of complex instances and makes the approach particularly well suited for real-time operational management systems.

4 Conclusion

We proposed a structured approach extending interval-based dynamic programming algorithms to handle bounded clique constraints. Under mild structural assumptions, we showed that any problem solvable in polynomial time without capacity limits remains polynomial-time solvable for any fixed clique bound m. This result unifies and explains the tractability of several interval scheduling and resource allocation problems under limited resources. Applications to multi-resource and time-dependent settings further illustrate the versatility and practical relevance of the approach.

References

1. Enrico Angelelli, Nicola Bianchessi, and Carlo Filippi. Optimal interval scheduling with a resource constraint. *Computers & Operations Research*, 51:268–281, 2014.
2. Antoon W.J. Kolen, Jan Karel Lenstra, Christos H. Papadimitriou, and Frits C.R. Spieksma. Interval scheduling: A survey. *Naval Research Logistics*, 54(5):530–543, 2007.
3. Leo G Kroon, Marc Salomon, and Luk N Van Wassenhove. Exact and approximation algorithms for the operational fixed interval scheduling problem. *European Journal of Operational Research*, 82(1):190–205, 1995.
4. I. Zaidi, A. Oulamara, L. Idoumghar, and M. Basset. Maximizing the number of satisfied charging demands of electric vehicles on identical chargers. *Omega*, 127:103–106, 2024.

Valid Equalities and Dimension of the Optimal Classification Tree Problem

Zacharie Ales[1,2][0000−0003−4602−2638]

[1] UMA, ENSTA Paris, Institut Polytechnique de Paris, 91120 Palaiseau, France
[2] CEDRIC, CNAM, 75003 Paris, France
zacharie.ales@ensta.fr

Abstract. The optimal classification tree problem consists in learning a decision tree of bounded depth that minimizes misclassification errors on a given dataset. While mixed-integer linear and quadratic programming formulations have recently enabled the exact resolution of small to medium-sized instances, their scalability remains limited by the size of the data. Understanding the polyhedral structure of these formulations could improve their computational performance.
In this work, we study an adaptation of the Fortet linearization of a quadratic formulation for optimal univariate classification trees. We focus on characterizing the dimension of the convex hull of its feasible integer solutions. Our analysis relies on the identification of families of valid equalities derived from the routing structure of data points through the tree.
We prove that, under a mild and natural assumption on the dataset, any hyperplane containing the convex hull can be expressed as a linear combination of these valid equalities. We then exhibit a subset of associated normal vectors that is linearly independent and generates the full space of valid hyperplanes. This allows us to derive an explicit formula for the dimension of the convex hull as a function of the number of variables and the identified equalities.

Keywords: optimal classification trees · valid equalities · polytope dimension

1 Problem Definition

Let $\mathcal{D} = \{(x_i, y_i)\}_{i\in\mathcal{I}}$ be a dataset composed of pairs of feature vectors $x_i \in [0,1]^{|\mathcal{J}|}$, each associated with a class label $y_i \in \mathcal{K}$.

A classification tree is a binary tree $T = (\mathcal{N} \cup \mathcal{L}, A)$ whose nodes are partitioned into two sets: the internal nodes $\mathcal{N}$ and the leaves $\mathcal{L}$. Each internal node $t \in \mathcal{N}$ is associated with a split $a_t x < b_t$, where $a_t \in \mathbb{R}^{|\mathcal{J}|}$ and $b_t \in [0,1]$ are the split coefficients, and $x \in [0,1]^{|\mathcal{J}|}$ is a feature vector. Each leaf $\ell \in \mathcal{L}$ is associated with a class $k_\ell \in \mathcal{K}$.

Such a tree predicts the class y_i of a data point x_i by routing it along a path from the root to one of the leaves. The data point is sent to the left child of any internal node $t \in \mathcal{N}$ it reaches if $a_t x_i < b_t$. If the condition is not satisfied, the

data point is sent to the right child of t. The class k_ℓ associated with the leaf ℓ reached by the data point is the predicted class. The data point is therefore correctly classified if $y_i = k_\ell$.

The optimal classification tree problem consists in determining a tree of a given maximal depth which minimizes the number of misclassifications on a dataset $\mathcal{D}$.

In this work, we restrict our attention to *univariate* trees, in which each vector a_t is required to be a unit vector. In other words, a single feature is used for each split.

2 State of the Art

This problem is generally solved using greedy heuristics [5, 7, 6]. In 2017, two initial MILP formulations were introduced [4] for the univariate and multivariate versions of the problem, respectively. A flow-based formulation, restricted to univariate trees and binary datasets (i.e., $\mathcal{D} \subset \{0,1\}^{|\mathcal{J}|}$), was subsequently considered [1]. [3] presents a quadratic formulation of the problem as well as a generalization of the flow formulation to non-binary data and multivariate trees.

The optimal solution of these formulations is, however, quickly limited by the size of the dataset $\mathcal{D}$ under consideration. As a result, [2] introduces an approach based on an initial data partitioning that is iteratively refined, allowing the optimal solution of larger datasets.

3 Contributions

We consider an adaptation (QF') of the Fortet linearization of the quadratic formulation proposed in [3]. We characterize the dimension of the convex hull of the integer solutions of the formulation.

We first identify two families of valid equalities that we then generalize. Finally, identifying the maximum number of independent hyperplanes associated with these equalities allows us to determine the dimension of the convex hull of the solutions of the problem.

We now succintly present the main results which enable to characterize the dimension of the polytope.

3.1 Valid equalities

Let S be the set of feasible solutions of formulation (QF') for a dataset $\mathcal{D}$ and a given maximal depth d. For each sample $i \in \mathcal{I}$ and each leaf $\ell \in \mathcal{L}$, this formulation consider a binary variable $z_{i\ell}$ equals to 1 if and only if sample i reaches leaf ℓ. Let $\mathcal{L}(t) \subseteq \mathcal{L}$ be the set of leaves under a node $t \in \mathcal{N} \cup \mathcal{L}$. Let $I_t(T) \subseteq \mathcal{I}$ be the set of samples which reach node $t \in \mathcal{N} \cup \mathcal{L}$ in solution $T \in S$.

The valid equalities are obtained from subsets of vectors $\{V^t\}_{t \in \mathcal{N} \cup \mathcal{L}}$ defined as follows.

Definition 1. *For a given $t \in \mathcal{N} \cup \mathcal{L}$, let*

$$V^t \triangleq \{v \in \mathbb{R}^{|\mathcal{I}|} \mid \sum_{i \in \mathcal{I}_t(T)} v_i = 0 \ \forall T \in S\}.$$

This definition leads to the following equalities.

Proposition 1 *For any $t \in \mathcal{N} \cup \mathcal{L}$, $v \in V^t$ if and only if*

$$\sum_{i \in \mathcal{I}} \left(v_i \sum_{\ell \in \mathcal{L}(t)} z_{i\ell} \right) = 0 \quad \forall T \in S \tag{1}$$

3.2 Dimension of the polytope

We first prove that all valid hyperplanes for $conv(S)$ can be obtained by linear combinations of Equalities (1) provided that the dataset $\mathcal{D}$ is not *evenly spaced* on all features.

Definition 2. *Dataset $\mathcal{D}$ is said to be evenly spaced on dimension $j \in \mathcal{J}$ if the difference between two consecutive distinct values on feature j is always the same (i.e., if $p_1 < p_2 < ... < p_K$ are the distinct values $\{x_{i,j}\}_{i \in \mathcal{I}}$ sorted in increasing order, then $p_{k+1} - p_k = p_{k+2} - p_{k+1}$ for all $i \in \{1, 2, ..., K-2\}$).*

Let Ω be the set of normal vectors of the equations (1) associated to all $t \in \mathcal{N} \cup \mathcal{L}$ and all $v \in V^t$.

Theorem 1 *If $\mathcal{D}$ is not evenly spaced on all features, any hyperplane including $conv(S)$ can be obtained by a linear combination of vectors in Ω.*

We then identify a subset $\Omega^s \subseteq \Omega$ which generates Ω and prove the following theorem.

Theorem 2 *If $\mathcal{D}$ is not evenly spaced on all features, all the vectors in Ω^s are linearly independents.*

From the two theorems, we deduce that the dimension of $conv(S)$ is equal to the number of variables in the formulation minus $|\Omega^s|$.

Note that since the set Ω^s is obtained from the vectors $\{V^t\}_{t \in \mathcal{N} \cup \mathcal{L}}$, the number of equalities depends on the dataset $\mathcal{D}$ considered.

4 Conclusion

We investigate the polyhedral structure of a formulation of the optimal univariate classification tree problem. By identifying families of valid equalities induced by the routing of samples in the tree, we characterized all hyperplanes containing

the convex hull of feasible integer solutions under a mild assumption on the dataset.

This analysis led to a characterization of the dimension of the associated polytope. These results provide new structural insights into exact formulations of optimal classification trees and constitute a first step towards the systematic use of polyhedral information to strengthen such models. Future work will focus on exploiting these equalities experimentally and extending the analysis to broader classes of datasets and MILP formulations.

References

1. Aghaei, S., Gomez, A., Vayanos, P.: Learning Optimal Classification Trees: Strong Max-Flow Formulations (May 2020). https://doi.org/10.48550/arXiv.2002.09142, http://arxiv.org/abs/2002.09142, arXiv:2002.09142 [cs, math, stat]
2. Ales, Z., Huré, V., Lambert, A.: Clustering data for the Optimal Classication Tree Problem (May 2024), https://hal.science/hal-04589656, working paper or preprint
3. Alès, Z., Huré, V., Lambert, A.: New optimization models for optimal classification trees. Computers & Operations Research **164**, 106515 (2024)
4. Bertsimas, D., Dunn, J.: Optimal classification trees. Machine Learning **106**(7), 1039–1082 (Jul 2017). https://doi.org/10.1007/s10994-017-5633-9, https://doi.org/10.1007/s10994-017-5633-9
5. Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A.: Classification and Regression Trees. Taylor & Francis (Jan 1984)
6. Carreira-Perpinan, M.A., Tavallali, P.: Alternating optimization of decision trees, with application to learning sparse oblique trees. In: Advances in Neural Information Processing Systems. vol. 31. Curran Associates, Inc. (2018), https://proceedings.neurips.cc/paper/2018/hash/185c29dc24325934ee377cfda20e414c-Abstract.html
7. Quinlan, J.R.: C4.5: programs for machine learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. (1993)

AI-Guided Multi-Objective Optimization for Green Centralized Collaborative Transportation

Elham Jelodari Mamaghani

Institute of Sustainable Business and Organisations Sciences and Humanities
Confluence Research Center-UCLY, ESDES, Lyon, France
ejelodarimamaghani@univ-catholyon.fr

1 Introduction

Logistics service providers face growing pressure to improve efficiency while meeting high service-level expectations and increasingly stringent environmental requirements, particularly in road freight transportation [1]. While large carriers benefit from scale and extensive networks, small LTL carriers face fleet and capacity constraints that limit competitiveness and sustainability [2].Collaborative logistics has therefore been widely studied as a means to improve vehicle utilization and reduce empty mileage through resource and request sharing, with horizontal collaboration among carriers operating [3, 4]. Recent studies emphasize that collaboration opportunities arise from capacity complementarities across carriers' transportation requests and can be identified using advanced decision-support tools [5]. However, the benefits of collaboration critically depend on its operationalization. It can be centralized or decentralized, but decentralized negotiation-based approach often fail to realize system-wide synergies due to limited information sharing [6]. Consequently, centralized collaborative transportation planning has gained attention, enabling joint request assignment across carriers to optimize overall network performance. Such approaches typically distinguish between reserved and exchangeable requests, both based on the pickup and delivery points with time window, allowing reassignment to a carrier to improve efficiency [3, 7]. These centralized planning approaches naturally give rise to complex decision trade-offs. From a modeling perspective, such problems are inherently multi-objective, since service-related criteria such as workload balance are essential alongside cost minimization for practical implementation [8]. Environmental sustainability has become also a critical dimension given the significant contribution of road freight transportation to greenhouse gas emissions and increasingly strict climate regulations [9]. Nevertheless, existing studies often focus on isolated cost and emissions [10] without focusing fairness, overlooking realistic centralized multi carrier characteristics such as heterogeneous fleets [11]. Moreover, despite significant advances in metaheuristic approaches for collaborative transportation, the integration of machine learning into multi objective models remains limited [12]. This paper addresses these challenges by proposing a green centralized collaborative transportation planning framework for the LTL pickup and delivery problem and makes four main contributions. First, we develop

a multi objective optimization model that integrates centralized multi carrier coordination, reserved and exchangeable requests, heterogeneous fleets, and time window constraints. The model jointly minimizes total transportation costs, CO_2 emissions, and maximum route duration to enhance operational robustness. Second, we develop a hybrid NSGA-II–based algorithm that improves convergence and Pareto-front diversity for large size problems through KNN-based offspring adaptation and adaptive Gaussian mutation. Third, we introduce a machine learning driven parameter tuning procedure based on Random Forest Regression (RFR) to ensure algorithmic robustness and scalability across diverse instance sizes. Finally, we conduct computational comparisons with multi objective simulated annealing (MOSA) and NSGA-II, which confirm the superior efficiency of the proposed AI-driven algorithm across multiple multi objective performance metrics.

2 Problem definition

We consider a set of collaborating carriers, each operating a heterogeneous fleet from a dedicated depot, serving pickup and delivery points defined by locations, demands, travel attributes, and time windows. Requests, each defined by a pickup-delivery pair, are classified as reserved or exchangeable, with only the latter eligible for reassignment by a central planner to improve system wide performance. The system is modeled on a directed graph, with decision variables defining request assignment, vehicle selection, and routing sequences. The primary formulation is nonlinear due to the minimization of the maximum route duration and the distance- and load-dependent emissions function. Hence, it is reformulated as a multi objective mixed integer linear program (MILP) through auxiliary variables and linear constraints. The objectives are to minimize (i) total transportation costs, including routing and vehicle usage costs, (ii) total CO_2 emissions, and (iii) the maximum route duration to improve fairness and operational robustness. Feasibility is ensured through constraints enforcing assignment exclusivity, pickup–delivery precedence, flow conservation, vehicle capacity limits, time window compliance, consistency between routing and assignment decisions, and linearization constraints. Since the problem is NP-hard, efficient heuristic or metaheuristic algorithms are required [9].

3 Solution approach

To solve large size instances of the green CCTP, we propose AGKN-NSGA-II, a learning-enhanced evolutionary algorithm extending NSGA-II. In this algorithm, solution density information obtained through K-nearest neighbors (KNN) learning is used to dynamically adjust crossover and mutation probabilities, promoting intensification in promising regions and exploration in the areas of the Pareto front. In addition, an adaptive Gaussian mutation operator dynamically adjusts mutation strength to reduce premature convergence and enhance robustness across instances.

4 Computational experiments

To ensure robust algorithmic performance, the parameters of the proposed NSGA-II–DALNS algorithm are tuned using a data driven approach based on supervised machine learning rather than traditional design-of-experiments or Taguchi methods [9]. A Random Forest Regression enables parameter exploration and reliable calibration under stochastic algorithm behavior through repeated evaluations.

The performance of AGKN-NSGA-II is evaluated through computational experiments on benchmark instances. For larger instances (300 instances), AGKN-NSGA-II is benchmarked against standard NSGA-II and a MOSA using established convergence and diversity indicators (MID, NoS, Diversity, and Time). Table 1 presents the results for a representative subset of instances selected to illustrate the comparative performance of the algorithms across the metrics considered. Each instance is denoted as M-R-RR-ER, indicating carriers, total requests, reserved requests, and exchangeable requests, respectively. To enable fair comparison across instances of different sizes, performance is evaluated using the relative percentage deviation (RPD), which normalizes results and removes the effect of problem size. As can be observed, AGKN-NSGA-II consistently outperforms the benchmark methods, demonstrating superior solution quality, Pareto-front diversity, and scalability.

Table 1. Evaluation of AGKN-NSGA-II (AL1) against NSGA-II (AL2) and MOSA (AL3) in terms of RPD across multiple performance metrics

AL1 vs. AL2								
	Time		NoS		Diversity		MID	
Instance	AL1	AL2	AL1	AL2	AL1	AL2	AL1	AL2
5-340-137-203	16.68	38.86	16.60	31.58	10.76	34.08	14.98	51.16
5-341-300-41	15.64	42.40	13.19	27.65	30.23	49.17	8.85	30.23
5-341-132-209	10.11	36.02	17.79	25.54	18.45	45.64	28.71	52.15
5-341-306-35	11.85	28.64	11.07	30.06	22.39	27.50	12.73	31.93
AL1 vs. AL3								
	NoS		Diversity		MID		Time	
Instance	AL1	AL3	AL1	AL3	AL1	AL3	AL1	AL3
5-340-137-203	7.24	17.30	18.23	23.24	12.07	22.63	25.70	37.10
5-341-300-41	12.45	22.01	12.37	22.38	18.55	41.05	19.85	32.10
5-341-132-209	11.03	33.42	18.44	26.13	13.67	34.47	25.89	37.17
5-341-306-35	9.40	35.51	17.70	30.98	21.11	26.58	19.55	35.52

5 Conclusion

This study investigated a green centralized collaborative transportation planning problem for multi carrier pickup and delivery networks, with LTL conditions. A

multi objective MILP model was developed to jointly minimize transportation costs, CO_2 emissions, and the maximum route duration, capturing key economic, environmental, and operational trade-offs. To address the computational complexity of the problem, an AI-guided hybrid evolutionary algorithm was proposed, combining NSGA-II with machine learning–based adaptive mechanisms. Computational experiments on benchmark and large size instances demonstrated that the proposed approach outperforms classical metaheuristics in terms of convergence and diversity. Future research could extend this work by incorporating dynamic and stochastic elements, such as real time demand arrivals, travel time uncertainty, and disruptions, to better reflect operational conditions in e-commerce logistics.

References

1. Hu, Q., Zhang, Z., Lim, A.: Transportation service procurement problem with transit time. Transportation Research Part B: Methodological 86, 19–36 (2016).
2. Najafi, M., Zolfagharinia, H.: No longer in the dark: utilizing imperfect advance load information for single-truck operators. Transportation Science **56**(6), 1573–1597 (2022).
3. Padmanabhan, B., Huynh, N., Ferrell, W., Badyal, V.: Potential benefits of carrier collaboration in the vehicle routing problem with pickup and delivery. Transportation Letters **14**(3), 258–273 (2022).
4. Korkmaz, S.G., Soysal, M., Sel, C., Dündar, H.: Modeling a location and routing problem for mobile lockers in last-mile delivery with horizontal collaboration. Networks **86**(1), 3–29 (2025).
5. Luan, J., Daina, N., Reinau, K.H., Sivakumar, A., Polak, J.W.: A data-based opportunity identification engine for collaborative freight logistics based on a trailer capacity graph. Expert Systems with Applications **210**, 118494 (2022).
6. Los, J., Schulte, F., Gansterer, M., Hartl, R.F., Spaan, M.T., Negenborn, R.R.: Large-scale collaborative vehicle routing. Annals of Operations Research **350**(1), 201–223 (2025).
7. Santos, M.J., Martins, S., Amorim, P., Almada-Lobo, B.: A green lateral collaborative problem under different transportation strategies and profit allocation methods. Journal of Cleaner Production **288**, 125678 (2021).
8. Bertazzi, L., Golden, B., Wang, X.: Min–max vs. min–sum vehicle routing: A worst-case analysis. European Journal of Operational Research **240**(2), 372–381 (2015).
9. Mamaghani, E.J., Battaia, O.: Toward sustainable and customer-centric reverse logistics: Machine learning-enhanced multi-objective optimization. Computers & Industrial Engineering, 111622 (2025).
10. Vaziri, S., Etebari, F., Vahdani, B.: Development and optimization of a horizontal carrier collaboration vehicle routing model with multi-commodity request allocation. Journal of Cleaner Production **224**, 492–505 (2019).
11. Stoia, S., Laganà, D., Ohlmann, J.W.: Dynamic pickup-and-delivery for collaborative platforms with time-dependent travel and crowdshipping. European Journal of Operational Research **322**(1), 70–84 (2025).
12. Okdinawati, L., Simatupang, T.M., Sunitiyoso, Y.: Multi-agent reinforcement learning for value co-creation of collaborative transportation management (CTM). International Journal of Information Systems and Supply Chain Management **10**(3), 84–95 (2017).

Olive oil supply chain planning under uncertainty

Bilge Bilgen[1][0000-0002-2361-0413]

[1] Dokuz Eylul University, Department of Industrial Engineering, Buca, Izmir, Türkiye

bilge.bilgen@deu.edu.tr

Abstract. Decision-making under uncertainty is a major challenge in the agri-food industry. Agri-food supply chains (AFSC) are growing increasingly complex, making them more vulnerable to various uncertainty and risks. AFSCs are subject to various uncertainties and risks that can lead to serious consequences if not managed properly. To improve performance, it is also essential to manage risks in AFSCs effectively and efficiently. This paper focuses on design and planning of resilient olive-oil supply chain network under risk and uncertainty. Severe weather events have the potential to present a substantial risk to the agricultural economy. So, the risk of weather extreme conditions needs to be considered. To deal with stochastic yield, and olive loss due to climate conditions, we develop risk-neutral and risk-averse two-stage stochastic programming (SP) formulations. To quantify the associated risks, and to overcome the variability of the random outcomes, conditional value at risk (CVaR), will be incorporated in the framework of two-stage SP. This paper aims to provide a stochastic optimization framework to manage various uncertainties for more efficient olive-oil supply chain management.

Keywords: Olive oil supply chain, two-stage stochastic programming.

1 Introduction

Operations research has been applied to agriculture, leading to a rich collection of literature. However, the uncertain elements of AFSCs have only received academic focus in the last two decades. Several review papers have been published on AFSCs (see e.g [1-3]. Agri-food products are subject to perishability and seasonality. Also, annual production variations make it challenging to manage both quality and quantity of outputs. These issues threaten AFSCs, making their distribution networks particularly vulnerable to various risks. There is a growing body of literature devoted to the development of optimization models in the agri-food industry. Although there is a rich and still growing body of literature on AFSC, very little attention has been paid to olive oil supply chains. Our study addresses olive-oil supply chain management problem under uncertainty, which has been gain less attentions by the previous literature.

2. Problem Definition

In this paper, we mainly focus on the harvest, production and transportation planning problem in the olive oil supply chain network. The olive oil supply chain includes internal olive groves, external olive suppliers, olive oil mills, harvesting teams and trucks. Harvesting season is a limited period of the year, generally 3 months (from September to December). Due to this limitation, harvesting operations need to be planned carefully. The start of the harvesting season depends on the ripeness of the olive. The harvesting teams are assigned to only one olive grove on a daily basis for harvesting. The olive groves are harvested by only one harvesting team. Moreover, the harvesting time window is a constraint for harvesting operations in the olive groves. The internal olive groves are harvested to satisfy demand. If there are insufficient olives in the olive groves, the olive oil producer may purchase olives from external olive suppliers. Harvested olives in the olive groves and purchased olives from external olive suppliers are transferred to the olive oil mill in order to produce olive oil. Trucks, which transfer olives from the olive groves to olive-oil mills, are outsourced with an agreed price per kilometer. Olive oil mills have maximum and minimum capacity. After producing olive oil, the olive oil is classified based on the quality level. The quality level is determined according to acidity levels. The olive oil is produced as bulk and is sold to wholesaler company. Decisions about harvesting operations are directly linked to two main items. One of them is the quantity of olive oil. The percentage of oil concentration, which is a decisive on the quantity of olive oil, is increased by the ripeness of olive. The olive oil producer takes a strategic decision to select an olive oil mill option for construction. A deterministic MILP model is developed to optimize the harvesting, production, and transportation of olives, taking into account the specific characteristics of the olive oil industry. The model includes operational, tactical and strategic level decision variables. Olive loss, and yield parameters are uncertain.

2. Risk-neutral and risk-averse model formulations

In this section, we propose a risk-neutral and risk-averse model formulations. Two-stage stochastic programming is a common approach used to account for uncertainties. It is assumed that all the stochastic variables are described in a common and whole probability space $(\Omega, \mathcal{P})$. Ω is a set of scenarios and $\mathcal{P}$ is a probability set. $\omega \in \Omega$ refers to the realizations of the scenarios in the second stage. The two-stage stochastic linear program [4] is formulated as below:

$$\begin{aligned} \min_{x \in R^n} \; & c^T x + E_\omega \left[\min_{y(\omega) \in R^m} q(\omega)^T y(\omega) \right] \\ & s.t. \quad Ax = b \\ & W(\omega)^T y = h(\omega)^T - \; T(\omega)^T \qquad \forall \omega \in \Omega \\ & x \geq 0 \\ & y(\omega) \geq 0 \end{aligned}$$

With respect to this mathematical formulation, x is presented as the first stage decision and $y(\omega)$ corresponds the second stage decision. E represent the expected value of the second stage. The aim of the two-stage stochastic program is to minimize the total cost, which includes investment cost in the first stage and the expected operational cost in the second stage. Decisions (x) in the first stage are determined at the beginning of the period as "here and now" and deterministic decisions. After the realization of uncertainty, decisions (y) in the second stage are decided at the end of the period. These are considered as "wait and see" or recourse decisions.

In the two-stage stochastic programming, the decision maker is risk-neutral. In decision-making problems under uncertainty, where outcomes are subject to variability, adopting a risk-averse approach provides more robust solutions compared to a risk-neutral approach [5]. In this work, we incorporate Conditional value at risk (CVaR) into the model formulation. Since the introduction of CVaR, it has been recognized as an effective risk measure in assessing risk in the literature, due to its coherence in assessing risk. Practically, CVaR focuses on penalizing negative deviations from an efficiency target.

3. Conclusion and future research

This paper proposes a two-stage stochastic programming model for the olive oil supply chain planning problem. The proposed model maximizes total profit. Constraints on the system are related to harvesting, production and transportation. The extended model considers the uncertainties on the yield and the olive loss. Future research implications include: extending the model to a multi-objective framework, incorporating disruptions and resilience strategies, and solving the model using matheuristic approaches.

Acknowledgments. The author's research was funded by The Scientific and Technological Research Council of Türkiye (TUBITAK 2219 program).

Disclosure of Interests. The author has no competing interests to declarethat are relevant to the content of this article

References

1. Behzadi, G., OSullivan, M.J., Olsen, T.L., Zhang, A., Agribusiness supply chain risk management: A review of quantitative decision models. Omega 79, 21–42 (2018)
2. Borodin, V., Bourtembourg, J., Hnaien, F., Labadie, N., 2016. "Handling uncertainty in agricultural supply chain management: A state of the art". European Journal of Operational Research. 254 (2), 348–359.
3. Taşkıner, T.; Bilgen, B. 2021. "Optimization Models for Harvest and Production Planning in Agri-Food Supply Chain: A Systematic Review". Logistics 5, 52.
4. J. R. Birge and F. Louveaux, Multistage Stochastic Programs. 1997. Introduction to Stochastic Programming, 233–252.
5. Noyan, N. 2012. "Risk-averse two-stage stochastic programming with an application to disaster Management", Computers & Operations Research, 39 (3) pp. 541-559.

A Simple Backtracking Algorithm for Solving the Minimum Spanning Tree Problem with Disjunctive Conflict Constraints

Murat Umut İzer[1][0000−0002−4200−5382], Temel Öncan[2][0000−0003−0595−0673], and İ. Kuban Altınel[1][0000−0003−1119−8844]

[1] Department of Industrial Engineering, Boğaziçi University, 34342, Bebek, İstanbul, Türkiye
[2] Department of Industrial Engineering, Galatasaray University, 34357, Ortaköy, İstanbul, Türkiye

Abstract. The Minimum Spanning Tree problem with Disjunctive Conflict Constraints is an NP-hard extension of the classical Minimum Spanning Tree problem in which certain pairs of edges cannot coexist in the spanning tree. Motivated by the combinatorial structure of the problem, we propose a simple exact backtracking algorithm that recursively dives for an optimal conflict-free spanning tree. The method utilizes depth-first search, integrating infeasibility tests, a simple lower bound, and probing. Compared to existing algorithms, our proposed method offers a transparent, easily implementable alternative.

Keywords: Minimum spanning tree problem · Conflict constraints · Backtracking algorithm.

1 Introduction

The Minimum Spanning Tree (MST) problem is a fundamental problem in combinatorial optimization. Given a connected graph, the objective is to find a spanning tree of minimum total weight. Numerous real-world extensions have been proposed, including degree-, bottleneck-, and capacity-constrained variants. Among these, the *Minimum Spanning Tree problem with Disjunctive Conflict Constraints (MSTC)* introduces disjunctive restrictions, where specific pairs of edges cannot be selected together. Such conflicts arise naturally in communication networks with interference, in transportation planning with mutually exclusive routes, and in design networks with overlapping resources.

MSTC was introduced in [1], where the authors show that the problem is strongly NP-hard. Later, [2] developed several Binary Integer Linear Programming formulations, Lagrangean relaxation schemes, heuristics, and infeasibility tests. Exact solution approaches based on branch-and-cut were proposed in [3], introducing valid inequalities and comparing different separation strategies.

Conflict constraints have been studied in a wide range of combinatorial optimization problems (e.g., [4, 5]). We observe that they typically lead to increased

computational complexity. This study contributes to the existing literature by presenting a new exact method that recursively explores feasible solutions using depth-first search (DFS), infeasibility tests, simple lower bounds, and probing.

2 Diving for an Optimal Solution Recursively

Let $G = (V(G), E(G))$ be a connected graph with edge weights $c_e \geq 0$. The *conflict graph* $C = (V(C), E(C))$ is defined on the edge set of G, where each vertex of C represents an edge of G, and $\{e, f\} \in E(C)$ denotes that edges e and f are in conflict, i.e., mutually incompatible. MSTC can thus be formulated as:

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

$$\text{s.t.} \quad \sum_{e \in E(G)} x_e = |V(G)| - 1 \tag{2}$$

$$\sum_{e \in \gamma_G(U)} x_e \leq |U| - 1 \qquad U \subset V(G), |U| \geq 2 \tag{3}$$

$$x_e + x_f \leq 1 \qquad \{e, f\} \in E(C) \tag{4}$$

$$x_e = 0, 1 \qquad e \in E(G). \tag{5}$$

Feasible edge subsets correspond to conflict-free spanning trees. The problem is NP-hard, which is expected since the edge set of any feasible spanning tree forms a stable set in the conflict graph C. We denote by I, F, and X the sets of included, free, and excluded edges, respectively. During recursive exploration, the following conditions allow early pruning:

1. The subgraph $G[I \cup F]$ (induced by edge subset $I \cup F$) is disconnected.
2. Two leaf edges of $G[I \cup F]$ are in conflict.
3. The number of remaining free edges $|F|$ is less than $|V(G)| - 1 - |I|$.

Conditions (1) and (3) are concerned with graph feasibility, while condition (2) deals with the structural limitations of the conflict graph [6]. Instead of leaf edges, a test based on bridges of the graph induced by free and included edges can be considered. Any bridge must belong to every spanning tree. Bridges are identified during recursion using DFS labels. If two bridge edges are in conflict, the partial solution is infeasible and can be pruned.

At each recursion level, a basic combinatorial lower bound is computed as

$$LB = \sum_{e \in I} c_e + \sum_{e \in M} c_e, \tag{6}$$

where M denotes the $(|V(G)| - 1 - |I|)$ cheapest edges in F that are conflict-free with I. If $LB \geq \overline{z}$, where $\overline{z}$ is the current upper bound, the branch is pruned.

Diving is an exact recursive backtracking algorithm for MSTC. It starts from an empty partial tree, incrementally adding edges while enforcing conflict constraints. Recursion stops when an optimal solution is found, an infeasibility is detected, or time limit is exceeded. The procedure is outlined as Algorithm 1.

Algorithm 1 Recursive Diving (Rec-D)

Input: $G = \big(V(G), E(G)\big)$, $C = \big(V(C), E(C)\big)$, (I, X, F), best upper bound $\overline{z}$;
Output: An optimal solution, if one exists, or an infeasibility decision;
Run infeasibility test on the current subproblem;
if if subproblem is feasible (i.e., I forms a spanning tree) **and** $c(I) < \overline{z}$ **then**
  Update incumbent and $\overline{z}$; **return**
else if subproblem is infeasible **then**
  Pruning by infeasibility; **return**
end if
Apply probing to reduce the size of the subproblem based on a threshold;
Order free edges $F = \{e_1, \ldots, e_p\}$ (optional)
for $i = 1$ to p **do**
  if $I \cup \{e_i\}$ is acyclic **then**
    Compute a lower bound LB for $I \cup \{e_i\}$;
    if $LB < \overline{z}$ **then**
      Set $I \leftarrow I \cup \{e_i\}, X \leftarrow X \cup (N_C(e_i) \cup \{e_1, \ldots, e_{i-1}\})$;
      Set $F \leftarrow F \setminus (N_C(e_i) \cup \{e_1, \ldots, e_i\})$;
      Call Diving with updated (I, X, F);
      Undo changes (backtrack);
    end if
  end if
end for
end

Rec-D algorithm starts with a partition of edge subsets (I, X, F). Infeasibility test may find a feasible solution. If it has a better objective value than the best upper bound, then the incumbent solution is updated. The current subproblem is pruned if it is infeasible. When the test is indecisive, subproblem size might be reduced. Probing is applied with respect to a threshold value to further reduce that size. Free candidate edges are recursively included one at a time, excluding conflicting and previously considered edges, while ensuring that the tree is acyclic. A simple lower bound is used to prune suboptimal branches. After the recursive call, the edge subsets (I, X, F) are restored to their previous states for backtracking. This DFS continues until optimality or infeasibility is established.

Besides Rec-D, we implement an iterative diving algorithm (BB-D) using a loop to prevent stack overflow. This avoids deep recursion and allows finer control over time limits and backtracking. When there is no candidate free edge in BB-D, the last included edge is excluded. It may be beneficial in terms of more efficient backtracking and a reduced size of the solution space.

3 Computational Results

We test Rec-D and BB-D on our benchmark MSTC instances. All tests were conducted on a server running Microsoft Windows Server 2016 with Intel Xeon E5-2650 @ 2.20 GHz CPU and 288 GB RAM. The algorithms were implemented in C++ 20 and compiled with Microsoft Visual Studio 2022. Gurobi version 10.0.3 was used as the reference solver with default settings. A time limit of 1800 seconds was imposed on all runs. The performance of all algorithms is given in

Table 1 in terms of the number of optimal and feasible solutions, average number of explored and active nodes, average best upper bound, average CPU time of all instances (given in parenthesis) and instances commonly solved optimally.

Table 1. Performance of BB algorithms.

Test Statistic	Algorithms		
	GUROBI	Rec-D	BB-D
Opt. Sols.	219	246	257
Feas. Sols.	14	7	10
Explored Nodes	30404	56701221	18456655
Active Nodes	13893	16343373	15640383
Best UB	3885.62	4294.49	4359.94
CPU (secs.)	90.08 (690.45)	16.52 (510.95)	2.49 (447.19)

214 of the 335 instances are solved optimally in common by all algorithms. Results show that both diving algorithms can solve a significant percentage of the instances optimally in reasonable CPU time. Specifically, the search space is greatly reduced by using infeasibility tests and probing. Due to its effective backtracking mechanism and explicit state management, BB-D consistently achieves faster runtimes and finds the highest number of optimal and feasible solutions.

4 Conclusions

We present a straightforward exact backtracking algorithm for MSTC which combines recursive diving with DFS, utilizing simple lower bounds, infeasibility tests and probing. Despite its simplicity, the method seems promising. Our findings show that efficient exact solution methods can be achieved by analyzing the structural properties of the underlying graph and conflict graph. Future research directions include stronger bounds and integration with other branching rules.

References

1. Darmann, A., Pferschy, U., Schauer, J., Woeginger, G. J.: Paths, trees and matchings under disjunctive constraints. Discrete Applied Mathematics **159**(16), 1726-1735 (2011).
2. Zhang, R., Kabadi, S. N., Punnen, A. P.: The minimum spanning tree problem with conflict constraints and its variations. Discrete Optimization **8**(2), 191-205 (2011).
3. Carrabs, F., Cerulli, R., Pentangelo, R., Raiconi, A.: Minimum spanning tree with conflicting edge pairs: a branch-and-cut approach. Annals of Operations Research, **298**(1), 65-78 (2021).
4. Altınel, İ. K., Aras, N., Şuvak, Z., Taşkın, Z. C.: Minimum cost noncrossing flow problem on layered networks. Discrete Applied Mathematics, **261**, 2–21 (2019).
5. Pamuk, B., Öncan, T., Altınel, İ. K.: An efficient branch-and-bound algorithm for the one-to-many shortest path problem with additional disjunctive conflict constraints. European Journal of Operational Research, **324**(2), 398–413 (2025).
6. İzer, M.U.: *Minimum Spanning Tree Problems with Conflict Constraints.* PhD thesis, Boğaziçi University, Istanbul, Türkiye (2025)

Explain or Argue? A Question-Driven Decision Layer for Transparent Scheduling

Miguel A. Salido, Adriana Giret, and Christian Perez

Universitat Politècnica de València
msalido@dsic.upv.es, agiret@dsic.upv.es, cripeber@upv.es

Abstract. Nowadays, scheduling systems face a question in real-life applications: given a user query, should the system provide an *explanation* to support understanding, or an *argument* to support defensible acceptability under objections and preferences? Using the Job Shop Scheduling Problem (JSSP) as a reference setting, in this paper, we propose a compact question-driven decision layer that routes user questions to three response modes: Explain, Argue, or Hybrid (explain-then-argue). This layer is based on established concepts from explainability and computational argumentation, plus the disjunctive-graph view of JSSP. By means of a JSSP example, we demonstrate how "why accept" queries map to preference-aware dialectical justification and "why/why-not" queries map to contrastive or certificate-style explanations. We conclude with implications for evaluation and deployment.

Keywords: Scheduling · JSSP · Explainability · Argumentation · Interaction · Decision Support

1 Introduction

Optimal schedules are not necessarily *accepted* schedules. In practice, planners balance objective performance (e.g., makespan), stability w.r.t. prior plans, due-date service, and organizational policies. As a result, transparency requests vary: some questions ask for *understanding* (explainability), while others demand *defensible acceptability under objections* (argumentation). A frequent failure mode is providing the wrong type of transparency for the question at hand. In this paper, we propose a lightweight *decision layer* that, given a user question, selects whether the system should Explain, Argue, or follow a Hybrid flow. This separates response-mode selection from the underlying scheduling solver and makes interactive transparency more predictable.

2 Background: Scheduling Problems

Typically, scheduling problems are categorized using $\alpha|\beta|\gamma$ [1] (machine environment, constraints/features, objective). JSSP is a canonical NP-hard problem (see [2]) with machines $\mathcal{M} = \{1, \ldots, m\}$ and jobs $\mathcal{J} = \{1, \ldots, n\}$, where each job $j \in \mathcal{J}$ consists of operations $(o_{j,1}, \ldots, o_{j,k_j})$ with machine assignments and processing times. Intra-job precedence, single-capacity machines (pairwise disjunctions), and non-preemption are the traditional constraints, which usually minimize makespan $C_{\max}$.

The *disjunctive graph* [3] is a crucial representation: each machine conflict is a disjunctive edge whose orientation chooses an ordering, and precedence edges are fixed. Because many user questions are local ("why o before o' on machine m?") and have global implications (critical paths, bottlenecks), this representation is helpful for transparency.

According to our operational definition, an explanation is an artifact E that explains to a human why a solution or decision s is suggested for an instance I [4]. Explainability in scheduling and optimization can be categorized by *what* is explained: model-centric accounts whose evaluation is still difficult [6], solver-based "why-not" accounts rooted in CP/CSP explanation traditions [5], and solution-centric accounts. Additionally, interactive textual explanations for scheduling decisions are explored in scheduling-specific work [7].

In abstract argumentation, arguments and attacks are captured by an argumentation framework and acceptability is determined by semantics (e.g., grounded, preferred, stable) [8]. In recent works, argumentation has been applied to scheduling-related optimization, demonstrating its ability to adapt to change and offer stakeholder-facing explanations [9]. Argumentative XAI examines the ways in which argumentation facilitates interactive objection handling and dialectical explanations [10]. In this context, *persuasion* is understood in the dialectical sense: altering acceptance through reasons that can be examined by the public and clearly addressing objections, as opposed to using rhetorical devices.

3 Explain or argue? A proposed question-driven decision layer

Given a user query q and context such as policies or a prior schedule π_0, the system selects a response mode: Explain, Argue, Hybrid). Intuitively, Explain targets understanding, Argue targets defensible acceptability (i.e., rational persuasion) under objections, and Hybrid supports common interactions where an initial explanation triggers objections that require argumentation. This mode activates when user objections arise during an explanation, allowing the system to shift from informative to persuasive responses seamlessly.

We use three criteria that can be checked from the query and context.

Criteria 1: *User intent.* If the query requests causal/structural understanding ("why is o before o'?", "what causes $C_{\max}$?"), select Explain. If it asks for acceptance despite competing considerations, for instance "why should I accept this change?", select Argue.

Criteria 2: *Explicit objections or value conflict.* If the question introduces competing criteria, policies, or preferences such as stability vs makespan, due-date vs disruption, fairness rules, select Argue; otherwise, prefer Explain.

Criteria 3: *"Why-not" feasibility/bound queries.* If the question asks why a constraint/threshold cannot be met, for instance "why not enforce $o' \prec o$ while keeping $C_{\max} \leq \tau$?", select Explain with a certificate-style or core-based account. If it is posed as a challenge to acceptance, select Hybrid.

Table 1 summarizes a compact routing map for common scheduling question families.

Table 1. Question-driven routing: when to explain vs argue (or use a hybrid flow).

Question type	Typical phrasing	Mode and expected output
Why	Why is o before o' on machine m? Why is $C_{\max}$ high?	EXPLAIN: structural/contrastive reasons (disjunctive choice, critical block, sensitivity).
Why-not	Why not enforce $o' \prec o$ and keep $C_{\max} \leq \tau$?	EXPLAIN: certificate/core-based "why-not" (constraint core, bound evidence).
What-if	What if we prioritize job J? What changes if we forbid overtime?	EXPLAIN: counterfactual deltas; optionally propose minimal changes.
Why accept	I see the gain, but why should I accept the change? Policy says X.	ARGUE: pro/con arguments, preference order, dialectical justification (rational persuasion).
Mixed challenge	Explain the decision *and* address my objection about stability/fairness.	HYBRID: explain first, then argue using preferences/policies and objections.

3.1 A minimal decision rule (implementable)

To make the routing operational, we can express the selection as a simple rule set over the question q and conversational context c (e.g., prior plan π_0, current policy mode, and whether the user has already challenged a recommendation). Let $obj(q, c)$ denote the presence of an explicit objection or value conflict (e.g., stability, fairness, policy compliance), and let $bound(q)$ denote "why-not" feasibility/threshold queries (e.g., "keep $C_{\max} \leq \tau$"). Then a routing rule is:

$$\text{MODE(q,c)} = \begin{cases} \text{ARGUE} & \text{if obj(q,c)}, \\ \text{EXPLAIN} & \text{if } \neg\text{obj(q,c)} \wedge (\text{bound(q)} \vee \text{why(q)} \vee \text{whatif(q)}), \\ \text{HYBRID} & \text{if why(q)} \wedge \text{obj(q,c)} \text{ (typically after a follow-up).} \end{cases}$$

The key point is that routing depends less on the underlying optimization model and more on whether the user is requesting *understanding* or *acceptance under dispute*.

In interactive schedulers, $obj(q, c)$ is often signaled by contrastive connectives and normative language ("but", "however", "should we accept", "policy says", "this is unfair/too disruptive"), or by references to prior commitments ("the previous plan", "do not change the sequence", "stability"). Conversely, $bound(q)$ is signaled by thresholds and feasibility constraints ("keep $C_{\max} \leq \tau$", "meet due date d", "no overtime") and is best answered with certificate-style evidence.

Once the decision layer selects a response mode, the system should follow a simple and predictable template. The goal is to keep answers brief, faithful to the scheduling structure, and easy to extend in follow-up interactions.

- EXPLAIN: state the decision; provide the key structural reason (precedence/conflict); add a small contrast (local swap/alternative) or solver evidence (certificate/core) for "why-not" queries.
- ARGUE: state the claim to be accepted; present the main supporting and opposing arguments; resolve using the explicit preference/policy order (e.g., urgent vs normal mode).

– Hybrid: deliver a concise Explain answer first, then switch to Argue when an objection or value conflict is raised.

We give a minimal JSSP illustration of the routing mechanism; the extended example with figures is provided in Appendix A.

4 Conclusions and open challenges

We proposed a question-driven decision layer to make the decision to Explain, Argue, or adopt a Hybrid explain-then-argue approach for scheduling systems. The JSSP example demonstrates how the same decision may require contrastive/certificate-style or preference-aware argumentative explanations, depending on the user query. Future work includes integrating certified reasoning into dialogue systems, selecting meaningful alternatives without combinatorial blow-up, handling dynamic rescheduling, and building benchmarks to assess both quality and transparency. We believe this lightweight layer can be smoothly integrated into current scheduling systems, improving transparency without sacrificing efficiency.

Acknowledgements

This paper has been partially supported by Grant PID2024-158315NB-I00, funded by MICIU/AEI/10.13039/501100011033 and the European Union.

References

1. Graham, R.L., Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G.: Optimization and approximation in deterministic sequencing and scheduling: a survey. Annals of Discrete Mathematics **5**, 287–326 (1979)
2. Pinedo, M.L.: Scheduling: Theory, Algorithms, and Systems. Springer (2022)
3. Błażewicz, J., Pesch, E., Sterna, M.: The disjunctive graph machine representation of the job shop scheduling problem. European Journal of Operational Research **127**(2), 317–331 (2000)
4. Rudin, C., Chen, C., Chen, Z., Huang, H., Semenova, L., Zhong, C.: Interpretable machine learning: Fundamental principles and 10 grand challenges. Statistics Surveys **16**, 1–85 (2022)
5. Dev Gupta, S., Genc, B., O'Sullivan, B.: Explanation in constraint satisfaction: A survey. In: Proc. IJCAI-21, pp. 4400–4407 (2021)
6. Nauta, M., Trienes, J., Pathak, S., Nguyen, E., Peters, M., Schmitt, Y., Schlötterer, J., van Keulen, M., Seifert, C.: From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable AI. ACM Computing Surveys **55**(13s), Article 295 (2023)
7. Powell, C., Riccardi, A.: Generating textual explanations for scheduling systems leveraging the reasoning capabilities of large language models. Journal of Intelligent Information Systems **63**(4), 1287–1337 (2025)
8. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence **77**(2), 321–357 (1995)
9. Leigh, J., Letsios, D., Mella, A., Machetti, L., Toni, F.: Argumentation for Explainable Workforce Optimisation (with Appendix). CoRR abs/2508.15118 (2025)
10. Čyras, K., Rago, A., Albini, E., Baroni, P., Toni, F.: Argumentative XAI: A survey. In: Proc. IJCAI 2021, pp. 4392–4399 (2021).

A Appendix: Decision Routing Example in JSSP

This appendix presents a 3×3 Job Shop Scheduling Problem (JSSP) example to illustrate how the decision layer routes user queries to EXPLAIN, ARGUE, or a hybrid mode. We focus on a single disjunction on machine C, showing how the system reacts depending on whether the query seeks understanding or questions a decision. The example combines contrastive and certificate-style explanations with preference-based argumentation. Table 2 displays the JSSP instance used, where each cell specifies the machine and processing time for each operation.

Table 2. JSSP instance. Each cell shows (machine, processing time).

Job	Op1	Op2	Op3
J_1	$(A, 3)$	$(B, 2)$	$(C, 2)$
J_2	$(B, 2)$	$(C, 1)$	$(A, 4)$
J_3	$(C, 4)$	$(A, 1)$	$(B, 3)$

We focus on the operations assigned to machine C: $o_{3,1}$ (duration 4), $o_{2,2}$ (duration 1), and $o_{1,3}$ (duration 2). Assume that $o_{3,1}$ occupies C during $[0, 4]$. Due to precedence constraints, $o_{2,2}$ becomes available at time 4 and $o_{1,3}$ at time 5. Operation $o_{2,3}$ (duration 4 on machine A) can only begin once $o_{2,2}$ has completed.

Local decision. On machine C, the decision is whether to schedule

$$P: \; o_{2,2} \prec o_{1,3} \qquad \text{or} \qquad \neg P: \; o_{1,3} \prec o_{2,2}.$$

We describe this local scheduling conflict as follows: after $o_{3,1}$ finishes at time 4, $o_{2,2}$ is ready at $t = 4$ and $o_{1,3}$ at $t = 5$. The disjunction determines their order on machine C.

A.1 Routing to EXPLAIN: "why" (contrastive) and "why-not" (certificate-style)

Explain-routing (why). For the question "Why is $o_{2,2} \prec o_{1,3}$ on C?", the decision layer selects EXPLAIN. A concise contrastive explanation compares P with the nearby alternative $\neg P$ (i.e., reversing the disjunction):

- Under P, $o_{2,2}$ runs on C during $[4, 5]$ and $o_{1,3}$ during $[5, 7]$. As a result, $o_{2,3}$ can start at time 5 and complete at time 9, so $C_{\max} = 9$.
- Under $\neg P$, $o_{1,3}$ cannot start before time 5, so it runs in $[5, 7]$ and pushes $o_{2,2}$ to $[7, 8]$. This delays the start of $o_{2,3}$ to time 8, which means it finishes at 12.

Therefore, choosing P reduces the makespan by $\Delta = 3$.

Explain-routing (why-not). For the question "Why not enforce $\neg P$ while keeping $C_{\max} \leq 9$?", the decision layer still selects EXPLAIN. However, in this case a certificate-style answer is more appropriate. A solver can return a small UNSAT core (including precedence constraints, machine no-overlap constraints, and $\neg P$), showing that $start(o_{2,3}) \geq 8$, and thus $C_{\max} > 9$ [5].

Figure 1 summarizes the two alternatives.

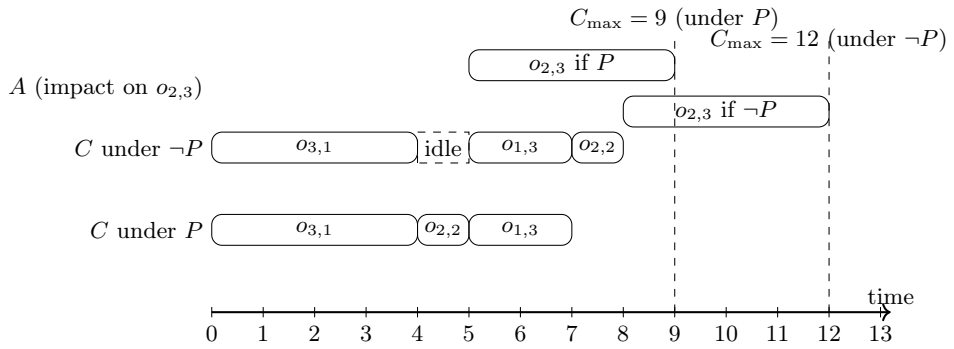

Fig. 1. Time-line summary: under P the downstream operation $o_{2,3}$ starts at 5 and finishes at 9; under $\neg P$ it starts at 8 and finishes at 12.

A.2 Routing to ARGUE: acceptance under preferences (rational persuasion)

Now consider the question: "I see the makespan gain, but why should I accept changing the prior plan?" This introduces an explicit value conflict (efficiency vs. stability), so the decision layer selects ARGUE.

Assume that a prior schedule π_0 used $\neg P$, and that planners tend to penalize deviations from π_0. We can consider the following two arguments:

- A_1 (Efficiency): accept P because it reduces the makespan by 3.
- A_2 (Stability): accept $\neg P$ because it preserves π_0, resulting in lower disruption.

These arguments rebut one another. In *Urgent* mode, we prioritize Efficiency over Stability (Efficiency $\succ$ Stability), so A_1 outweighs A_2 and P is accepted. In *Normal* mode, we reverse the preference order (Stability $\succ$ Efficiency), which may lead to $\neg P$ being accepted instead.

This is not just an explanation of the decision (D), but a justification based on preference reasoning under objection.

A.3 Mini-dialogue (optional)

Planner: Why is $o_{2,2}$ scheduled before $o_{1,3}$ on machine C?
System (EXPLAIN): Because reversing their order would delay $o_{2,2}$ to $[7, 8]$, which in turn delays $o_{2,3}$ and increases $C_{\max}$ from 9 to 12.
Planner: I see, but why should I accept this change if it disrupts the previous plan?
System (ARGUE): Under urgent-mode priorities (Efficiency $\succ$ Stability), the efficiency argument takes precedence, so P is the accepted option. In normal mode, where stability is prioritized, the opposite decision may be justified.

Minimizing Response Time in District-Based Security Networks via Integrated Policy Comparison*

Fatih Mehmet Yılmaz[1] and Tınaz Ekim[1]

Department of Industrial Engineering, Boğaziçi University, Istanbul, Turkey
{fatih.yilmaz1, tinaz.ekim}@bogazici.edu.tr

Abstract. This research investigates response time optimization in private security networks, balancing stochastic alarm response with mandatory patrol coverage. Using a Discrete-Event Simulation (DES) framework, we contrast distance-minimizing flows against the time-minimizing "Quick Supply" method. Results across five real-world topologies reveal a critical density threshold: while proactive rebalancing reduces response times by ∼4% at mid-densities, simple reactive baselines outperform complex optimization algorithms under critical scarcity (5% density).

Keywords: Security Networks · Dynamic Rebalancing · Discrete Event Simulation · Graph Theory · Dominating Sets.

1 Introduction and Literature Review

Private security operations differ from public emergency services by balancing stochastic incident response with mandatory visible patrol coverage. Dispatching a vehicle creates a coverage gap, necessitating a dynamic trade-off between response speed and contractual presence. We model this system on a graph governed by three coupled decisions: assignment (dispatch), coverage (positioning), and rebalancing (restoration).

Operations Research in this domain has evolved from static to dynamic models. Early studies focused on fixed resource allocation, employing simulated annealing for district design [1], the Patrol Car Allocation Model (PCAM) for fleet sizing [3], and maximal covering models for patrol areas [4]. Recent literature addresses real-time management, utilizing simulation to minimize police response times [6] and network flow models for relocation [5]. Theoretically, Bessy et al. [2] advanced these concepts through generalized domination sets.

This work adapts "fluidic" rebalancing algorithms from Demand-Responsive Transport (DRT), specifically Cheng's Min-Cost flows [7], to security constraints. We

* Supported by the European Commission's Horizon Europe Research and Innovation programme through the Marie Skłodowska-Curie Actions Staff Exchanges (MSCA-SE) under Grant Agreement no.101182819 (COVER).

develop a unified Discrete-Event Simulation (DES) framework to benchmark rebalancing strategies across distinct real-world city topologies. Unlike prior studies, we simultaneously address assignment, coverage, and rebalancing, providing a comparative evaluation of policy effectiveness under varying fleet densities.

2 Proposed Methodology and Algorithms

We use a Discrete-Event Simulator (DES) modeling a city divided into static, non-overlapping hexagonal districts (400m radius). Nodes represent weighted street centers, and edges connect nodes with driving distances under 750m. We generated 5 distinct real-world traffic maps based on these rules. Table 1 summarizes the topological characteristics, and Fig. 1 illustrates a representative topology.

Table 1. Properties of generated networks.

Map	$\|V\|$	$\|E\|$	**Diam.**	**MDS** (γ)
Map 1	535	1145	45	303
Map 2	214	458	22	131
Map 3	270	529	40	159
Map 4	1174	2025	63	504
Map 5	666	1366	57	390

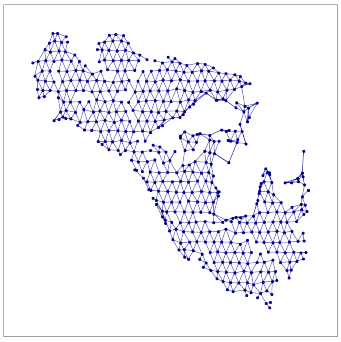

Fig. 1. District-based topology.

The simulation runs with a 10:1 time compression ratio under an expected load of 500 alerts per day, generated according to a Poisson distribution with mean $\lambda = 500/144$.

To isolate rebalancing effects, we employ a strict Nearest Neighbor heuristic, assigning the closest unit to an alarm regardless of global fleet distribution. Rebalancing is triggered by determined 100 unit time steps. The target state is a single **Minimum Dominating Set (MDS)** calculated via a greedy approximation.

The rebalancing strategies are: (1) **No Rebalancing**, where vehicles stay at their last drop-off; (2) **Greedy Heuristic**, where vehicles move to the nearest gap; (3) **Minimum Cost Maximum Flow (MCMF)**, which computes the minimum total distance traveled by all cars to satisfy the demand; and (4) **Quick Supply (Min-Makespan)**, which minimizes the **maximum response time**. Quick Supply ensures the rebalancing operation completes in the minimal number of time steps. It operates iteratively:

1. Deficit Calculation: Compute total flow imbalance F to satisfy demand.

2. Iterative Expansion: Let $d_G(u, v)$ denote the shortest travel time between districts u and v. Iterate horizons $i = 1, \ldots, \text{diam}(G)$ constructing a time-expanded network N_i:
   - Split district v into v^-, v^+. Add supersource s, supersink t.
   - Add supply (s, v^-), demand (v^-, t), and throughput (v^-, v^+).
   - Add temporal edges (u^+, v^-) if travel time $d_G(u, v) \leq i$.

A valid flow of value F in N_i represents a plan resolving imbalances within i steps.

3 Experimental Evaluation

Alert arrivals follow a Poisson distribution. Service times are Exponential ($\mu = 3$). Fleet Density (ρ) is the ratio of fleet size (k) to the MDS size ($\gamma(G)$). An alarm is "Missed" if total service time exceeds 12 unit times.

Table 2. Avg. Response Time (ART) and Handle Rate (HR) at critical fleet densities.

Density	Map	Metric	Q. Supply	MCMF	Greedy	No Rebal
(Mid) 35%	Map 1	ART	**6.20**	6.29	6.20	6.23
		HR	100%	100%	100%	100%
	Map 2	ART	6.72	6.91	**6.63**	7.05
		HR	100%	100%	100%	100%
	Map 3	ART	**7.34**	7.36	7.54	7.53
		HR	100%	100%	100%	100%
	Map 4	ART	**6.51**	6.51	6.63	6.77
		HR	100%	98.1%	97.9%	94.9%
	Map 5	ART	6.11	**6.08**	6.26	6.51
		HR	100%	100%	100%	100%
(Scarcity) 5%	Map 1	ART	17.00	16.92	17.06	17.11
		HR	48.8%	49.8%	47.8%	**51.0%**
	Map 2	ART	17.44	17.50	17.60	17.87
		HR	33.9%	33.4%	33.5%	**35.5%**
	Map 3	ART	17.81	17.51	17.22	17.47
		HR	31.9%	32.6%	**35.1%**	34.9%
	Map 4	ART	15.44	14.95	15.46	**14.76**
		HR	71.4%	74.7%	74.2%	**79.5%**
	Map 5	ART	15.61	16.36	15.69	16.23
		HR	58.9%	58.2%	60.4%	**60.8%**

At **Mid-Range Density (35%)**, the system remains stable with near-perfect handle rates; proactive optimization effectively minimizes response times. Conversely, **Extreme Scarcity (5%)** triggers an "Over-Optimization Trap," where rebalancing consumes capacity required for immediate dispatch. Consequently, the reactive **No Rebalancing** baseline proves superior, achieving the highest Handle Rate in 4 out of 5 topologies.

4 Future Work

Future work aims to evolve this simulation framework into a comprehensive **Decision Support Tool** for security operations, focusing on three key expansions:

1. **Adaptive Target Selection:** Using a library of Dominating Sets $\Omega = \{D_1, \dots\}$ and dynamically selecting the target D^* that minimizes movement cost.
2. **Robust Domination Strategies:** Expanding the target definitions to include advanced graph-theoretical models:
   - *Distance k-domination* to strictly minimize maximum response times;
   - *Eternal domination* (secure sets) to maintain coverage during consecutive attacks;
   - *m-eternal* and *defensive domination* (m-secure sets), which allow the entire fleet to relocate cooperatively ("all guards move") to counter simultaneous incidents.
3. **Dynamic Demand Modeling:** Moving beyond stable Poisson rates to model heterogeneous demand, where specific districts have varying risk rates based on time of day.

References

1. D'Amico, S.J., Wang, S.J., Batta, R., Rump, C.M.: A simulated annealing approach to police district design. Computers & Operations Research **29**(6), 667–684 (2002)
2. Bessy, S., et al.: Emergency response sets in graphs. arXiv preprint arXiv:2401.00000 (2024)
3. Chaiken, J.M., Dormont, P.: A patrol car allocation model: Capabilities and algorithms. Management Science **24**(12), 1291–1300 (1978)
4. Curtin, K.M., et al.: Determining optimal police patrol areas with maximal covering and backup covering location models. Netw. Spat. Econ. **10**, 125–145 (2010)
5. Yıldırım, B., Soylu, B.: Relocating emergency service vehicles with multiple coverage and critical levels partition. Comp. & Ind. Eng. **177**, 109016 (2023)
6. Dewinter, M., et al.: Reducing police response times: Optimization and simulation of everyday police patrol. Networks **84**(3), 363–381 (2024)
7. Cheng, L.: Extending the functionality of the DRT simulation infrastructure in MATSim. PhD thesis, TU Berlin (2024)

Specimen Collection Problem with Multiple Facilities and Batch Deliveries

İhsan Sadati[1,2], Halenur Şahin[3],, F. Sibel Salman[4], and Eda Yücel[3]

[1] Faculty of Engineering and Natural Sciences, Sabanci University, Istanbul, Türkiye
[2] Smart Mobility and Logistics Lab, Sabanci University, Istanbul, Türkiye
[3] Department of Industrial Engineering, TOBB University of Economics and Technology, Ankara, Türkiye
[4] College of Engineering, Koç University, Istanbul, Türkiye

Abstract. Clinical testing laboratories collect time-sensitive medical specimens from geographically dispersed healthcare facilities, such as hospitals and clinics, and deliver them to processing facilities with limited daily processing capacity. Collection sites may require multiple visits throughout the day, and specimens are released in batches with release times. We study the *Collection for Processing Problem with Multiple Facilities and Batch Deliveries* (CfPP-MFBD) to optimize the collection process. In this problem, a fleet of vehicles performs multiple tours, and each tour may deliver collected batches to one of several processing facilities. The objective is to minimize the total (size-weighted) arrival time of batches at processing facilities, thereby enabling earlier processing and reducing idleness. Assuming the batch sizes and release times are known, a routing and scheduling plan should be generated at the beginning of the day. We provide a mixed-integer linear programming (MILP) formulation for CfPP-MFBD that integrates routing, visit timing, batch feasibility, and workload evolution across multiple processing facilities. For computational experiments, small- to medium-sized instances have been generated from a clinical laboratory's historical data. These instances can be solved to optimality in a short time using the Gurobi solver. The model provides feasible baselines for larger instances derived from real operations.

Keywords: Healthcare logistics · specimen collection · routing · batch deliveries · MILP

1 Motivation and related work

Clinical testing laboratories must collect biomedical specimens from geographically dispersed sites (e.g., clinics and hospitals) and transport them to processing facilities for analysis. Since specimens are perishable and operational decisions affect both turnaround time and laboratory utilization, routing and scheduling decisions play a central role in obtaining the results timely.

Specimen logistics has been studied explicitly in the collection-for-processing setting. In particular, [1] introduce the Collection for Processing Problem (CfPP)

for clinical testing laboratories and highlight the interaction between pickup operations and limited processing capacity. Complementing this line of work, [2] study a multi-vehicle collection-for-processing problem and propose a constant-factor approximation algorithm that explicitly accounts for repeated vehicle tours and coordinated pickup decisions. Real-life biomedical transportation settings have also been investigated through case-study-driven optimization, such as the province-wide specimen transportation problem in Quebec [3], and exact/optimization-based approaches for centralized laboratory delivery [4].

In this extended abstract, we focus on a generalization that is motivated by current operational practice: (i) multiple processing facilities operate in parallel, (ii) specimens are released from sites in batches with release times during the day, and (iii) vehicles may perform multiple tours. We formulate a mixed-integer linear program (MILP) that captures the coupling between routing/timing decisions and processing dynamics, with the objective of minimizing the total delivery times of released batches.

2 Problem statement (CfPP-MFBD)

The problem is defined on a directed complete graph $G = (N, A)$ where $N = N^P \cup N^S$ is the set of nodes, N^P is the set of processing facilities, and N^S is the set of collection sites. Each arc $(i, j) \in A$ has a nonnegative travel time t_{ij}. A set of batches B is given. Each batch $b \in B$ has a release time r_b, a size s_b, and a binary indicator a_{ib} equal to 1 if batch b is accumulated at site $i \in N^S$. Each processing facility $i \in N^P$ can process specimens over a horizon $[0, \tau]$. A fleet of vehicles M starts at processing facilities at time 0; each vehicle can perform multiple tours during the day.

A *tour* starts at a processing facility, visits a sequence of sites, and ends at a processing facility. The key decisions are: (i) the sequence of visits in each tour, (ii) the visit times (including waiting), (iii) the assignment of collected batches to tours (hence to delivery facilities), and (iv) the implied arrival process at each processing facility. The objective is to minimize the sum of size-weighted delivery times of batches to processing facilities, i.e., to deliver larger batches earlier when possible.

3 MILP formulation (compact presentation)

We provide an MILP that combines the routing structure, visit timing, batch feasibility, and processing facility dynamics in a single formulation.

Sets and parameters. Vehicles are indexed by $m \in M$. For each vehicle, K_m denotes an ordered set of tours. For each processing facility $i \in N^P$, K_i denotes an ordered set of delivery positions at that facility (used to model the sequence of arrivals). Parameters include travel times t_{ij}, planning horizon τ, batch sizes s_b, release times r_b, and indicators a_{ib}.

Routing and batch variables. Binary variables $X_{ijmk'}$ indicate whether node j is visited immediately after node i in vehicle m's tour $k' \in K_m$; binary variables $Y_{imk'}$ indicate whether node i is visited in that tour; binary variables $B_{bmk'}$ indicate whether batch b is collected in that tour.

Timing variables. Continuous variables $T_{imk'}$ and $W_{imk'}$ represent visit and waiting times, respectively, and $E_{mk'}$ represents the end time of tour k' of vehicle m. To connect vehicle tours to the delivery sequence at facilities, binary variables $A_{imkk'}$ assign vehicle tour (m, k') to position $k \in K_i$ of facility i. Additional continuous variables are defined to compute the delivery time C'_b of each batch.

Objective. The objective minimizes size-weighted batch delivery times:

$$\min \sum_{b \in B} s_b \, C'_b. \tag{1}$$

Key constraints (selection). The full formulation includes the following constraint groups. *(i) Tour structure and flow.* Flow balance constraints enforce valid predecessor/successor relations for visited nodes and ensure that tours start and end at processing facilities, while allowing empty tours. *(ii) Batch feasibility.* Each batch can be collected at most once; a batch can be collected only if a tour visits its originating site, and only after the batch is released. *(iii) Time propagation.* Visiting times are propagated along selected arcs using big-M constraints, capturing travel and waiting times and linking them to tour end times. *(iv) Facility coupling.* Assignment constraints ensure that each non-empty vehicle tour is assigned to exactly one facility position.

Linearization. A small number of constraints contain bilinear terms that multiply continuous time variables with binary assignment variables. These terms are linearized using standard auxiliary variables and big-M bounds, yielding a pure MILP.

4 Computational results: Gurobi on the MILP

We evaluated the proposed MILP using Gurobi with a 2-hour (7,200-second) time limit per instance. In addition, we set `model.Parameters.NoRelHeurTime = 60` to control the time spent in heuristics before focusing on improving the incumbent bound. Computational experiments were conducted on a computer equipped with an Intel Core i7-8700 (3.2 GHz) CPU and 32 GB RAM. The mathematical model was implemented in C# and solved using Gurobi through its .NET interface (Microsoft® Visual Studio 2026).

We considered 10 benchmark instances generated from historical data belonging to a clinical laboratory. As in the rest of the paper, instances are denoted as *p*p*s*s*v*v*b*b, where p is the number of processing centers, s is the number of sites, v is the number of vehicles, and b is the number of batches.

Table 1. Gurobi results on the MILP (time limit = 2 hours; `NoRelHeurTime` = 60; run time in seconds).

Instance	Objective	Time (s)
1p5s1v5b	8 963.24	20.12
1p5s1v10b	145 641.30	19.58
1p5s1v15b	277 040.20	21.26
2p10s1v10b	272 220.942	210.29
2p10s1v15b	502 945.128	1 103.25
2p10s2v10b	205 243.438	1 475.67
3p10s1v10b	320 176.56	361.50
3p10s3v10b	149 252.215	5 896.59
3p6s3v7b	80 539.292	68.89
3p6s3v6b	58 753.422	80.98
Average	–	925.81

Table 1 reports, for each instance, the optimal objective value found by Gurobi, and the run time. Gurobi solved all 10 instances to proven optimality within the time limit. The average run time is 925.81 seconds (approximately 15.43 minutes). The results suggest that the proposed MILP can be solved effectively for small and medium-sized instances, and can provide optimal solutions within a reasonable time on the tested set.

5 Conclusion

CfPP-MFBD models clinical specimen collection with repeated site visits, batch-based releases, and multiple processing facilities. We present an MILP formulation for this routing and scheduling problem that can be solved to optimality on small and medium-sized instances. The Gurobi results indicate that the integrated formulation becomes computationally challenging as the problem size grows, motivating advanced solution approaches (e.g., decomposition or metaheuristics) for large-scale daily operations while retaining the MILP as a benchmarking tool and as a core component within hybrid methods.

References

1. Yücel, E., Salman, F., Gel, E., Örmeci, E., Gel, A.: Optimizing specimen collection for processing in clinical testing laboratories. *European Journal of Operational Research* **227**, 503–514 (2013)
2. Yücel, E., Salman, F.S., Örmeci, E.L., Gel, E.S.: A constant-factor approximation algorithm for multi-vehicle collection for processing problem. *Optimization Letters* **7**(7), 1627–1642 (2013)
3. Anaya-Arenas, A.M., Chabot, T., Renaud, J., Ruiz, A.: Biomedical sample transportation in the province of Quebec: A case study. *International Journal of Production Research* **54**, 1–14 (2015)
4. Zabinsky, Z.B., Dulyakupt, P., Zangeneh-Khamooshi, S., Xiao, C., Zhang, P., Kiatsupaibul, S., Heim, J.A.: Optimal collection of medical specimens and delivery to central laboratory. *Annals of Operations Research* **287**, 537–564 (2020)

Modeling the Colorful Components Problems via Representative Formulations

Carmine Sorgente[1][0000−0001−6053−7318], Martina Cerulli[2][0000−0002−6341−8143], Claudia Archetti[3][0000−0002−3524−1600], and Diego Delle Donne[4][0000−0002−7656−5003]

[1] Department of Mathematics, University of Salerno, Fisciano, Italy
csorgente@unisa.it
[2] Department of Computer Science, University of Salerno, Fisciano, Italy
mcerulli@unisa.it
[3] Department of Economics and Management, University of Brescia, Brescia, Italy
claudia.archetti@unibs.it
[4] IDO Department, ESSEC Business School, Cergy-Pontoise, France
delledonne@essec.edu

Abstract. In a node-colored graph, a colorful component is a connected subgraph in which no two nodes share the same color. Colorful components problems are graph modification problems that aim to transform a node-colored graph into a collection of colorful components by deleting edges.
We consider three variants of colorful components problems that have been studied in the literature: the MOP, MCC, and MEC problems. For them, we propose new integer programming formulations based on representative nodes, where each component is identified by a representative, eliminating the symmetry inherent in classical assignment-based models. Moreover, we introduce a novel flow-based modeling approach to enforce component connectivity through additional variables and constraints. To further improve computational efficiency, we also develop a preprocessing procedure that reduces the number of variables in the representative formulations. Computational experiments demonstrate that the proposed models, particularly those using flow-based connectivity constraints, significantly outperform the current state-of-the-art approach, solving all benchmark instances to optimality within negligible computational time.

Keywords: combinatorial optimization · colorful components · representative formulations · flow connectivity

1 Introduction

In this work, we address the NP-hard colorful components problems considered in [2], namely the Minimum Orthogonal Partition (MOP) problem [4] (also known as Colorful Components [3]), the Minimum Colorful Components (MCC) problem [1], and the Maximum Edges in Transitive Closure (MEC) problem [5].

Let $\mathcal{G} = (V, E, C)$ be a node-colored undirected graph, where each node $u \in V$ is assigned a color $c_u \in C$. A connected component induced by a node set $S \subseteq V$ is said to be colorful if $c_u \neq c_v$ for all distinct $u, v \in S$. The transitive closure of $\mathcal{G}$ is the graph $\mathcal{H} = (V, E_{\mathcal{H}})$, where $\{i, j\} \in E_{\mathcal{H}}$ if and only if i and j belong to the same connected component of $\mathcal{G}$.

The three problems aim to delete a subset of edges so that all resulting components are colorful, while respectively minimizing the number of removed edges (MOP), minimizing the number of resulting components (MCC), or maximizing the number of edges in the transitive closure (MEC).

We propose representative-based formulations in which each connected component is associated with a unique representative node, chosen as the smallest-index node in the component to eliminate symmetry. For each node $u \in V$, we define the set of nodes representable by u as $V_u = \{v \in V : u < v \text{ and } c_u \neq c_v\} \cup \{u\}$, and let E_u be the set of edges induced by V_u.

2 Formulations of the Problems

In order to model the partition of the graph $\mathcal{G}$ into colorful components via representative formulations, for each pair of nodes $u, v \in V$, with $v \in V_u$, we define a binary decision variable $z_{uv} = 1$ if u is the representative of v.

For the MOP problem, no connectivity constraints are needed, as already observed in [2]. Instead, for the MCC and the MEC problems, connectivity within each component is enforced through two different approaches.

First, *flow variables and constraints* are used to impose the connectivity of the components. The key idea is that the component represented by a node $u \in V$ must admit a feasible flow exiting from u, where each node assigned to u consumes one unit of flow, therefore guaranteeing that all such nodes are connected. Although the graph is undirected, connectivity is modeled using directed arcs in both directions. Let $A = \{(v, w), (w, v) : v, w \in E\}$ denote the set of directed arcs. In addition to the representation variables z, we introduce nonnegative flow variables f, where f^u_{vw} denotes the amount of flow of the component represented by u on arc $(v, w) \in A$. These variables and the associated constraints ensure that all nodes assigned to the same representative are reachable from the representative, and hence from one another.

Second, leveraging the approach proposed in [2], the connectivity is enforced through an exponential number of *connectivity constraints*.

3 Solution Approach

The flow-based formulations involve a polynomial number of variables and constraints, so they are inserted as such in the state-of-the-art solver Gurobi. Instead, the connectivity-constraints formulations involve an exponential number of connectivity constraints. Therefore, we propose a branch-and-cut approach based on the dynamic generation of these constraints, through classic *lazy constraints* callback technique.

Furthermore, we propose a preprocessing procedure that restricts the set V_u by only considering the nodes connected to u by means of a colorful path, i.e., a path in which every node has a different color. If such a path does not exist, then u and v cannot belong to the same component. In particular, given a node u, the preprocessing procedure consists of two stages. At the first stage: (i) we compute the shortest path between u and all the nodes in V_u, (ii) we remove from V_u all the nodes v for which the shortest path connecting v with u has a length greater than the number of colors. At the second stage, for each node that is still in V_u, we check whether the identified shortest path from u is colorful. If so, these nodes can be represented by u. For the remaining nodes, we verify if a colorful path exists connecting them to u.

4 Numerical Results

We performed computational experiments using the dataset considered in [2], which consists of 409 test instances with up to 210 nodes and 248 edges, using up to 9 colors. A 3.40GHz Intel(R) Core(TM) i7-3770 CPU with 16 GB RAM was used to implement all formulations in Python 3.10 and solve them using Gurobi 10.0.2. We set a one-hour time limit and a 10 GB memory limit.

We compare the following approaches:

- **B&C**: Best configuration of the branch-and-cut approach proposed in [2];
- **RC**: Representative formulations with connectivity constraints (for MCC and MEC), with the preprocessing procedure.
- **RF**: Representative formulations with flow variables and constraints to model connectivity (for MCC and MEC), with the preprocessing procedure.

The results for the three problems are reported in Table 1. Since the MOP problem requires no connectivity constraints, only B&C and RF are compared. The headings of the table are: *#opt*, which is the number of instances solved to optimality; *time*, which is the average computational time in seconds; *gap*, which is the average percentage gap at termination, computed as $100(UB - LB)/UB$.

	MOP			**MCC**			**MEC**		
	#opt	**time**	**gap**	**#opt**	**time**	**gap**	**#opt**	**time**	**gap**
B&C	409	5.47	0.00%	397	113.57	0.43%	394	164.17	2.05%
RC	409	0.64	0.00%	409	0.67	0.00%	405	45.24	0.05%
RF				409	0.64	0.00%	409	0.87	0.00%

Table 1.

For the MOP, we note that both B&C and RF solve all the 409 instances to optimality. However, the RF approach is faster.

The differences among the considered approaches are more evident when considering the MCC and the MEC problem. In particular, we observe that the two formulations introduced in this paper achieve better results w.r.t. the branch-and-cut approach proposed in [2]. All versions of the RC formulation yield smaller average percentage gaps and faster average computational times than both B&C. The RF formulation is able to solve all the instances to optimality in less than one second on average.

References

1. Adamaszek, A., Popa, A.: Algorithmic and hardness results for the colorful components problems. In: Pardo, A., Viola, A. (eds.) LATIN 2014: Theoretical Informatics. pp. 683–694. Springer Berlin Heidelberg (2014). https://doi.org/10.1007/978-3-642-54423-1_59
2. Archetti, C., Cerulli, M., Sorgente, C.: Branch-and-cut algorithms for colorful components problems. INFORMS Journal on Computing (2025). https://doi.org/10.1287/ijoc.2024.0927
3. Bruckner, S., Hüffner, F., Komusiewicz, C., Niedermeier, R., Thiel, S., Uhlmann, J.: Partitioning into colorful components by minimum edge deletions. In: Kärkkäinen, J., Stoye, J. (eds.) Combinatorial Pattern Matching. pp. 56–69. Springer Berlin Heidelberg (2012). https://doi.org/10.1007/978-3-642-31265-6_5
4. He, G., Liu, J., Zhao, C.: Approximation algorithms for some graph partitioning problems. Journal of Graph Algorithms and Applications **4**, 1–11 (2000). https://doi.org/10.7155/jgaa.00021
5. Zheng, C., Swenson, K., Lyons, E., Sankoff, D.: OMG! Orthologs in Multiple Genomes – Competing Graph-Theoretical Formulations. In: Przytycka, T.M., Sagot, M.F. (eds.) Algorithms in Bioinformatics. pp. 364–375. Springer Berlin Heidelberg (2011). https://doi.org/10.1007/978-3-642-23038-7_30

On the Polyhedral Structure of the Split-Interval Coloring Problem

Diego Delle Donne[1][0000−0002−7656−5003] and Javier Marenco[2][0000−0003−2694−4758]

[1] ESSEC Business School, France (delledonne@essec.edu)
[2] Universidad Torcuato Di Tella, Argentina (javier.marenco@utdt.edu)

Abstract. We introduce the *split-interval coloring* problem (SIC), a natural extension of interval coloring in which each vertex may be assigned up to two non-overlapping integer intervals whose total length satisfies a prescribed demand. The model is motivated by berth allocation settings where vessel service may be preempted at most once, and where additional incompatibilities arise from shared resources such as tugboats, cranes, or access channels. We propose an integer programming formulation and study the associated polytope. Our main contribution is a polyhedral toolbox that systematically transforms valid inequalities for the classical interval-coloring polytope into (typically nonlinear) valid inequalities for SIC, provides linearization mechanisms based on McCormick-type arguments, and exploits the inherent symmetry between each vertex and its twin to generate additional valid and facet-inducing inequalities. We illustrate the approach by deriving several families of valid inequalities and identifying conditions under which they induce facets.

Keywords: interval coloring · integer programming · polyhedral combinatorics

1 Introduction

Given a graph $G = (V, E)$, demands $d \in \mathbb{Z}_+^{|V|}$, and a set of consecutive colors $C = \{1, \ldots, c\}$, the classical *interval coloring* problem assigns to each vertex $i \in V$ a bounded integer interval $I_i \subseteq C$ of length $|I_i| = d_i$ such that adjacent vertices receive disjoint intervals [5]. Interval coloring models several resource allocation problems, including bandwidth allocation and berth allocation [1, 2].

We study an extension of the classical interval coloring problem motivated by berth allocation in container terminals, where vessels must be assigned time intervals for loading and unloading operations. In realistic settings, the service of a vessel may be split into multiple non-contiguous intervals, and certain pairs of vessels cannot be served simultaneously due to shared resources such as tugboats, cranes, or access channels. These incompatibilities translate into adjacency constraints in a conflict graph, where overlapping service intervals are forbidden for adjacent vertices. This setting naturally leads to a generalized interval coloring framework in which each vertex may receive multiple time intervals while

respecting pairwise incompatibility constraints. We define the *split-interval coloring* (SIC) as follows.

Definition 1 (Split-interval coloring). *Given (G, d, c), decide whether each vertex $i \in V$ can be assigned up to two non-overlapping bounded integer intervals $I_i, I'_i \subseteq C$ such that $|I_i|+|I'_i| = d_i$ and any interval assigned to i does not overlap any interval assigned to a neighbor of i.*

SIC is NP-complete since it contains vertex coloring as the special case $d_i = 1$ for all $i \in V$. This motivates integer programming and polyhedral methods.

Twin-graph view. Let $G' = (V', E')$ be obtained from G by duplicating each vertex $i \in V$ into a twin i' and adding edges so that i and i' are adjacent and the duplication preserves adjacencies (i.e., edges among originals are replicated across all pairs of originals/twins). A split-interval coloring of G corresponds to an interval coloring of G' where the lengths assigned to i and i' sum to d_i.

Related "multi-interval" variants have been considered in different contexts (e.g., constrained preemption and splittable tasks) [4, 9, 10]. Our focus is on the polyhedral structure induced by the two-interval restriction and the twin-graph representation.

2 Formulation and the SIC polytope

For each vertex $i \in V'$, we introduce integer variables l_i, r_i representing the interval $[l_i, r_i)$ (possibly empty if $l_i = r_i$). For each edge $ij \in E'$ (assume $i < j$), we use binary variables x_{ij} and x_{ji} where $x_{ij} = 1$ means $[l_i, r_i)$ is entirely before $[l_j, r_j)$. A feasible split-interval coloring is encoded by:

$$(r_i - l_i) + (r_{i'} - l_{i'}) = d_i \qquad \forall i \in V \tag{1}$$

$$r_i \le l_j + c(1 - x_{ij}) \qquad \forall ij \in E',\ i < j \tag{2}$$

$$r_j \le l_i + c(1 - x_{ji}) \qquad \forall ij \in E',\ i < j \tag{3}$$

$$x_{ij} + x_{ji} = 1 \qquad \forall ij \in E',\ i < j \tag{4}$$

$$0 \le l_i \le r_i \le c,\ \ l_i, r_i \in \mathbb{Z}_+ \qquad \forall i \in V' \tag{5}$$

$$x_{ij}, x_{ji} \in \{0, 1\} \qquad \forall ij \in E',\ i < j. \tag{6}$$

Constraints (2)–(4) prevent overlaps on every edge of G', while (1) enforces total demand per original vertex. We denote by $P_{SIC}(G, d, c)$ the convex hull of all (l, r, x) satisfying (1)–(6).

Let χ_{IC} (resp. χ_{SIC}) be the minimum number of colors required by IC (resp. SIC). Clearly, $\chi_{\mathrm{SIC}} \le \chi_{\mathrm{IC}}$ and examples in which the inequality is strict are simple to find.

3 From IC inequalities to SIC inequalities

Let $P_{IC}(G, d, c)$ be the classical interval-coloring polytope expressed over (l, x) (with $r_i = l_i + d_i$ for original vertices). Many strong inequalities are known for $P_{IC}(G, d, c)$ [6, 7].

Our main observation is that IC inequalities typically depend on the demands d_i. In SIC, however, the service of a vertex i may be split between i and its twin i', so that the *effective* length assigned to a single interval is generally smaller than d_i. This naturally leads to replacing occurrences of d_i by the variable length $(r_i - l_i)$ associated with one of the two intervals of vertex i. For instance, the IC-valid inequality $d_i x_{ij} \leq l_j$ becomes the SIC-valid (but nonlinear) inequality $(r_i - l_i)x_{ij} \leq l_j$.

Conversion principle. We formalize this replacement via our *conversion lemma*, which maps any valid inequality for $P_{IC}(G, d, c)$ into a valid (possibly nonlinear) inequality for $P_{SIC}(G, d, c)$ by replacing each demand coefficient d_i with the effective length $(r_i - l_i)$ assigned to the original vertex i, together with the appropriate projection onto the variables associated with the original graph.

Linearization principle. Converted inequalities are often bilinear due to terms such as $(r_i - l_i)x_e$. We provide linearization tools inspired by McCormick envelopes [8]. In particular, when $\alpha > 0$, a term $\alpha(r_i - l_i)x_e$ can be replaced by a linear expression using the bound $0 \leq r_i - l_i \leq d_i$ implied by (1)–(5). This yields our *flattening lemma*, which preserves validity and, under mild conditions, also facetness.

Symmetry principle. Finally, because each vertex i and its twin i' play symmetric roles in the formulation, our *replacement lemma* allows us to transfer any valid (or facet-inducing) inequality to its counterpart obtained by swapping i and i'. In practice, this ensures that all derived inequality families naturally apply to both intervals of each original vertex.

Together, *conversion, flattening and replacement* yield a systematic way to transfer polyhedral knowledge from IC to SIC.

4 Families of valid inequalities

We illustrate the toolbox with representative families of valid inequalities for $P_{SIC}(G, d, c)$. All inequalities are derived by combining the conversion, flattening, and replacement principles introduced above.

Clique-type inequalities. Let $i \in V'$ and let $K \subseteq N_{G'}(i)$ be a clique in G'. Then the following inequality is valid for $P_{SIC}(G, d, c)$:

$$l_i \;\geq\; \sum_{k \in K} (r_k - l_k) \;-\; d_k \, x_{ik}. \tag{7}$$

When K contains no vertex together with its twin and c is sufficiently large with respect to χ_{IC} and the demands, (7) is facet-inducing. Compared with the IC setting, splitting provides additional degrees of freedom that strengthen the inequality and simplify facetness conditions.

Double-clique inequalities. Let $ij \in E'$ with $j \neq i'$ and let $K \subseteq N_{G'}(i) \cap N_{G'}(j)$ be a clique. Then:

$$l_j - r_i \;\geq\; \sum_{k \in K} (r_k - l_k) \;-\; d_k\big(2 - x_{ik} - x_{kj}\big) \;-\; (c - \sum_{k \in K} d_k)(1 - x_{ij}) \qquad (8)$$

is valid for $P_{SIC}(G, d, c)$, and under analogous mild conditions it induces a facet.

Path inequalities. For four distinct vertices $i, j, k, t \in V$ with $\{ij, jk, kt\} \subseteq E$, we obtain valid (and, for $c > \chi_{\mathrm{IC}}$, facet-inducing) path-type inequalities that generalize classical IC path inequalities [3] after conversion and partial flattening.

5 Concluding remarks

We initiated a polyhedral study of SIC, motivated by berth allocation with limited preemption and additional incompatibility constraints. Our formulation and polyhedral toolbox provide a principled way to derive strong inequalities for SIC from the rich literature on interval coloring. Future work includes computational separation, branch-and-cut implementations, and a deeper investigation of structural graph classes relevant to port operations.

References

1. Bierwirth, C., Meisel, F.: A survey of berth allocation and quay crane scheduling problems in container terminals. European Journal of Operational Research **202**(3), 615–627 (2010)
2. de Werra, D., Gay, Y.: Chromatic scheduling and frequency assignment. Discrete Applied Mathematics **49**(1), 165–174 (1994), special Volume Viewpoints on Optimization
3. Gerhardt, A.: Polyedrische Untersuchungen von Zwei-Maschinen-Scheduling-Problemen mit Antiparallelitätsbedingungen. Master thesis, Technische Universität Berlin (1999)
4. Halldórsson, M.M., Kortsarz, G., Proskurowski, A., Salman, R., Shachnai, H., Telle, J.A.: Multicoloring trees. Information and Computation **180**(2), 113–129 (2003)
5. Kubale, M.: Interval vertex-coloring of a graph with forbidden colors. Discrete Mathematics **74**(1), 125–136 (1989), special Double Issue
6. Marenco, J., Wagler, A.: On the combinatorial structure of chromatic scheduling polytopes. Discrete Applied Mathematics **154**(13), 1865–1876 (2006)
7. Marenco, J.L., Wagler, A.K.: Chromatic scheduling polytopes coming from the bandwidth allocation problem in point-to-multipoint radio access systems. Annals of Operations Research **150**(1), 159–175 (March 2007)
8. McCormick, G.P.: Computability of global solutions to factorable nonconvex solutions: Part I: Convex underestimating problems. Mathematical Programming **10**, 147–175 (1976)
9. Shachnai, H., Tamir, T., Woeginger, G.J.: Minimizing makespan and preemption costs on a system of uniform machines. Algorithmica **42**(3), 309–334 (2005)
10. Thevenin, S., Zufferey, N., Potvin, J.Y.: Graph multi-coloring for a job scheduling application. Discrete Applied Mathematics **234**, 218–235 (2018)

Anonymizing networks via mathematical programming

Raffaele Cerulli[1][0000−0002−3277−6802], Ivana Ljubić[2][0000−0002−4834−6284], and Carmine Sorgente[1][0000−0001−6053−7318]

[1] Department of Mathematics, University of Salerno, Fisciano, Italy
{raffaele,csorgente}@unisa.it
[2] IDO Department, ESSEC Business School, Cergy-Pontoise, France
ivana.ljubic@essec.edu

Abstract. The release of social network data sets for analysis purposes often raises privacy concerns among users. Even when explicit user identifiers are removed, the risk of identity disclosure is not negligible, since uniquely distinguishable structural features may be exploited by adversaries to identify the nodes of the network associated with specific individuals. To protect individuals' privacy, networks are commonly anonymized before publication. We consider the k-degree anonymity model, which protects against re-identification attacks based on degree background knowledge. According to this model, a network is k-degree anonymous if each user has the same number of connections as at least $k-1$ other users. The k-degree anonymization problem aims to identify the smallest set of modifications of the edge set of a network needed to make it k-degree anonymous. We address three NP-hard versions of the problem, involving edge insertions, edge deletions, or both, with mathematical programming tools. Specifically, we introduce two families of ILP formulations and propose an ILP-based heuristic approach based on imposing that the order of the nodes of the network inherited from its degree sequence remains unchanged after performing the modifications, which allows for an efficient ILP reformulation with the k-degree anonymity constraints. We additionally derive combinatorial bounds on the largest degree of the anonymized network, helping reduce the size of the proposed formulations. We conduct computational experiments on benchmark social networks and scale-free networks, we compare the obtained results with those produced by two state-of-the-art approaches, and we perform an instance space analysis to identify the features that mostly affect the performance of the formulations.

Keywords: Combinatorial Optimization · Mathematical Programming · Social Networks · k-Degree Anonymity · Cyber Security

1 Introduction

Identity disclosure is one of the main risks to be considered when publishing social network data sets. Naïve anonymization, consisting in replacing user identifiers such as names and social security numbers with random IDs, does not

protect the network from adversarial attacks that exploit structural properties of the anonymized nodes to recognize the identities of target individuals [1]. If multiple identities are revealed, the adversary can derive information about how the users participate in the network and potentially discover confidential connections and communities. To mitigate this risk, the network can be modified to reduce the likelihood that adversaries can re-identify anonymized users. This modification process aims to ensure that the network complies with a specified anonymity model while preserving network utility as much as possible.

An attacker typically uses some background knowledge to recognize specific patterns in the network data. Different anonymity models exist, each one protecting from attacks assuming a specific type of background knowledge. One of the fundamental forms of knowledge is the number of social connections established by an individual. If such a number is unique or rare within the network, the adversary can easily identify the corresponding node in the released network.

We consider the following definition of k-degree anonymity of a graph, introduced in [5] to protect against attacks based on degree knowledge.

Definition 1. *A graph $G = (V, E)$ is k-degree anonymous if and only if, for each node $i \in V$, there exist at least other $k-1$ nodes in $V \setminus \{i\}$ having the same degree as i.*

To ensure k-degree anonymity, we address the following optimization problem.

Definition 2. *Given a graph G with n nodes, and an integer $2 \leq k \leq n-1$, the k-degree anonymization problem aims to perform as few graph-modification operations as possible to transform G into a k-degree anonymous graph.*

Graph modifications include edge insertions, edge deletions, or both, and we refer to the three related versions of the problem as Anonym-E-Ins, Anonym-E-Del, and Anonym-E-Ins-Del, respectively. All three versions of the problem are NP-hard on general graphs [4,7], and several heuristic approaches have been proposed to tackle them in the literature [2,3,5,6].

2 Contributions

This work applies integer linear programming (ILP) techniques to k-degree anonymous networks. We present two families of mathematical formulations for the three versions of the k-degree anonymization problem, allowing for edge insertions, deletions, and both. The first family includes *extended* formulations that make use of auxiliary variables to identify pairs of nodes having the same degree in the anonymized network. The second family instead includes *disaggregated* formulations, which incorporate multiple variables per node indexed by their possible node degrees in the anonymized network.

We analyze some properties of the problem and derive combinatorial bounds on the smallest and largest degrees of a node in the anonymized network, which help reduce the size of the proposed formulations. Finally, to tighten these bounds, we exploit the solutions obtained by an ILP-based heuristic approach,

working by imposing that the nodes have the same non-increasing degree ordering both in the original and in the anonymized networks, which enables an efficient reformulation of the k-degree anonymity constraints.

3 Results

We implemented the proposed formulations in Python 3.8 and tested them, using the Gurobi solver (version 10.0.2), on two sets of instances, including benchmark social and scale-free networks. The results showed that the disaggregated formulations outperform the extended ones for all three versions of the problem, both in terms of solution quality and computational efficiency. Furthermore, the bounds computed starting from the ILP-based heuristic solutions considerably speed up the solution process.

An instance space analysis [9], conducted using the Melbourne Algorithm Test Instance Library with Data Analytics (MATILDA) tools [8], showed that network size, sparsity, and degree discrepancy significantly affect the performance of the formulations. Nevertheless, the proposed formulations consistently improved the heuristic solutions produced by two state-of-the-art approaches.

References

1. Backstrom, L., Dwork, C., Kleinberg, J.: Wherefore art thou r3579x? anonymized social networks, hidden patterns, and structural steganography. In: Proceedings of the 16th International Conference on World Wide Web. p. 181–190. WWW '07, Association for Computing Machinery, New York, NY, USA (2007). https://doi.org/https://doi.org/10.1145/1242572.1242598
2. Casas-Roma, J., Herrera-Joancomartí, J., Torra, V.: k-Degree anonymity and edge selection: improving data utility in large networks. Knowledge and Information Systems **50**(2), 447–474 (2017). https://doi.org/10.1007/s10115-016-0947-7
3. Hartung, S., Hoffmann, C., Nichterlein, A.: Improved upper and lower bound heuristics for degree anonymization in social networks. In: Gudmundsson, J., Katajainen, J. (eds.) Experimental Algorithms. pp. 376–387. Springer International Publishing, Cham (2014)
4. Hartung, S., Nichterlein, A., Niedermeier, R., Suchý, O.: A refined complexity analysis of degree anonymization in graphs. Information and Computation **243**, 249–262 (2015). https://doi.org/10.1016/j.ic.2014.12.017, 40th International Colloquium on Automata, Languages and Programming (ICALP 2013)
5. Liu, K., Terzi, E.: Towards identity anonymization on graphs. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data. p. 93–106. SIGMOD '08, Association for Computing Machinery, New York, NY, USA (2008). https://doi.org/https://doi.org/10.1145/1376616.1376629
6. Lu, X., Song, Y., Bressan, S.: Fast Identity Anonymization on Graphs. In: Liddle, S.W., Schewe, K.D., Tjoa, A.M., Zhou, X. (eds.) Database and Expert Systems Applications. pp. 281–295. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
7. Meyerson, A., Williams, R.: On the complexity of optimal k-anonymity. In: Proceedings of the Twenty-Third ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems. p. 223–228. PODS '04, Association for Computing Machinery, New York, NY, USA (2004). https://doi.org/10.1145/1055558.1055591

8. Smith-Miles, K., Muñoz, M.A., Neelofar, N.: Matilda: Melbourne Algorithm Test Instance Library with Data Analytics (2020), https://matilda.unimelb.edu.au/matilda/
9. Smith-Miles, K., Muñoz, M.A.: Instance space analysis for algorithm testing: Methodology and software tools. ACM Computing Surveys **55**(12), 1–31 (2023)

Optimized Deployment of UAV-Base Stations in Post-Earthquake Scenarios

Safiye Aybala Kılıç[1][0009-0003-9927-8415], Betül Çoban Yelken[2][0009-0004-3384-678X], and Eda Yücel[1][0000-0002-3448-1522]

[1] TOBB University of Economics and Technology, Ankara, Turkey
[2] Abdullah Gul University, Kayseri, Turkey
betul.coban@agu.edu.tr

Abstract. In the aftermath of large-scale disasters, terrestrial communication infrastructure is often severely damaged, leading to significant disruptions in information flow and emergency coordination. Unmanned Aerial Vehicles (UAVs) deployed as mobile base stations offer a promising solution for providing critical communication support in the immediate post-disaster period. However, the effective utilization of such systems is constrained by limited battery capacity, fluctuating user demand, and the need for coordinated operation with charging infrastructure. This study addresses the UAV-Based Mobile Base Station Deployment Problem by considering a fleet of UAVs operating in a disaster-affected area that must periodically visit charging stations due to energy limitations. The positioning of UAVs, service assignment, movement decisions, and charging operations are jointly modeled through a mixed-integer linear programming (MILP) formulation. The objective is to maximize service coverage and communication reliability under operational constraints. Computational experiments conducted using the Gurobi solver demonstrate that the proposed model yields effective results for realistic settings with limited size. The findings indicate that while the proposed model performs effectively on the tested problem instances, it also reveals the necessity of adopting heuristic approaches for larger-scale grid structures and more extensive UAV fleets.

Keywords: UAV base stations · Disaster management · UAV configurations · Configuration-based modeling · Optimization · MILP

1 Motivation and Related Work

Reliable communication is a critical requirement in disaster response operations. Earthquakes, floods, and large-scale natural disasters often damage terrestrial base stations, preventing affected populations from accessing essential communication services. In recent years, Unmanned Aerial Vehicle–based base stations (UAV-BSs) have emerged as a flexible and rapidly deployable solution to restore connectivity in such environments.

Previous studies have mainly focused on positioning UAVs to maximize coverage, considering both two-dimensional and three-dimensional deployment scenarios [1,2,3]. In these works, coverage performance, quality of service, and user density are treated as primary performance metrics, while the deployment is generally assumed to be static. Studies incorporating energy constraints have aimed at minimizing UAV flight and hovering time by jointly optimizing UAV placement and bandwidth allocation [4,5]. However, in most of these approaches, charging processes are either simplified or considered independently from operational planning. Due to the NP-hard nature of the UAV placement problem, heuristic and learning-based approaches have also been proposed. In this context, particle swarm optimization and reinforcement learning–based methods have been employed to address the deployment of heterogeneous UAVs with different capabilities [6,7]. Moreover, several studies have formulated joint UAV base station placement and bandwidth allocation problems using mixed-integer optimization models [8,9]. Nevertheless, existing studies do not consider the temporal dimension, charging requirements, and operational continuity in post-disaster scenarios in an integrated manner.

In this study, we adopt a configuration-based modeling approach that differentiates from existing methods in the literature. Under this framework, the operational state of the UAV fleet at any given time is represented by a set of predefined configurations that jointly encode key attributes, including UAV locations and service statuses. This representation enables a systematic and structured characterization of the decision space while rigorously preserving the core operational constraints of the problem. In particular, it naturally captures the overlap of coverage regions among multiple UAVs, which precludes independent modeling of individual UAV service rates and is often oversimplified in conventional formulations. Through the proposed approach, UAV mobility, coverage decisions, and energy management components can be jointly addressed within a unified optimization framework, allowing the overall problem structure to be modeled in a more effective and coherent manner.

2 Problem Statement

In this study, a grid-based modeling framework is adopted to represent post-disaster communication operations. The affected area is first partitioned into equal-sized grid cells, each representing a specific geographical region where user demand arises. Accordingly, the problem is formulated over a discrete spatial domain rather than a continuous area, and UAV-based Base Stations (UAV-BSs) are deployed and operated over this grid structure.

In the proposed model, the set of candidate locations for UAV-BSs is denoted by I, the set of UAV-BSs by J, and the set of UAV configurations by K. The UAV-BSs operate over a discrete-time planning horizon $[0, T]$, during which communication services are provided to users located within their coverage areas. Each UAV $j \in J$ is equipped with a limited battery capacity u_j, and must therefore be periodically directed to a charging station to sustain its operations.

UAVs may either relocate among predefined locations to provide service or visit charging stations to recharge. Movement between locations incurs both travel time and energy consumption, causing the battery level of each UAV to evolve dynamically over time. The system operates in discrete time periods.

The primary decisions considered in this study involve selecting the time-dependent UAV-BS configurations and scheduling their charging operations. The objective of the problem is to construct an operational plan that maximizes service coverage while minimizing unmet demand, subject to energy limitations, mobility constraints, and the availability of a single charging station.

The problem is formulated as a mixed-integer optimization model that jointly captures integrated configuration and location decisions, service provision, battery dynamics, and charging processes. The proposed model provides a realistic and practically applicable representation of UAV-based communication operations in post-disaster environments.

3 MILP Formulation

Sets and Indices Candidate locations are indexed by $i \in I$, configurations by $k \in K$, UAV-BSs by $j \in J$, and time periods by $t \in T$.

Parameters R_{kt} denotes the total service level obtained when configuration k is selected at time period t. Additional parameters include (i) the binary parameters that indicate the locations where UAV-BS are deployed for each configuration, (ii) the maximum airborne time of a fully charged UAV-BS expressed in time periods, (iii) distances between locations and the charging station, (iv) distances between any two locations, and (v) energy consumption coefficient associated with UAV-BS movement.

Variables The binary configuration variable z_{kt} indicates whether configuration k is selected at time period t. The binary deployment variables x_{ijt}, y_{it}, and s_{jt} indicate whether UAV-BS j is located at point i at time t, whether at least one UAV-BS is deployed at location i at time t, and whether UAV-BS j is charging at time t, respectively. The binary movement variable $h_{ii'jt}$ indicates whether UAV-BS j moves from location i to location i' at time t. The continuous movement variable v_{jt} represents the total distance traveled by UAV-BS j during time period t. The continuous charging state variable u_{jt} denotes the remaining flight time of UAV-BS j at time t, in terms of time periods.

4 Computational Results

The proposed MILP model is evaluated using representative test scenarios. In both scenarios, the disaster area is modeled as a 10×10 grid, resulting in 100 candidate locations, and a single charging station is available. The planning horizon is set to 10 time periods.

Scenario 1 (Two UAV-BSs). The system consists of one charging station and two UAV-BSs. Based on all feasible combinations of UAV locations and service states, an initial configuration set of 5,051 configurations is generated. To reduce the computational complexity, a period-wise filtering procedure is applied: for each time period, configurations yielding a service level more than 20% below the maximum achievable value are eliminated. This filtering process reduces the number of feasible configurations to 1,942. The model is solved using Gurobi 13.0.0 and attains an objective function value of 1.955 with a 0% optimality gap. The solution time is approximately 20 min (1,261.92 sec).

Scenario 2 (Three UAV-BSs). To examine the impact of fleet size, we consider a second scenario with three UAV-BSs while keeping the grid size, planning horizon, and single charging-station setting unchanged. The initial configuration set contains 166,750 configurations and is reduced to 22,167 feasible configurations after applying the same period-wise 20% filtering rule. The model achieves an objective value of 2.865 with a 0% optimality gap. The total computational time is 21,757.98 sec.

Overall, the results indicate that explicitly modeling energy and charging constraints significantly influences UAV-BS deployment decisions, while the configuration filtering approach plays a key role in maintaining tractable solution times. Although the proposed model performs effectively for the tested instances, the computational results highlight the need for heuristic approaches when scaling to larger grids and UAV-BS fleets.

5 Conclusion

This study proposes a configuration-based MILP formulation for the deployment of UAV-based mobile base stations in post-disaster environments, explicitly incorporating energy limitations and charging operations into the decision-making process. By jointly modeling UAV movement, configuration-related coverage decisions, and energy constraints, the proposed approach provides a realistic representation of post-disaster communication operations.

Computational results demonstrate the effectiveness of the model for small-scale scenarios and highlight the impact of energy constraints on optimal deployment strategies. Future research directions include the development of scalable heuristic methods, the consideration of multiple charging stations, and the incorporation of stochastic demand and dynamic disaster conditions.

Acknowledgement

This research is supported by the Scientific and Technological Research Council of Turkey under grant 124M630.

References

1. E. Kalantari, H. Yanikomeroglu, and A. Yongacoglu. On the number and 3d placement of drone base stations in wireless cellular networks. In *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*, pages 1–6. IEEE, 2016.
2. Y. Park, P. Nielsen, and I. Moon. Unmanned aerial vehicle set covering problem considering fixed-radius coverage constraint. *Computers & Operations Research*, 119:104936, 2020.
3. N. Adam, C. Tapparello, W. Heinzelman, and H. Yanikomeroglu. Placement optimization of multiple uav base stations. In *IEEE Wireless Communications and Networking Conference (WCNC)*, 2021.
4. H. Hydher, D. N. K. Jayakody, K. T. Hemachandra, and T. Samarasinghe. Intelligent uav deployment for a disaster-resilient wireless network. *Sensors*, 20(21):6140, 2020.
5. B. Hu, R. Liu, and L. Zhou. A task-based location deployment method for uav-enabled multi-region networks. *IEEE Access*, 10:10231–10243, 2022.
6. X. Zhang and L. Duan. Energy-saving deployment algorithms of uav swarm for sustainable wireless coverage. *IEEE Transactions on Vehicular Technology*, 69(9):10320–10335, 2020.
7. J. Song, B. Zhang, and J. Li. Deep reinforcement learning empowered particle swarm optimization for aerial base station deployment. In *APSCON 2023 – IEEE Applied Sensing Conference*, 2023.
8. C. T. Cicek, H. Gultekin, B. Tavli, and H. Yanikomeroglu. Uav base station location optimization for next generation wireless networks. In *2019 1st International Conference on Unmanned Vehicle Systems (UVS)*, pages 1–6. IEEE, 2019.
9. Cihan Tugrul Cicek, Hakan Gultekin, Bulent Tavli, and Halim Yanikomeroglu. Backhaul-aware optimization of uav base station location and bandwidth allocation for profit maximization. *arXiv preprint arXiv:1810.12395*, 2018.

Neural Approximate Dynamic Programming for the Ultra-fast Order Dispatching Problem

Arash Dehghan[1], Mucahit Cevik[1*], and Merve Bodur[2]

[1] Department of Mechanical, Industrial and Mechatronics Engineering, Toronto Metropolitan University, Toronto, Ontario M5B 2K3, Canada, arash.dehghan@torontomu.ca, mcevik@torontomu.ca

[2] School of Mathematics and Maxwell Institute for Mathematical Sciences, University of Edinburgh, Edinburgh, UK, merve.bodur@ed.ac.uk

Abstract. Ultra-fast same-day delivery requires real-time dispatching of online orders to a fleet of couriers under delivery deadlines. We study the resulting order dispatching problem for a single depot and propose a Markov Decision Process model that captures key operational features such as order batching and courier assignments. We solve the problem using NeurADP, a hybrid approximate dynamic programming and Deep Reinforcement Learning (DRL) approach, and show in numerical experiments that it consistently outperforms myopic and DRL baselines.

Keywords: Order dispatching· Ultra-fast delivery· NeurADP.

1 Introduction

The widespread adoption of online shopping has pushed retailers toward faster, more direct fulfillment models, with same-day delivery emerging as a key service offering. As same-day delivery has scaled, centralized fulfillment facilities (e.g., warehouses) have become operational hubs that receive orders and dispatch courier fleets to meet service targets. Ensuring high operational efficiency in these systems requires coordinating courier shifts, vehicle capacities, and routing/dispatch decisions under uncertainty in order timing, volume, locations, and delivery deadlines; these challenges are commonly studied under the Same-Day Delivery Problem (SDDP) framework, which is often decomposed into a routing component and a real-time matching/dispatch component (ODP).

In this paper, we focus on the *ultra-fast ODP* arising in Quick Commerce (Q-commerce), where small-format vehicles (e.g., e-bikes, mopeds) deliver small batches under *hard* delivery deadlines. We consider a single-depot setting in which a central dispatcher dynamically assigns and dispatches a fleet of couriers with heterogeneous shift schedules to maximize the number of orders served over the day, given stochastic arrivals and stringent time constraints. This delivery requirement distinguishes Q-commerce dispatching from much of the SDDP/ODP literature, which typically assumes more permissive delivery windows, and remains comparatively underexplored despite its rapid adoption in practice. Our work builds on the single-depot ODP studied by Kavuk et al. [1], who use deep reinforcement learning (DRL) to make accept/reject decisions while relying on predetermined heuristics for courier assignment.

Our main contributions are threefold. First, we formulate a novel Markov Decision Process (MDP) model for the single-depot ultra-fast ODP that explicitly captures operational features central to practice, including (i) batching with courier queues, (ii) explicit order–courier assignments, and (iii) hard delivery deadlines, yielding a realistic yet tractable model for learning-based control. Second, we solve this problem using Neural Approximate Dynamic Programming (NeurADP) [2], which combines ADP with a neural value function approximation to handle high-dimensional, one-to-many matching decisions; to the best of our knowledge, this is the first application of NeurADP outside of ride-pooling problems. Third, we provide an extensive computational study comparing NeurADP with several baselines and performing sensitivity analyses over key operational factors (e.g., fleet size and vehicle capacity), complemented by multiple datasets to support managerial insights.

2 Model Formulation and Solution Methodology

Ultra-fast delivery platforms must dynamically match a stream of incoming orders to a fixed fleet of heterogeneous couriers dispatched from a single depot, with the primary goal of maximizing the number of orders successfully delivered within strict, arrival-relative deadlines, subject to shift, capacity, and feasibility constraints. Decisions are made at frequent epochs over a one-day horizon: multiple orders may arrive per epoch, and any order not accepted and assigned in its arrival epoch is dropped, reflecting the need for immediate customer confirmation; once assigned, an order is served by that courier and cannot be reassigned. Couriers have predefined shift start times and finite working durations, may be available at the depot, in the field, or temporarily on break, and are eligible for assignment only if they have remaining vehicle capacity and can complete their entire queue before shift end and before any order deadline. At each epoch, the platform jointly decides which orders to assign to which couriers and whether to dispatch depot-based couriers immediately; assigned orders are sequenced into a depot-starting-and-ending tour to satisfy deadlines and shifts. Field couriers may receive additional orders into their existing queues when feasible; upon returning to the depot, their queues clear and, if still eligible and new orders exist, they are dispatched again. Travel times enter either directly in the objective or implicitly through feasibility checks that govern assignment and routing decisions.

In our MDP model, the planning horizon is discretized into decision epochs $\mathcal{T} = \{0, \dots, T\}$ of length δ, where decisions are taken at the start of each epoch and exogenous information (new orders) arrives between epochs. The system state at epoch t is $S_t = (C_t, O_t)$, comprising (i) the status of all couriers and (ii) the set of newly arrived, yet-unassigned orders. A courier is represented by $c = (c_{\texttt{shift}}, c_{\texttt{ret}}, c_{\texttt{ords}})$, encoding shift start time, remaining time until return to the depot, and its queue of assigned orders; each order is $o = (o_{\texttt{dest}}, o_{\texttt{dead}})$, with deadline determined upon arrival. At each epoch, the action $\mathbf{a}_t \in \mathbf{A}_t(S_t)$ jointly selects (a) feasible batching of current orders and (b) explicit assignment of batches to eligible couriers, subject to capacity limits, deadline feasibility, and completion before shift end; assigned orders are sequenced into a feasible tour starting/ending at the depot using travel times $\texttt{time}_t(\ell, \ell')$. The reward

scalarizes a lexicographic objective by prioritizing fulfillment and then speed:

$$R_t(\mathbf{a}_t) = \sum_{c \in C_t} \Big(\beta\, q_t(\mathbf{a}_{tc}) - \omega_t(\mathbf{a}_{tc}) \Big), \tag{1}$$

where $q_t(\cdot)$ counts the number of orders served due to decision at t for courier c, and $\omega_t(\cdot)$ is the induced route duration; β is set large enough that serving more orders always dominates any delay reduction. The post-decision state $S_t^{\texttt{Courier-Post}} = \texttt{statepost}(S_t, \mathbf{a}_t)$ captures courier updates immediately after implementing $\mathbf{a}_t$, while the next pre-decision state is obtained by adding exogenous arrivals W_{t+1} via $S_{t+1} = \texttt{statenext}(S_t^{\texttt{Courier-Post}}, W_{t+1})$, with $O_{t+1} = W_{t+1}$ and $C_{t+1} = S_t^{\texttt{Courier-Post}}$. The objective is to find a policy maximizing expected cumulative reward over the horizon. Exact dynamic programming is computationally intractable due to the scale of the state, action, and outcome spaces, motivating an ADP approach with neural value-function approximation. Specifically, NeurADP learns a nonlinear approximation of the post-decision value function with a neural network and embeds that learned value inside a structured optimization step, yielding coordinated decisions over all couriers at each epoch.

At each decision epoch t, NeurADP applies a two-step decomposition. First, it enumerates the feasible courier–batch actions by generating batches B_t (batch size between 1 and $\texttt{queue}_{\texttt{max}}$) and retaining only those pairs (i, b) for which adding batch b to courier i preserves queue-capacity feasibility and admits a route that delivers all queued and newly added orders before their deadlines and before shift end. Let F_t denote the resulting feasible set, and $\varnothing$ denote the action of assigning no new orders. Second, it solves a matching integer program that assigns each courier exactly one action while ensuring each order is assigned at most once:

$$\texttt{MatchingIP:} \quad \max \sum_{i \in C_t} \sum_{a \in \mathcal{A}_{t,i}} \Big(r_{tia} + \texttt{score}_t(i \leftrightarrow a) \Big)\, x_{tia} \tag{2a}$$

$$\text{s.t.} \quad \sum_{a \in \mathcal{A}_{t,i}} x_{tia} = 1 \qquad \forall i \in C_t \tag{2b}$$

$$\sum_{i \in C_t} \sum_{b \in B_t:\ (i,b) \in F_t,\ o \in b} x_{tib} \le 1 \qquad \forall o \in O_t \tag{2c}$$

$$x_{tia} \in \{0, 1\} \qquad \forall i \in C_t,\ a \in \mathcal{A}_{t,i}, \tag{2d}$$

where $\mathcal{A}_{t,i} := \{b \in B_t : (i, b) \in F_t\} \cup \{\varnothing\}$ and r_{tia} implements the lexicographic operational objective. The term $\texttt{score}_t(i \leftrightarrow a)$ is the learned estimate of downstream value for choosing action a for courier i at epoch t: we construct the corresponding post-decision state induced by (i, a), evaluate it with the neural value function, and treat that value as a constant in (2a). The neural value function is trained by minimizing squared Bellman error using transitions collected from simulated rollouts (with replay to reuse experience and target-network style stabilization), and its per-courier evaluation makes scoring scale linearly in the number of couriers even though `MatchingIP` produces a globally coordinated assignment. In numerical experiments, we use a 24-hour horizon with 5-minute decision epochs and a *fixed* delivery-window model (deadlines defined as arrival time plus a constant allowance $\tau_{\texttt{max}}$).

3 Results and Discussion

We evaluate `NeurADP` against two benchmark classes that reflect common approaches to ultra-fast dispatching. The `Myopic` policies are greedy batching-and-assignment rules without learning or lookahead. The `DRL` policies (from [1]) use a learned accept/reject module for incoming orders and subsequently rely on a heuristic matcher to allocate accepted orders to couriers. Both classes are paired with four matching heuristics: `DC` (Distance-Closest), `DF` (Distance-Farthest), `CE` (Capacity-Emptiest), and `CF` (Capacity-Fullest). To contextualize achievable performance, we employ a reference ceiling, which trains `NeurADP` directly on each test day (i.e., an anticipative upper bound under perfect knowledge).

Table 1 presents results under the baseline configuration. We normalize performance using the fixed-ceiling reference given by the average number of orders fulfilled by `NeurADP` under `NeurADP-Fixed`; the "% Filled" column reports each policy's fulfillment as a percentage of this reference (mean±stdev), and "% Incr. NeurADP" reports `NeurADP`'s relative improvement over each benchmark. `NeurADP` achieves near-ceiling performance (99.19%±0.62) and dominates all benchmark variants. The strongest heuristic baseline, `Myopic-DC`, reaches 94.35% ± 1.08, leaving a 4.84% gap, while other myopic variants trail further. The `DRL` baselines perform substantially worse (approximately 83–88% filled), with `DRL-DC` the best among them yet still 11.13% below `NeurADP`.

Table 1: Performance of dispatching policies.

Policy	**NeurADP-Fixed Ceiling**	**% Filled**	**% Incr. NeurADP**
`NeurADP`	819.20	99.19 ± 0.62	-
`DRL-DC`	-	88.06 ± 0.44	+11.13
`DRL-DF`	-	83.30 ± 0.73	+15.89
`DRL-CE`	-	83.52 ± 0.91	+15.67
`DRL-CF`	-	86.10 ± 1.07	+13.09
`Myopic-DC`	-	94.35 ± 1.08	+4.84
`Myopic-DF`	-	89.59 ± 1.13	+9.60
`Myopic-CE`	-	92.44 ± 1.45	+6.75
`Myopic-CF`	-	92.30 ± 1.09	+6.89

Overall, these sample results indicate that combining a learned downstream value estimate with a globally coordinated matching optimization yields consistent gains over both greedy batching heuristics and DRL policies that depend on simpler, sequential decision logic.

Disclosure Statement The authors have no competing interests.

References

1. Kavuk, E. M., A. Tosun, M. Cevik, and others. Order dispatching for an ultra-fast delivery service via deep reinforcement learning. *Applied Intelligence 52*, 1–26.
2. Shah, S., M. Lowalekar, and P. Varakantham (2020). Neural approximate dynamic programming for on-demand ride-pooling. In *AAAI Conference*.

Improving rank inequalities for covering and packing problems

Leonardo de Abreu[1][0000-0003-0701-0853] and
Manoel Campêlo[2][0000-0003-0701-0853]

[1] Graduate Program in Computer Science, Federal University of Ceará,
Fortaleza CE 60440-900, Brazil.
leonardo.abreu@lia.ufc.br
https://mdcc.ufc.br
[2] Statistics and Applied Mathematics Department, Federal University of Ceará,
Fortaleza CE 60440-900, Brazil
mcampelo@ufc.br

Abstract. The set covering polytope is defined as the convex hull of characteristic vectors of solutions to Set Cover. Set covering problems are related to clutters, that is, collections of inclusion-wise minimal subsets of a set. We extend the study of $\mathcal{N}$-set inequalities, a generalization of rank inequalities, to these problems. We define two classes of inequalities grounded on transversals and independent sets, and provide sufficient conditions for them to be facet-defining. We also describe lifting procedures based on clutter minors. In addition, we present the counterpart of one of these two classes of inequalities for set packing problems.

Keywords: Combinatorial optimization · Integer program · Facets of polyhedra · Graph theory.

1 Introduction

A *clutter* (or *Sperner family*) $\mathscr{C}$ consists of an ordered pair $(V(\mathscr{C}), E(\mathscr{C}))$, where E is a set of minimal subsets of V (no member of E is contained in another). The sets V and E are, respectively, the sets of *vertices* and *edges* of $\mathscr{C}$.

A *transversal* is a set of vertices that intersects all edges of $\mathscr{C}$, while a *matching* is a set of pair-wise disjoint edges of $\mathscr{C}$. Moreover, a set of vertices is *independent* if it contains no edge of $\mathscr{C}$.

Denote by $\mathbf{M}(\mathscr{C})$ the $\{0,1\}$-matrix $(m_{ij})_{|E|\times|V|}$ whose columns are indexed by $V(\mathscr{C})$ and whose rows are characteristic vectors of the edges of $\mathscr{C}$, i.e., $m_{ij} = 1$ if $j \in e_i$, and $m_{ij} = 0$ otherwise. Then, consider the following polyhedra:

$$P(\mathscr{C}) = \{\mathbf{x} \in \mathbb{R}^n_+ : \mathbf{M}(\mathscr{C})\ \mathbf{x} \geq \mathbf{e},\ \mathbf{x} \leq \mathbf{e}\},$$
$$Q(\mathscr{C}) = \{\mathbf{x} \in \mathbb{R}^n_+ : \mathbf{M}(\mathscr{C})\ \mathbf{x} \leq \mathbf{e},\ \mathbf{x} \leq \mathbf{e}\},$$

where $n = |V|$ and $\mathbf{e}$ stands for the vector of ones, i.e., $\mathbf{e} = (1, 1, \ldots, 1)$. Note that the Set Cover problem can be stated as $\min\{\mathbf{w}^{\mathrm{t}}\,\mathbf{x} : \mathbf{x} \in P(\mathscr{C}) \cap \mathbb{Z}^n\}$. Analogously, Set Packing can be stated as $\max\{\mathbf{w}^{\mathrm{t}}\,\mathbf{x} : \mathbf{x} \in Q(\mathscr{C}) \cap \mathbb{Z}^n\}$. We are

interested in the facets of the polytopes $P_I(\mathscr{C}) = \text{conv}(P(\mathscr{C}) \cap \mathbb{Z}^n)$ and $Q_I(\mathscr{C}) = \text{conv}(Q(\mathscr{C}) \cap \mathbb{Z}^n)$, specially the ones that are derived from clutter minors.

A minor of a clutter $\mathscr{C}$ is the clutter obtained by deleting and contracting vertices [3]. Contracting a vertex $i \in V(\mathscr{C})$ is removing i from all the edges that include i, and then discarding the edges that are no longer minimal. This operation results in a clutter, denoted by $\mathscr{C}/i$, that has $V(\mathscr{C}) \setminus \{i\}$ as vertex set, and the minimal elements of $\{S \setminus \{i\} : S \in E(\mathscr{C})\}$ as edge set. Moreover, deleting a vertex i is removing all edges that include i. The resulting clutter, denoted by $\mathscr{C} \setminus i$, also has $V(\mathscr{C}) \setminus \{i\}$ as vertex set, but its edge set is $\{S \in E(\mathscr{C}) : i \notin S\}$.

This work extends the study of the facets of $P_I(\mathscr{C})$ and $Q_I(\mathscr{C})$. Specifically, it generalizes $\mathscr{N}$-set inequalities for set covering and set packing problems. These inequalities are derived from a strategy used by [2] to find facet-defining structures for a polytope associated to a formulation of the vertex covering problem. The name was coined by [5] when studying the 2-class single group classification problem, and later used by [1] for the geodesic classification problem. We adapt such inequalities to general set covering problems, introducing $\mathscr{T}$-set and $\mathscr{S}$-set inequalities. We also presents sufficient conditions for them to be facet-defining for $P_I(\mathscr{C})$, and explores analogous results for $Q_I(\mathscr{C})$. Finally, we present a framework for lifting variables based on clutter minors.

2 Results

Let $\mathscr{C}$ be a clutter and let (SC) be the covering problem defined by $\mathscr{C}$. If $\tau(\mathscr{C})$ is the smallest size of a transversal, the inequality $\sum_{v \in V(\mathscr{C})} \mathbf{x}_v \geq \tau(\mathscr{C})$ is clearly valid for $P_I(\mathscr{C})$. This type of inequality is called *rank inequality*. Rank inequalities can be strengthened as follows.

For each vertex v of $\mathscr{C}$, denote by τ_v the minimum size of a minimal transversal of $\mathscr{C}$ that includes v, that is, $\tau_v := \min\{|W| : v \in W,\ W \in b(\mathscr{C})\}$. For simplicity, consider $\min(\emptyset) = +\infty$ and $\frac{1}{\tau_v} = 0$ if $\tau_v = +\infty$. Therefore, the inequality

$$\sum_{v \in V(\mathscr{C})} \frac{1}{\tau_v} \mathbf{x}_v \geq 1 \tag{1}$$

is valid for $P_I(\mathscr{C})$. Indeed, recall that a solution to (SC) is a characteristic vector $\mathbf{x}^W$ of a transversal $W \in b(\mathscr{C})$. The size of a transversal W of $\mathscr{C}$ must be at least τ_v for all $v \in W$ such that $\tau_v \neq +\infty$. Thus, if $\tau_{max} = \max_{v \in W}\{\tau_v : \tau_v < +\infty\}$, then $\sum_{v \in V(\mathscr{C})} \frac{1}{\tau_v} \mathbf{x}^W_v = \sum_{\substack{v \in W \\ \tau_v < +\infty}} \frac{1}{\tau_v} \geq \frac{|W|}{\tau_{max}} \geq 1$.

As aforementioned, (1) dominates the rank inequality, which can be rewritten as $\sum_{v \in V(\mathscr{C})} \frac{1}{\tau(\mathscr{C})} \mathbf{x}_v \geq 1$. This fact follows straightforwardly from $\tau_v \geq \tau(\mathscr{C})$, for all $v \in V(\mathscr{C})$. Inequalities with the same rationale as (1) appear in [4,5] and [1] for Euclidean and graph classification problems. However, the inequalities presented in these references are specific for the approached problems instead of arbitrary covering problems. Actually, only the formulation in [1] explicitly has the structure of a set covering problem.

An important notion in such works is that of $\mathcal{N}$-sets. Although it was not put that way, they can be seen as minimal transversals for the specific problem addressed in each paper [5,1]. To better characterize inequalities of the type (1) in a more general setting, we update this concept by introducing the definition of $\mathcal{T}$-sets. In the sequel, Theorem 1 generalizes the existing results for any covering problem. It also apparently weakens one of the sufficient conditions — condition (*ii*) below — for the inequality to be facet defining.

The blocker $b(\mathscr{C})$ of a clutter $\mathscr{C}$ is the clutter whose vertex set is the same of $\mathscr{C}$ and whose edges are the minimal transversals of $\mathscr{C}$. Let $\mathscr{H} = \mathscr{C} \setminus U$, for some $U \subset V(\mathscr{C})$ and $\tau_v^{\mathscr{H}} = \min\{|W| : v \in W,\ W \in b(\mathscr{H})\}$, for all $v \in V(\mathscr{H})$.

A $\mathcal{T}$-set of $\mathscr{H}$ is a transversal W of $\mathscr{H}$ where if $\tau_v^{\mathscr{H}} = |W|$, for all $v \in W$. The *$\mathcal{T}$-safe graph* of $\mathscr{H}$ is a simple graph, say H^{safe}, whose vertex set is $V(\mathscr{H})$ and $uv \in E(H^{\text{safe}})$ if and only if there exist two $\mathcal{T}$-sets W_u and W_v of $\mathscr{H}$, such that $W_u \setminus W_v = \{u\}$ and $W_v \setminus W_u = \{v\}$.

Theorem 1. *Suppose that $|V(\mathscr{H})| \geq 2$. The inequality*

$$\sum_{v \in V(\mathscr{H})} \frac{1}{\tau_v^{\mathscr{H}}} \mathbf{x}_v \geq 1 \tag{2}$$

is valid for $P_I(\mathscr{C})$ and is facet-defining if the following conditions hold:

i) For all $v \in U$, there is a $\mathcal{T}$-set that is also a transversal of $\mathscr{C} \setminus (U \setminus \{v\})$.
ii) Every connected component of H^{safe} contains a $\mathcal{T}$-set.

It is not clear whether the reciprocal of Theorem 1 is true for some class of clutters. However, if $\mathscr{C}$ has no edge with only one vertex, condition (*i*) is necessary. Besides, we can show a $\mathcal{T}$-set inequality is facet defining for $P_I(\mathscr{C})$ if and only if each part of the inequality with equal coefficients yields a (rank) facet-defining inequality for the minor restricted to the corresponding vertices.

The next theorem extends the result by [2] that inspired Theorem 1. Again, it is established in the (more general) context of clutters. One can use this theorem to provide an alternative proof for the result that inspired it.

Consider again a minor $\mathscr{H} = \mathscr{C} \setminus U$ for some $U \subseteq V(\mathscr{C})$. For each $v \in V(\mathscr{H})$, let $\alpha_v^{\mathscr{H}} = \max\{|S| : v \in S,\ E(\mathscr{H} \setminus \overline{S}) = \emptyset\}$, where $\overline{S} = V(\mathscr{H}) \setminus S$.

An $\mathcal{S}$-set of $\mathscr{H}$ is an independent set S of $\mathscr{H}$ where $\alpha_v^{\mathscr{H}} = |S|$, for all $v \in V(\mathscr{H})$. The *$\mathcal{S}$-safe graph* of $\mathscr{H}$ is a simple graph, say $\hat{H}^{\text{safe}}$, whose vertex set is $V(\mathscr{H})$ and $uv \in E(\hat{H}^{\text{safe}})$ if and only if there exist two $\mathcal{S}$-sets S_u and S_v, such that $S_u \setminus S_v = \{u\}$ and $S_v \setminus S_u = \{v\}$. Then, Theorem 2 holds:

Theorem 2. *Suppose that $|V(\mathscr{H})| \geq 2$. The inequality*

$$\sum_{v \in V(\mathscr{H})} \frac{1}{\alpha_v^{\mathscr{H}}} \mathbf{x}_v \geq \sum_{v \in V(\mathscr{H})} \frac{1}{\alpha_v^{\mathscr{H}}} - 1 \tag{3}$$

is valid for $P_I(\mathscr{C})$ and is facet-defining if the following conditions hold:

i) For all $v \in U$, there is an $\mathcal{S}$-set S such that $S \cup \{v\}$ is independent for $\mathscr{C}$.
ii) Every connected component of $\hat{H}^{\text{safe}}$ contains an $\mathcal{S}$-set.

Although inequalities (2) and (3) come from the same type of rank inequality, they are not equivalent in most cases. In fact, we show that they are equivalent for $P_I(\mathscr{C})$ if and only if there exist positive integers α' and τ', and a minimum

size transversal W of $\mathscr{H}$ such that $\frac{\tau(\mathscr{H})}{\alpha'} = \frac{\tau'}{\alpha(\mathscr{H})}$ and $\alpha_u^H = \alpha'$ for all $u \in W$ with $\{u\} \notin E(\mathscr{C})$, and $\tau_u^{\mathscr{H}} = \tau'$ for all $u \in V(\mathscr{H}) \setminus W$.

Similar results hold for set packing problems. The anti-blocker $a(\mathscr{C})$ of a clutter $\mathscr{C}$ is the clutter whose vertex set is the same as $\mathscr{C}$ and whose edges are the maximal sets such that no pair of vertices belong together to an edge of $\mathscr{C}$. Consider a minor $\mathscr{H}$ of $\mathscr{C}$ obtained by *contracting* a subset of vertex $U \subseteq V(\mathscr{C})$. Let $\dot{\alpha}_v^{\mathscr{H}} = \max\{|S| : v \in S,\ S \in E(a(\mathscr{H}))\}$, for all $v \in V(\mathscr{H})$.

An α-set of $\mathscr{H}$ is an edge $S \in E(a(\mathscr{H}))$ such that $\dot{\alpha}_v^{\mathscr{H}} = |S|$, for all $v \in S$. Let $\tilde{H}^{\text{safe}}$ be a simple graph whose vertex set is $V(\mathscr{H})$ and $uv \in E(\tilde{H}^{\text{safe}})$ if and only if there exist two α-sets S_u and S_v, such that $S_u \setminus S_v = \{u\}$ and $S_v \setminus S_u = \{v\}$.

Theorem 3. *Suppose that* $|V(\mathscr{H})| \geq 2$. *The inequality*

$$\sum_{v \in V(\mathscr{H})} \frac{1}{\dot{\alpha}_v^{\mathscr{H}}} \mathbf{x}_v \leq 1$$

is valid for $Q_I(\mathscr{C})$ *and is facet-defining if the following conditions hold:*

i) For each $v \in U$, *there is an* α*-set* S *such that* $v \in e \in E(\mathscr{C})$ *implies* $e \cap S = \emptyset$.

ii) Every connected component of $\tilde{H}^{\text{safe}}$ *contains an* α*-set.*

Procedures for lifting variables based on clutter minors were developed, but are omitted from this abstract due to space limitations.

3 Conclusion

The $\mathscr{T}$-set inequalities and $\mathscr{S}$-set inequalities introduced in this work dominate rank inequalities, but are strongly related to rank inequalities of clutter minors. We prove that they are distinct classes of inequalities and identity the cases where they coincide. Our results are based on transversals and independent sets of clutter minors, and unify and extend those of [4,5,1] to a more general setting.

Acknowledgments. Partially supported by Funcap ITR-0214-00096.01.00/23, CNPq 312417/2022-5, 442977/2023-9 and 404479/2023-5.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Araújo, P.H., Campêlo, M., Corrêa, R.C., Labbé, M.: Integer programming models and polyhedral study for the geodesic classification problem on graphs. European Journal of Operational Research **314**(3), 894–911 (2024)
2. Campêlo, M., Campos, V.A., Corrêa, R.C.: On the asymmetric representatives formulation for the vertex coloring problem. DAM **156**(7), 1097–1111 (2008)
3. Cornuéjols, G.: Combinatorial Optimization: Packing and Covering. SIAM, Philadelphia (2001)
4. Corrêa, R.C., Blaum, M., Marenco, J., Koch, I., Mydlarz, M.: An Integer Programming Approach for the 2-class Single-group Classification Problem. Electronic Notes in Theoretical Computer Science **346**, 321–331 (Aug 2019)
5. Corrêa, R.C., Delle Donne, D., Marenco, J.: On the combinatorics of the 2-class classification problem. Discrete Optimization **31**, 40–55 (2019)

An Exact Method for Fair Influence Maximization

Kübra Tanınmış and Meryem Şırnak

Koç University, İstanbul, Türkiye
{ktaninmis,msirnak25}@ku.edu

Abstract. Influence Maximization Problem aims at identifying a small set of seed nodes to maximize the expected spread of information on a social network. When the community structure inherent in a social network is overlooked, the optimal seed set often results in a highly imbalanced spread across different groups. This imbalance is undesirable in scenarios involving socially beneficial information, such as job postings and public health outreach campaigns. In contrast to the heuristic approaches commonly discussed in the literature, our objective is to determine the optimal seed set that meets a specified fairness criterion and to assess the price-of-fairness in a more reliable way. To achieve this, we introduce a mixed-integer programming model into which established fairness principles can be directly incorporated. We develop a unified Benders decomposition framework for several fairness principles to address the scalability challenges. Preliminary results demonstrate that the proposed decomposition method significantly outperforms standard commercial solvers.

Keywords: Influence Spread · Mixed-integer programming · Benders Decomposition.

1 Introduction and Problem Definition

Influence Maximization (IM) in social networks is a stochastic combinatorial optimization problem aimed at identifying a small set of *seed* nodes to maximize the spread of information, innovation, or influence. First formulated by Kempe et al. [3] as a discrete stochastic optimization problem, the IMP has vast applications ranging from viral marketing to public health awareness campaigns.

The classical IMP problem focuses solely on maximizing the total expected spread, often represented as a monotonically non-decreasing and submodular function under stochastic diffusion models such as Independent Cascade (IC) or Linear Threshold (LT). However, solutions driven purely by the goal of spread maximization favor densely connected large communities while marginalizing smaller or peripheral groups. In socially sensitive applications such as distributing critical health information or job opportunities, this structural bias leads to significant unfairness. To address this, the Fair Influence Maximization problem (FIMP) introduces fairness constraints such as proportional seed selection,

or proportional final spread, or objectives such as maximizing spread in the minimum affected group, to ensure an equitable distribution of influence across known communities.

While heuristic approaches for the FIMP exist, they lack optimality guarantees, which limits the precise assessment of the *Price of Fairness* (PoF) – the loss in total efficiency incurred to achieve fairness. This study addresses this gap by proposing an exact solution methodology based on Benders Decomposition for the Fair Influence Maximization problem under three well established principles: Rawls' maximin principle, equality, and demographic parity.

2 A Group-based Decomposition Approach

We model the diffusion process using the *live-edge* graph representation, which transforms the stochastic diffusion into a reachability problem over a set of sampled scenarios Ω, and generalizes the well known IC and LT models [3]. Let $G = (V, E)$ be the social network and $\mathcal{C} = \{C_1, \ldots, C_M\}$ be the given set of disjoint groups. Let R_i^ω represent the set of nodes that can reach (influence) node i in scenario ω. We start by formulating the classical IMP as a mixed-integer program where the group spread level is explicitly tracked, so that a group-based fairness criterion can be directly integrated into the model.

Let z_i be a binary variable equal to 1 if node i is selected as a seed, and 0 otherwise. Let x_i^ω indicate whether node i is activated in scenario $\omega \in \Omega$, and let θ_c be the number of activated nodes of group c. The deterministic equivalent formulation of the IMP, is given below.

$$\max \sum_{c \in C} \theta_c \tag{1a}$$

$$s.t. \sum_{i \in V} z_i \leq k \tag{1b}$$

$$x_i^\omega \leq \sum_{j \in R_i^\omega} z_j \qquad \forall i \in V, \forall \omega \in \Omega \tag{1c}$$

$$x_i^\omega \geq z_j \qquad \forall i \in V, \forall \omega \in \Omega, \forall j \in R_i^\omega \tag{1d}$$

$$\theta_c = \frac{1}{|\Omega|} \sum_{i \in V_c} \sum_{\omega \in \Omega} x_i^\omega \qquad \forall c \in C \tag{1e}$$

$$z_i \in \{0, 1\} \qquad \forall i \in V \tag{1f}$$

$$0 \leq x_i^\omega \leq 1 \qquad \forall i \in V, \forall \omega \in \Omega \tag{1g}$$

$$\theta_c \geq 0 \qquad \forall c \in C \tag{1h}$$

In this model, the objective is to maximize the average spread across all scenarios. (1b) limits the number of seed nodes. (1c) and (1d) ensure that node i is activated in scenario ω only if at least one seed node is selected from its reachability set R_i^ω. (1e) computes the group spread levels and finally, (1f), (1g), and (1h) define the variable domains. This model differs from existing IMP formulations, such as the one in [2], in two main aspects: it explicitly computes group

spread levels, and it enforces lower bounds on the x variables, ensuring that the model's correctness is independent of the chosen objective function. For each fairness principle considered in this study, we express the corresponding FIMP as a mixed-integer program (MIP), as described below.

The first principle, known as *maximin spread*, is the most widely used notion of fairness in influence spread [1, 6]. Based on Rawls' theory of justice, this principle maximizes the coverage (spread ratio) of the least-affected group. Next, we consider the *equality* principle, under which the number of seed nodes selected from each group must be proportional to the group size [4]. While the objective remains maximizing expected spread, a feasible seed set must satisfy parity constraints in addition to the budget constraint. The last fairness principle we adopt is *demographic parity*. This principle maximizes spread while bounding coverage differences between groups by ϵ [5]. Sufficiently small ϵ enforces equal coverage across all groups, ensuring that group spread is proportional to group size (also called *equity*). The resulting formulations are presented below.

$$\max\{\mu \mid \mu \leq \frac{\theta_c}{|V_c|} \, \forall c \in C, (1b)\text{–}(1h)\} \tag{Maximin}$$

$$\max\{\mu \mid \mu = \sum_{c \in C} \theta_c, \sum_{i \in V_c} z_i \geq k_c, \, \forall c \in C, (1b)\text{–}(1h)\} \tag{Equality}$$

$$\max\{\mu \mid \mu = \sum_{c \in C} \theta_c, \left| \frac{\theta_{c_1}}{|V_{c_1}|} - \frac{\theta_{c_2}}{|V_{c_2}|} \right| \leq \epsilon, \forall c_1, c_2 \in C, (1b)\text{–}(1h)\} \tag{Dem.Parity}$$

These formulations exhibit a block-angular structure, making them well suited for application of Benders decomposition. We therefore introduce a unified Benders Decomposition framework in which the master problem (MP) involves the seed selection ($\mathbf{z}$) and the estimation of group spread levels θ_c. Given a seed set and spread estimates, the subproblem (SP) decomposes into independent problems—one per group—each requiring evaluation of group spread. It becomes infeasible if the group spread level is wrongly estimated in the MP.

$$\begin{aligned} \max \quad & \mu \\ s.t. \quad & \text{Fairness constraints} \\ & \text{Benders feasibility cuts} \\ & \text{(1b),(1f),(1h).} \end{aligned} \tag{MP}$$

$$\begin{aligned} \max \; & 0 \\ \text{s.t.} \; & \frac{1}{|\Omega|} \sum_{i \in V_c} \sum_{\omega \in \Omega} x_i^\omega = \bar{\theta}_c \quad (\gamma) \\ & x_i^\omega \leq \sum_{j \in R_i^\omega} \bar{z}_j \quad \forall i \in V_c, \; \forall \omega \in \Omega \quad (\beta_i^\omega) \\ & x_i^\omega \geq \bar{z}_j \quad \forall i \in V_c, \; \forall \omega \in \Omega, \; \forall j \in R_i^\omega \quad (\sigma_{ij}^\omega) \\ & 0 \leq x_i^\omega \leq 1 \quad \forall i \in V_c, \; \forall \omega \in \Omega \quad (\alpha_i^\omega) \end{aligned} \tag{$\text{SP}_c(\bar{z}, \bar{\theta})$}$$

As a first approach, we solve the FIMP via branch-and-cut, solving the dual of the subproblem at each candidate solution and adding feasibility cuts whenever the dual is unbounded. Preliminary experiments were conducted on synthetic networks to evaluate the performance of the proposed approach. Table 1 presents the results for the maximin spread principle, using networks with varying sizes ($|V|$) and densities ($|E|$). The results show that proposed Benders Decomposition provides exact solutions up to 100 times faster. Furthermore, the PoF values indicate that fairness can be achieved with a relatively small sacrifice in total spread (approx. 2-7%), justifying the use of fair optimization models in practice.

Table 1. Comparison of IMP, Maximin FIMP (Compact), and Benders Decomposition.

$\|V\|$	$\|E\|$	m	Max. spread	Fair spread	PoF (%)	μ	t-IMP	t-FIMP	t-Ben
100	200	8.91	43.23	40.26	6.88	0.35	0.83	82.15	0.63
100	400	7.20	72.49	70.19	3.18	0.69	0.74	23.29	1.62
100	600	6.80	85.00	83.02	2.33	0.84	0.50	19.29	2.66
200	400	10.80	78.64	74.02	5.88	0.34	4.23	107.80	3.38
200	800	10.00	136.56	132.55	2.94	0.66	4.22	131.02	7.00
200	1200	9.00	163.93	159.63	2.62	0.80	4.05	113.20	12.00

(Each number is an average over 10 random networks. m: no. of groups, Max. spread: IMP objective, Fair spread: total spread under maximin objective, PoF (%) decrease in total spread, μ: minimum group coverage under maximin objective, t-IMP (t-FIMP): IMP (FIMP) compact formulation solution time in seconds. t-Ben: Benders algorithm time.)

Ongoing work indicates that incorporating the structure induced by the fairness criteria within the same framework can substantially improve algorithmic efficiency, via efficient initial solution and cut generation strategies.

Disclosure of Interests. The authors have no competing interests to declare.

References

1. Fish, B., Bashardoust, A., Boyd, D., Friedler, S., Scheidegger, C., Venkatasubramanian, S.: Gaps in information access in social networks? In: The world wide web conference. pp. 480–490 (2019)
2. Güney, E., Leitner, M., Ruthmair, M., Sinnl, M.: Large-scale influence maximization via maximal covering location. European Journal of Operational Research **289**(1), 144–164 (2021)
3. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 137–146 (2003)
4. Stoica, A.A., Chaintreau, A.: Fairness in social influence maximization. In: Companion proceedings of the 2019 world wide web conference. pp. 569–574 (2019)
5. Stoica, A.A., Han, J.X., Chaintreau, A.: Seeding network influence in biased networks and the benefits of diversity. In: Proceedings of The Web Conference 2020. pp. 2089–2098 (2020)
6. Tsang, A., Wilder, B., Rice, E., Tambe, M., Zick, Y.: Group-fairness in influence maximization. arXiv preprint arXiv:1903.00967 (2019)

Network Congestion Modeling under Demand Uncertainty via Wardrop Equilibria

Yasmine Beck[1], Francesca Giancola[2,3], Ivana Ljubić[4], and Sara Mattia[3]

[1] Eindhoven University of Technology, Eindhoven, The Netherlands y.beck@tue.nl
[2] DIAG, Sapienza Università di Roma, Roma, Italy
francesca.giancola@uniroma1.it
[3] IASI-CNR, Rome, Italy {francesca.giancola,sara.mattia}@iasi.cnr.it
[4] ESSEC Business School, Cergy-Pontoise, France ljubic@essec.edu

Abstract. This work addresses the challenge of evaluating worst-case traffic congestion in multi-commodity networks when travel demand is uncertain. The methodology combines robust optimization for handling demand variability with Wardrop equilibrium principles—both user equilibrium and system optimum—to represent traveler behavior. The resulting bilevel structure is reformulated into tractable mixed-integer nonlinear programs through KKT-based transformations. We conduct an extensive computational study on benchmark networks, analyzing the impact of different modeling choices including travel cost functions, congestion measures, and uncertainty set geometries.

Keywords: Traffic equilibrium · Robust optimization · Bilevel programming · Network congestion

1 Introduction

This work investigates the problem of determining maximum congestion levels in multi-commodity traffic networks subject to demand uncertainty, stress-testing infrastructure by identifying adverse scenarios.

We formulate this problem using a novel bilevel optimization framework where an evaluator (acting as the leader) selects worst-case demand scenarios while network users (the followers) determine their routes according to established traffic equilibrium principles. For the follower problem, we consider both the Wardrop user equilibrium—where travelers selfishly minimize individual travel costs—and the system optimum—where a central authority coordinates traffic to minimize aggregate travel time [1]. The demand uncertainty is modeled using techniques from robust optimization [2], allowing for various uncertainty set geometries.

We reformulate the bilevel problems into single-level formulations via KKT conditions and obtain mixed-integer nonlinear programs (MINLPs) using big-M linearization, enabling the application of modern general-purpose solvers.

We provide theoretical results ensuring the existence of optimal solutions, we derive tight variable bounds for big-M constants. Additionally, we intro-

duce strengthening techniques that exploit network topology, such as identifying articulation nodes and biconnected components, to reduce problem size and complexity. A distinctive contribution of this work is the investigation of modeling alternatives across four key dimensions: (i) travel cost functions, ranging from linear to the classical BPR formulation; (ii) congestion measures, including bottleneck-focused maximum utilization, aggregate total utilization, and nonlinear BPR-based metrics; (iii) uncertainty set geometries, considering budgeted, ellipsoidal, and hose-type formulations and (iv) users behavior, comparing user equilibrium and system optimum formulations.

2 Problem Formulation

Consider a directed graph $G = (V, A)$ representing the traffic network, with node set V and arc set A. Let $S \subseteq V$ and $T \subseteq V$ represent origin and destination nodes, respectively, and $K \subseteq S \times T$ represent origin-destination pairs (commodities) with associated demands $d = (d_k)_{k \in K}$. For each arc $a \in A$, the travel cost function $c_a : \mathbb{R}_{\geq 0} \to \mathbb{R}_{>0}$ captures traversal time as a function of arc flow, assumed to be convex, non-decreasing, and continuously differentiable.

The congestion assessment problem seeks to identify the most adverse combination of demand realization and equilibrium flow:

$$\max_{d,f} \quad L(f) \quad \text{s.t.} \quad d \in \mathcal{U},\ f \in F(d), \tag{1}$$

where $L(\cdot)$ is a continuous congestion measure, $\mathcal{U}$ represents the uncertainty set containing feasible demand realizations, and $F(d)$ denotes the set of equilibrium flows for demand d.

Wardrop Equilibria. The equilibrium flow set $F(d)$ is defined through the Wardrop principles. We adopt an aggregated node-arc formulation in which commodities with the same origin are treated as a single source-commodity. Let $x = (x^s)_{s \in S}$ with $x^s = (x^s_a)_{a \in A}$ denote the aggregated commodity flows, so that the total arc flow is $f = \sum_{s \in S} x^s$.

Under the *user equilibrium* (UE), travelers act selfishly by selecting routes that minimize their individual travel times. The UE can be obtained as an optimal solution to:

$$\min_{f, x \geq 0} \sum_{a \in A} \int_0^{f_a} c_a(\xi)\, d\xi \quad \text{s.t.} \quad f = \textstyle\sum_{s \in S} x^s,\ \text{flow conservation.} \tag{2}$$

Under the *system optimum* (SO), a central planner coordinates traffic to minimize total travel time. The SO is obtained by replacing the objective with $\sum_{a \in A} c_a(f_a) f_a$.

Travel Cost Functions. We adopt the BPR function with parameter β controlling nonlinearity: $L(f) = \sum_{a \in A} c^0_a (1 + \alpha (f_a / u_a)^\beta)$ where c^0_a is free-flow time, u_a is capacity, and $\alpha = 0.15$. Three variants are studied: (i) Linear ($\beta = 1$), (ii) Quadratic ($\beta = 2$), and (iii) Classical BPR ($\beta = 4$).

Congestion Measures. Three latency functions $L(f)$ quantify network congestion: (i) Maximum Utilization: $L(f) = \max_{a \in A} f_a/u_a$ for bottleneck identification; (ii) Total Utilization: $L(f) = \sum_{a \in A} f_a/u_a$ capturing aggregate load; (iii) BPR Congestion: $L(f) = \sum_{a \in A} c_a^0(1 + \alpha(f_a/u_a)^4)$ reflecting nonlinear effects.

3 Results

We conducted extensive experiments on two benchmark collections: subnetworks derived from the Sioux Falls network [3] and instances from the SNDlib telecommunications library [4]. For each network, multiple demand configurations were generated by varying the number of commodities.All experiments were performed using Gurobi 12.0.1 in Python 3.10.15.

First, we evaluated the impact of topology-based strengthening techniques, finding that they yield computational improvements, particularly for larger instances. Then, we studied the impact of different modeling choices on the computational performance and congestion estimates. The travel cost function emerges as the dominant factor affecting both computational tractability and congestion estimates, with linear costs being the easiest to solve and BPR costs the most challenging but yielding lower congestion estimates. The congestion measure has a secondary impact on computation time but influences congestion estimates: BPR-optimized flows lead to congestion levels similar to those obtained under other measures, resulting in a robust default choice when the most appropriate congestion metric is uncertain. As for uncertainty sets, Budgeted uncertainty provides the best computational tractability while maintaining meaningful protection against demand variability. Finally, we compared the UE and SO formulations, finding that UE solves significantly faster while achieving comparable congestion levels.

Acknowledgments. Y. Beck and I. Ljubić acknowledge ANR project "DDROP". S. Mattia is a member of GNAMPA-INdAM.

Disclosure of Interests. The authors have no competing interests.

References

1. Wardrop, J.G.: Some theoretical aspects of road traffic research. Proc. Inst. Civ. Eng. **1**(3), 325–362 (1952)
2. Bertsimas, D., Brown, D.B., Caramanis, C.: Theory and applications of robust optimization. SIAM Rev. **53**(3), 464–501 (2011)
3. LeBlanc, L.J., Morlok, E.K., Pierskalla, W.P.: An efficient approach to solving the road network equilibrium traffic assignment problem. Transp. Res. **9**(5), 309–318 (1975)
4. Orlowski, S., et al.: SNDlib 1.0—Survivable Network Design Library. Networks **55**(3), 276–286 (2010)

A Branch-and-Price Approach for Lot-sizing and Scheduling Problem in Co-Production Systems

Eyüp Ensar Işık[1,2][0000−0002−9180−0243], Nabil Absi[3][0000−0002−9701−5772], and Semra Ağralı[4][0000−0002−5735−3275]

[1] Boğaziçi University, Istanbul, Türkiye
[2] Yildiz Technical University, Istanbul, Türkiye
[3] Mines Saint-Étienne, Univ Clermont Auvergne, INP Clermont-Auvergne, Gardanne, France
[4] MEF University, Istanbul, Türkiye

Abstract. Co-production systems enable the simultaneous production of multiple products. Determining lot sizes that satisfy demand while decreasing inventory costs is therefore essential. Moreover, as the setup cost for co-production units may exceed those of individual products, the sequence of lots is also critical. Considering these challenges, this study addresses a lot-sizing and scheduling problem in co-production systems. We first adapt the well-known general lot-sizing and scheduling model, and then develop a mixed-integer programming (MIP) model that incorporates all possible production sequences in advance. Since the number of possible production sequences can be exponential, we propose a branch-and-price approach. We evaluate the performance of MIP models and the branch-and-price approach using benchmark data in the literature.

Keywords: Lot-sizing and scheduling problem · Co-production systems · Column generation · Branch-and-price.

1 Introduction

In today's production systems, several different products are produced simultaneously in a single production run, both for the efficient use of limited resources and due to the specific physical/chemical structure of the production process. The co-products can be of different quality levels, as in semiconductor production [2], or completely different products, as in glass production [7]. Production systems in which different products are produced simultaneously in the same production process are called co-production [1]. Co-production systems are divided into different categories depending on the nature of the production process: (i) controlled or uncontrolled, (ii) deliberated or non-deliberated [6]. Systems with different combinations of these categories exist.

Co-production decisions vary depending on which products are produced together, in what quantities, when they are produced, and in what sequence, all of which significantly affect a company's profitability. Examples of such production processes include flat glass production [7], metal press operations [1],

cogeneration systems in the energy sector [5]. There is also a strong connection between the circular economy and co-production, as the circular economy aims to minimize the waste and use of raw materials in the production of goods and services [6]. In this context, co-production processes are of great importance in today's world, and addressing production planning and scheduling problems in co-production systems can shed light on the solution of an important problem.

2 Problem Description and Solution Method

The problem can be formulated as a General Lot-sizing and Scheduling Problem (GLSP). In the GLSP framework, time is divided into macro-periods, which are then divided into micro-periods. The lengths of micro-periods are decision variables and are indirectly determined by the type and amount (or processing time) of the Co-production Unit (CU) produced in each micro-period. When a CU is produced, multiple products can be produced in different quantities. An MIP model is developed for the GLSP in co-production systems based on the standard formulation given in [3].

A new model is also proposed based on a pattern-based model. In this model, a product sequence pattern is selected for each period. This approach differs from the sequence-oriented models in [4], as it does not fix product sequences over the entire planning horizon or split sequences; instead, it fixes only the pattern for a sequence of products that will be produced in a period. With this approach, the minimum setup cost associated with each feasible pattern can be calculated in advance and incorporated into the objective function, while the remaining cost accounts for setup changeovers between periods.

3 A Branch-and-Price Approach

In the pattern-based model, the number of variables is very large and grows exponentially. Therefore, a Branch-and-Price (BP) approach based on column generation (CG) is proposed. Using CG, the solution process starts with a subset of patterns for each period and iteratively adds new columns until no column with a negative reduced cost remains. To do so, the binary variables that choose patterns in the pattern-based model are relaxed to obtain the linear programming relaxation of the restricted master problem.

The restricted master problem includes only a subset of all possible patterns. To generate patterns not yet included in the subset, their associated reduced cost must be evaluated. Reduced costs are calculated using the optimal dual multipliers from the master problem. A new column (i.e., a pattern) is added if its reduced cost is negative. Given an optimal solution of the restricted master problem, the pricing problem minimizes the reduced cost to find a feasible pattern. For each period, the pricing problem can be formulated as a prize-collecting traveling salesman problem.

The BP algorithm requires an initial set of patterns. In this study, all possible patterns that contain at least three CUs are generated and used as the initial

set of columns in the BP algorithm. This initial set is always feasible since at least one pattern is guaranteed to include CUs capable of satisfying the demand. Integer solutions are generated via a branching procedure based on arc flows, representing the setup changeover from one CU to another. If column generation terminates with a fractional solution, two branches are created: in one branch, the corresponding flow is enforced, while in the other, it is forbidden.

4 Computational Study

To evaluate the performance of the proposed approaches, benchmark instances from [5] are used. These instances include 20 products across three product families. Co-production units are generated randomly, while ensuring that each product belongs to at least one CU to guarantee feasibility. Parameters that are not part of the data set given in [5] are generated as follows: production times and minimum order quantity values are set to 1 unit. The capacity is fixed as a function of the number of products, which is sufficient to satisfy all demands for all instances. Setup costs are calculated as a function of variable production costs.

All solution methods are implemented in C++, and the IBM CPLEX 22.1.1 (64-bit) solver is used with a four-thread limit. For each instance, a time limit of 600 seconds is imposed. Two MIP formulations with the BP approach are compared. The results are analyzed on classified benchmark problem instances according to the problem sizes, and for each problem set, 10 random instances are created. The results are presented in Table 1. For all solution approaches, the average solution time, the number of instances for which the solution methods provide optimal/feasible solutions, and the average optimality gap are provided. For the BP approach, additional statistics: number of columns, number of nodes on the BP tree, and number of iterations in the CG process are also provided.

Table 1. Comparison of MIP Models and Branch-and-Price Approach

			GLSP-based Model			Pattern-based Model		
$\|P\|$	$\|T\|$	$\|N\|$	*Time*	*#opt / #fea*	*Gap (%)*	*Time*	*#opt / #fea*	*Gap (%)*
3	3	6	0.59	10/10	0.00	0.08	10/10	0.00
4	4	6	78.37	10/10	0.00	0.23	10/10	0.00
5	5	6	600.00	0/10	26.28	4.62	10/10	0.00
3	3	10	84.35	10/10	0.00	1.09	10/10	0.00
4	4	10	545.99	1/10	58.06	5.93	10/10	0.00
5	5	10	600.00	0/10	85.92	18.48	10/10	0.00

			Branch-and-Price Approach					
$\|P\|$	$\|T\|$	$\|N\|$	*Time*	*#opt / #fea*	*Gap (%)*	*#Cols*	*#BP Nodes*	*#CG Iters*
3	3	6	312.90	7/10	4.39	648.00	3983.60	4248.40
4	4	6	557.60	1/10	90.00	640.10	4900.50	5159.10
5	5	6	600.00	0/10	100.00	596.50	4232.00	4447.10
3	3	10	600.00	0/10	100.00	20861.60	415.20	607.80
4	4	10	600.00	0/10	100.00	18860.90	359.80	519.20
5	5	10	600.00	0/10	100.00	20680.00	19.50	40.70

5 Conclusion and Future Work

Co-production systems allow for the simultaneous production of multiple products. As a result, it is necessary to determine lot sizes that satisfy demand while minimizing inventory costs. In addition, since the setup cost between co-production units can be higher than that between individual products, the production sequence of lots is also crucial.

This study addresses the lot-sizing and scheduling problem in co-production systems. Two MIP models are proposed, referred to as the GLSP-based model and the pattern-based model. A branch-and-price approach is developed based on the pattern-based formulation. The proposed models and the BP approach are tested on benchmark instances. Computational results show that the pattern-based MIP model outperforms the other approaches, although further improvements to the BP algorithm could enhance its performance.

Acknowledgments. This research is supported by The Scientific and Technological Research Council of Türkiye (TUBITAK) Project No: 223M099. Eyüp Ensar Işık's research is supported by TUBITAK 2211-A National PhD Scholarship Program.

Disclosure of Interests. The authors declare that they have no competing interests.

References

1. Ağralı, S.: A dynamic uncapacitated lot-sizing problem with co-production. Optimization Letters **6**(6), 1051–1061 (Aug 2012). https://doi.org/10.1007/s11590-011-0287-1
2. Bitran, G.R., Gilbert, S.M.: Co-production processes with random yields in the semiconductor industry. Operations Research **42**(3), 476–491 (1994). https://doi.org/10.1287/opre.42.3.476
3. Fleischmann, B., Meyr, H.: The general lotsizing and scheduling problem. Operations-Research-Spektrum **19**(1), 11–21 (Mar 1997). https://doi.org/10.1007/BF01539800
4. Guimarães, L., Klabjan, D., Almada-Lobo, B.: Modeling lotsizing and scheduling problems with sequence dependent setups. European Journal of Operational Research **239**(3), 644–662 (2014). https://doi.org/https://doi.org/10.1016/j.ejor.2014.05.018
5. Pamuk, B., Ağralı, S., Taşkın, Z.C., Kabakulak, B.: A lot-sizing problem in deliberated and controlled co-production systems. IISE Transactions **54**(10), 950–962 (2022). https://doi.org/10.1080/24725854.2021.2022250
6. Suzanne, E., Absi, N., Borodin, V.: Towards circular economy in production planning: Challenges and opportunities. European Journal of Operational Research **287**(1), 168–190 (2020). https://doi.org/10.1016/j.ejor.2020.04.043
7. Taşkın, Z.C., Ünal, A.T.: Tactical level planning in float glass manufacturing with co-production, random yields and substitutable products. European Journal of Operational Research **199**(1), 252–261 (2009). https://doi.org/10.1016/j.ejor.2008.11.024

The Effect of Carbon Emission on Production and Investment Planning

İrem Sultan Ünver[1], Çağrı Koç[1], Eda Yücel[2], and Bahar Y. Kara[3]

[1] Department of Industrial Engineering, Hacettepe University, Ankara, Türkiye
[2] Department of Industrial Engineering, TOBB University of Economics and Technology, Ankara, Türkiye
[3] Department of Industrial Engineering, Bilkent University, Ankara, Türkiye

Abstract. This study introduces a novel integrated planning problem motivated by the European Union's Carbon Border Adjustment Mechanism (CBAM) and the increasing environmental consciousness of consumers. Our problem considers carbon emission sensitive production and investment planning decisions that jointly optimize equipment investment decisions, product composition choices, and production planning over a multi-period horizon. A distinguishing feature of the study is the modeling of emission-sensitive demand, where the market demand is endogenously determined by the carbon footprint of the production process. We formulate this problem as a mixed-integer linear programming model to analyze the strategic trade-offs between profitability and environmental performance under strict regulatory constraints.

Keywords: Production Planning · Investment Decisions · Carbon Emissions · CBAM

1 Motivation and Originality

The manufacturing landscape is undergoing a paradigm shift driven by strict regulatory frameworks, such as the European Union Emissions Trading System (ETS) and the Carbon Border Adjustment Mechanism (CBAM), alongside a growing shift in consumer behavior toward low-carbon products. While the existing literature extensively investigates carbon-aware production planning and emission regulation mechanisms [1, 6], as well as production and investment decisions under various carbon pricing schemes [2, 3], most models treat market demand as exogenous or solely price-dependent.

Only a limited number of studies explicitly account for emission-dependent demand, typically within simplified inventory or single-product settings [4]. However, in many real-world manufacturing systems, firms must simultaneously decide on long-term technology investments, alternative product compositions (Bill of Materials), and operational production levels while facing demand that reacts to their environmental performance.

The primary contribution of this research is the simultaneous optimization of *technology investment decisions* and *product BOM choices* under **emission-sensitive demand** in a multi-period setting. By explicitly modeling the feedback loop between production-related emissions and market demand, the proposed framework demonstrates that ignoring this interaction may lead to systematically suboptimal investment timing and production strategies, particularly under stringent emission regulations and environmentally conscious markets.

2 Problem Description

We consider a manufacturer producing multiple products for distinct customer segments over a finite planning horizon. The problem is characterized by an integrated decision structure where investment and production choices are tightly coupled. Our model includes sets of products, alternative product compositions, customer segments, time periods, raw materials, and production equipment.

2.1 Emission-Sensitive Demand Interaction

A core novelty of our problem definition is the treatment of environmental performance as a demand driver. We utilize an endogenous demand function where the potential market size for customer j in period t is inversely related to the total emissions (q_{jt}) allocated to their order:

$$f_{ijt}(q_{jt}) = D_{ijt} - \beta_j q_{jt} \tag{1}$$

where β_j represents the customer-specific sensitivity to carbon emissions. This mechanism captures the reality that reducing the carbon footprint is not merely a compliance cost but a lever to regain market share.

2.2 Integrated Decision Making

The firm faces a multi-period planning problem subject to budget constraints and emission caps. The model simultaneously determines:

- Equipment Investments: Decisions on when to acquire new, potentially cleaner but more expensive technologies (y_{et}) to expand capacity.
- Product Composition (BOM): Selection of alternative recipes (z_{ikt}) to activate for each product. Each recipe offers a different balance between raw material cost and emission intensity.
- Production Planning: Determination of optimal production quantities (x_{ikjt}), raw material procurement, and emission allocations given the investment and activation decisions.

All decisions are optimized jointly to maximize the total profit over the planning horizon.

3 Methodological Framework

The problem is modeled as a mixed integer linear programming (MILP) formulation, which is a commonly adopted approach for complex energy- and emission-related planning problems [5]. The simultaneous consideration of multi-period binary variables (investments and activations) and continuous flow variables creates a complex combinatorial structure capturing the dynamics of the production system.

The mathematical framework allows decision-makers to evaluate the long-term impact of strategic investments on operational efficiency and carbon compliance. Our objective function aims to maximize total profit, defined as sales revenue minus raw material costs, production costs, equipment investment expenditures, and carbon-related costs.

By solving the proposed MILP model, firms can identify the optimal timing for technology upgrades and the most profitable product compositions that satisfy both regulatory limits and emission-sensitive customer demand. The model captures the dynamic interactions between carbon taxes, investment budgets, and market share, providing robust insights for sustainable operations planning.

4 Preliminary Insights

Preliminary observations based on stylized instances suggest the following managerial insights:

- Investment Timing: High customer sensitivity (β_j) accelerates green technology investments. Firms prioritize early adoption of low-carbon equipment to prevent demand erosion, even if carbon taxes are moderate.
- BOM Flexibility: The ability to switch between product compositions acts as a buffer against carbon price volatility. The model dynamically shifts to low-emission recipes as tax rates or emission caps tighten over the horizon.

Acknowledgement

This work was supported by the Scientific and Technological Research Council of Türkiye (TÜBİTAK) under grant number 125R031. The first author was supported by the TÜBİTAK 2211 National PhD Scholarship Program. The third author was supported by the Turkish Academy of Sciences (TÜBA). These supports are gratefully acknowledged.

References

1. Benjaafar, S., Li, Y., Daskin, M.S.: Carbon footprint and the management of supply chains: Insights from simple models. IEEE Trans. Autom. Sci. Eng. 10, 99–116 (2013)

2. Hsieh, C.-C., Tsai, W.-H., Chu, P.-Y.: Analyzing production optimization models across various carbon pricing policies. Int. J. Prod. Res. 62(19), 6713–6731 (2023)
3. Wang, Q., Xu, S.X., Ji, X., Zhao, N.: Joint decisions on selling mode choice and emission reduction investment under cap-and-trade regulation. Int. J. Prod. Res. 62(19), 6753–6780 (2023)
4. Hovelaque, V., Bironneau, L.: The carbon-constrained EOQ model with carbon emission dependent demand. Int. J. Prod. Econ. 164, 285–291 (2015)
5. Liu, C., Wang, H., Tang, Y., Wang, Z.: Optimization of a multi-energy complementary distributed energy system based on comparisons of two genetic optimization algorithms. Processes 9, 1388 (2021)
6. Tapia, J.F., Lee, J., Ooi, R.E., Foo, D.C., Tan, R.R.: A review of optimization and decision-making models for the planning of CO2 capture, utilization and storage (CCUS) systems. Sustain. Prod. Consum. 13, 1–15 (2018)

Robustness of Centrality-Based Influence Maximization under Structural Noise

Melike Karatay[1,2][0000−0001−6941−4752], Yasemin Demirel[3][0000−0001−8051−6012], Fidan Nuriyeva[4][0000−0001−5431−8506], and Onur Ugurlu[5][0000−0003−2743−5939]

[1] Department of Management Information Systems, Fenerbahce University, Istanbul,Türkiye
[2] Cyberspace Studies Application and Research Center, Fenerbahce University, Istanbul, Türkiye
[3] Department of Data Science and Analytics , Istanbul Topkapi University, Istanbul, Türkiye
[4] Department of Mathematics, Yasar University, Izmir, Türkiye
[5] Department of Computer Engineering, Izmir Bakircay University, Izmir, Türkiye
onur.ugurlu@bakircay.edu.tr

Abstract. Influence Maximization is a fundamental problem in combinatorial optimization with applications ranging from viral marketing to epidemic control. While traditional approaches assume complete knowledge of the network topology, real-world networks are often inferred from noisy or incomplete data. This study investigates the robustness of centrality-based seed selection strategies under structural noise. We model network imperfection via stochastic edge deletion and analyze the stability of four major centrality metrics (Degree, k-Core, Closeness, Betweenness) on three distinct network topologies. Our theoretical analysis proves that local metrics exhibit bounded error under perturbation, whereas global path-based metrics are asymptotically fragile. Our empirical findings corroborate the theoretical predictions. In Erdos-Renyi graphs subject to 20% noise, global measures such as Closeness proved fragile, showing a ranking overlap of only 0.21 (Jaccard Index) and a diffusion loss of approximately 14% (Spread Ratio 0.86). Conversely, Degree centrality exhibited remarkable resilience, retaining 97% of its baseline performance. These results suggest that in the presence of structural imperfections, local metrics are a more reliable choice than global alternatives.

Keywords: Influence Maximization · Robustness · Centrality · Imperfect Networks · Combinatorial Optimization.

1 Introduction

The Influence Maximization (IM) problem seeks to identify a set of k seed nodes that maximize the expected information spread in a network. Since the problem is NP-hard [1], centrality-based heuristics are widely used as scalable alternatives to greedy approximation algorithms. However, a critical assumption in these

methods is that the underlying graph $G = (V, E)$ is perfectly known. In practice, networks are often imperfect—suffering from missing edges due to privacy constraints, sampling errors, or sensor failures.

In this work, we address the following research question: *How does structural noise affect the solution quality of centrality-based heuristics?* We provide a unified framework that combines theoretical bounds on ranking stability with empirical simulations using the Independent Cascade (IC) model [1]. To this end, we first provide a mathematical demonstration of the structural fragility inherent in global metrics compared to their local counterparts. Furthermore, through extensive benchmarking, we establish that local metrics exhibit graceful degradation, positioning them as the superior choice for optimization in noisy environments.

2 Theoretical Analysis

Let $G = (V, E)$ be the true graph and $G' = (V, E')$ be the observed imperfect graph obtained by deleting δ edges, i.e., $|E \setminus E'| = \delta$.

Lemma 1 (Local Stability). *For any node v, degree changes are bounded locally:*

$$|deg_G(v) - deg_{G'}(v)| \leq \delta_v \leq \delta,$$

where δ_v is the number of deleted edges incident to v.

Lemma 2 (Global Fragility). *For shortest-path-based metrics such as betweenness C_{bet}, there exist graphs where deleting a single edge ($\delta = 1$) changes a node's unnormalized score by $\Theta(|V|^2)$.*

Proof. Consider a barbell graph with two cliques of size n connected by a single bridge edge. The bridge lies on n^2 shortest paths, hence $C_{\text{bet}} = \Theta(n^2) = \Theta(|V|^2)$ for its endpoint(s). Deleting the bridge disconnects the cliques, yielding $C_{\text{bet}} = 0$ for cross-clique paths. □

3 Methodology

To evaluate the robustness of centrality metrics, we utilized the standard benchmark instances [2]. The dataset covers diverse topologies, including Barabasi-Albert (BA500) graph, which is scale-free; Erdos-Renyi (ER466) random graph; and Forest Fire (FF500) network. The latter reproduces stochastic fire-spreading behavior and, while sharing the scale-free nature of BA graphs, exhibits a notably densest structure [2]. We simulated structural imperfections via stochastic edge deletion with ratios $\alpha \in \{0.02, 0.05, 0.10, 0.20\}$ and identified the top-5% seed nodes using Degree (DC), k-Core (KC), Closeness (CC), and Betweenness (BC) centralities on the perturbed graphs. The quality of these seeds was subsequently validated on the ground-truth topology using the IC model ($\beta = 0.3$, 50 Monte Carlo runs). We quantify the results using the Jaccard Similarity index to measure ranking stability and the Spread Ratio ($\sigma_{G'}/\sigma_G$) to assess performance resilience.

4 Computational Results

The theoretical divergence between local and global metrics is most pronounced in the ER466 topology. Lacking the structural hierarchy of strong hubs, global pathways in random graphs are highly susceptible to fragmentation under noise. Consistent with the predictions of Lemma 2, path-based metrics collapsed: at a high noise level of $\alpha = 0.20$, CC dropped to a Jaccard index of 0.21, degrading the diffusion performance to 86% of the baseline (see Fig. 1(a) and 2(a)). Conversely, DC proved structurally robust, maintaining a Jaccard similarity of 0.53 and retaining 97% of the optimal spread. This disparity confirms that in homogenous topologies, local metrics offer a far more stable approximation of influence potential than their global counterparts.

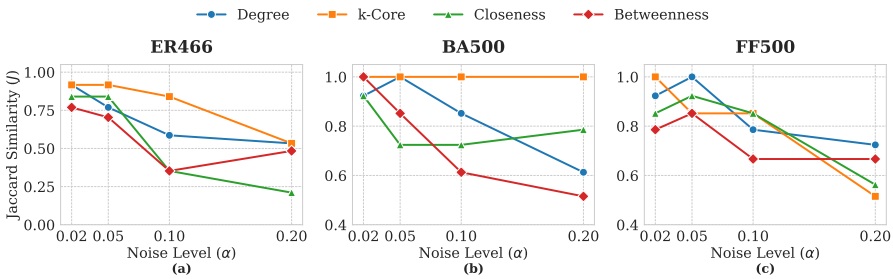

Fig. 1. Jaccard similarity indices comparing seed sets selected from clean (G) versus noisy (G') networks across varying noise levels (α).

In scale-free BA500 networks, the presence of hubs provides inherent stability, yet the performance gap between strategies persists. BC exhibited notable ranking instability ($J = 0.51$) due to the fragility of bridge edges, leading to a spread recovery drop to 93% at $\alpha = 0.20$ (Figs. 1(b) and 2(b)). On the other hand, DC leveraged the robustness of hubs to outperform its global counterpart, achieving a 96% Spread Ratio. We further observed that KC aligned closely with DC in denser configurations, while remaining strictly invariant ($J = 1.0$) in sparse, tree-like structures, demonstrating that core decomposition is structurally insensitive to leaf-node perturbations.

Finally, experiments on the FF500 revealed a counter-intuitive noise-induced exploration, where the Spread Ratio occasionally exceeded the baseline (e.g., 1.02 for DC, see Fig. 2(c)). Although ranking alignment degrades significantly (Fig. 1(c)), structural perturbations prevent the selection of redundant nodes within a dense rich-club. This stochastic disruption forces the heuristic to bypass local optima and discover spatially dispersed—and effectively more virulent—seeds. This finding implies that moderate noise can inadvertently mitigate the overfitting problem inherent in standard centrality metrics applied to community-rich topologies.

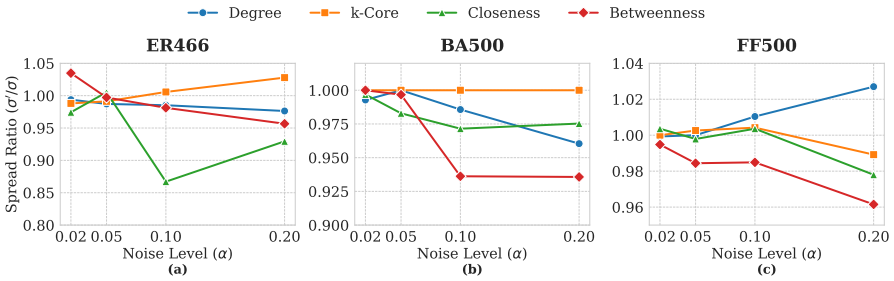

Fig. 2. The ratio of influence spread achieved by seeds selected from noisy networks relative to the clean baseline ($\sigma_{G'}/\sigma_G$) across varying noise levels (α).

5 Conclusion

This study revisits the trade-off between algorithmic complexity and robustness in Influence Maximization. While sophisticated path-dependent metrics often yield superior results in ideal conditions, we demonstrated both theoretically and empirically that they can be highly sensitive to structural perturbations. Our findings indicate that global metrics suffer notable performance degradation in noisy environments due to the fragility of long-range paths. In contrast, local heuristics exhibit a structural resilience that effectively compensates for their limited scope, consistently retaining near-optimal diffusion potential.

We aim to expand this work into a comprehensive framework suitable for broader mathematical analysis by addressing three key dimensions. First, we will transition from synthetic benchmarks to real-world networks to validate the ecological validity of these findings under complex topological constraints. Second, the analysis will be broadened to include next-generation centrality metrics, such as semi-local indices, which attempt to balance the local-global trade-off. Finally, we plan to develop advanced noise models to better reflect the multi-faceted nature of data imperfection.

References

1. Ye, Y., Chen, Y., Han, W.: Influence maximization in social networks: Theories, methods and challenges. Array **16**, 100264 (2022)
2. Aringhieri, R., Grosso, A., Hosteins, P., Scatamacchia, R.: A general evolutionary framework for different classes of critical node problems. Engineering Applications of Artificial Intelligence **55**, 128–145 (2016)
3. Das, K., Samanta, S., Pal, M.: Study on centrality measures in social networks: a survey. Social Network Analysis and Mining **8**(1), 13 (2018)
4. Borgatti, S.P., Carley, K.M., Krackhardt, D.: On the robustness of centrality measures under conditions of imperfect data. Social Networks **28**(2), 124–136 (2006)

Effects of Density Changes on a Novel Graph Burning Algorithm

Arianne Nantel[1], Arda Utku[1], and Erin Meger[1][0000−0003−3392−8211]

Queen's University, Kingston ON, K7K 2N8, Canada erin.meger@queensu.ca

Abstract. Wildfires are an increasingly devastating natural disaster worldwide, posing significant threats to ecosystem, infrastructures, and human life. In this extended abstract, we propose a new information diffusion framework, *the Wildfire Burning Model*, combing appraoches from Graph Burning and Linear Threshold Models. We evaluated the model on four well-studied graph models to assess the impact of network topology on the $wf(G)$ parameter, measuring the time to fully burn the graph. This model provides interesting future questions in both theoretical graph theory and applied wildfire modeling.

Keywords: Graph Burning · Linear Threshold Model · Wildfire Burning Model · Graph Algorithms · Infromation Diffusion

1 Introduction

As climate change continues to devastate many of the worlds forest ecosystems, it is essential that we learn to better predict the patterns of destructive wildfires. Many efforts have been made in this field, including mathematical, physics-based, and AI-driven approaches [14] [16] [17]. In this extended abstract, we propose the *Wildfire Burning Model* a novel discrete-time dynamical process, and provide preliminary results on well-studied complex network models [5].

The *Wildfire Burning Model* is built upon the Graph Burning problem, introduced by Roshanbin in 2017 [15]. *Graph Burning* is a deterministic process that describes how information, viruses or data can spread through a network. Each round a new source of fire is chosen, and the fire spreads to all adjacent unburned nodes. The Burning number of a graph $b(G)$ as the minimum number of turns needed for the entire graph to burn. It was conjectured then recently proven that for any finite connected graph G with n nodes that $b(G) \leq \sqrt{n}$ [11]. It is important to note, that determining the burning number of a graph is NP-Complete [15]. Current research has focused on graph classes and random graphs or introducing game dynamics [3] [4] [7] [12].

In linear threshold models, information cannot be passed from one node to another node without some additional parameter being surpassed. We define the traditional *Linear Threshold Model* [10]. Each vertex v influences its neighbors w based on an edge weight $b(v, w)$. Each vertex is assigned a threshold, $th(v)$ For any given node to become active, the sum of the weights of all of its active neighbors must be bigger than or equal to its threshold.

2 Algorithm Description

We introduce the *Wildfire Burning Model*. This model approximates a network of forests, where each node represents a forest with its associated dryness conditions. In our case, we consider an edge between forests to represent the possible transmission of fire. Later, we will model these simulated forests using traditional random graph models $G(n, p)$, Preferential Attachment, Duplication Divergence, and Watts–Strogatz [5].

Given a simple connected undirected graph G, we assign each $v \in V(G)$ a *dryness attribute* $f : V(G) \to [0, 1] \in \mathbb{R}$. We define $\rho \in [0, 1]$ to be the *drying rate* for a forest; this is the proportion by which its dryness increases when a fire is nearby for each nearby fire. At time $t = 0$, we begin the initial ignition of the fire at vertex u_1, where $f(u_1) = \max_{v \in V(G)} f(v)$. If there exist multiple vertices satisfying the definition of u_1, that is $f(u_1) = f(u_2) = \max_{v \in V(G)} f(v)$, then one such u_1 is chosen uniformly at random. For each subsequent timestep $t \geq 0$. Let $N_B(u)$ be the set of burning neighbors of the vertex u. The proportion $|N_B(u)|/|N(u)|$ represents how much head the forest u is receiving from adjacent forests. The forest u will burn if this ratio exceeds its dryness. That is for each $u \in V(G)$, the vertex u will catch fire (and become burning) if

$$\frac{|N_B(u)|}{|N(u)|} \geq 1 - f(u)$$

If a vertex $u \in V$ does not meet this threshold, then we say that u increases its dryness by modifying $f(u) += \rho * |N_B(u)|$. In much of this algorithm, we assume arbitrarily that $\rho = 0.05$ unless otherwise stated. The process ends once all $v \in V$ have burned. We define the *wildfire number of the graph* as $wf(G)$ to be the minimum number of steps it takes for the process to complete given the initial dryness assigned to G. Notice that $wf(G)$ is bounded below by the diameter for any graph G.

3 Results

For our experimentation, we ran a set of simulations by generating one of four different complex network models: Watts-Strogatz [18, **?**], Preferential Attachment [2], $G(n, p)$ [8], and Duplication–Divergence [9]. For each network model, we generated multiple instances using varying parameters each with 30 nodes, and on each individual graph ran the model 50 times, measuring $wf(G)$. We implemented the model in Python using NetworkX, set the dryness of each node using the pseudo-random Random function [13], and we set $\rho = 0.05$.

We note the density of the graphs, which measures the proportion of edges to vertices; as the density increases $wf(G)$ decreases across all models. In Figure **??** we show the plot of $wf(G)$ against desnity. When p is small, we notice a larger variation in $ws(G)$.For the duplication divergence model, we noticed a steeper decrease in $wf(G)$, but a consistent variance, as shown in Figure 2.

For the remaining two models we summarize their results in the following Tables. Overall, as expected, $wf(G)$ decreases as the clustering increases, as is seen in Table 3. In Watts-Strogatz where the clustering is non-monotone as p increases, while maintaining constant regularity at $k = 3$. See Table 3.

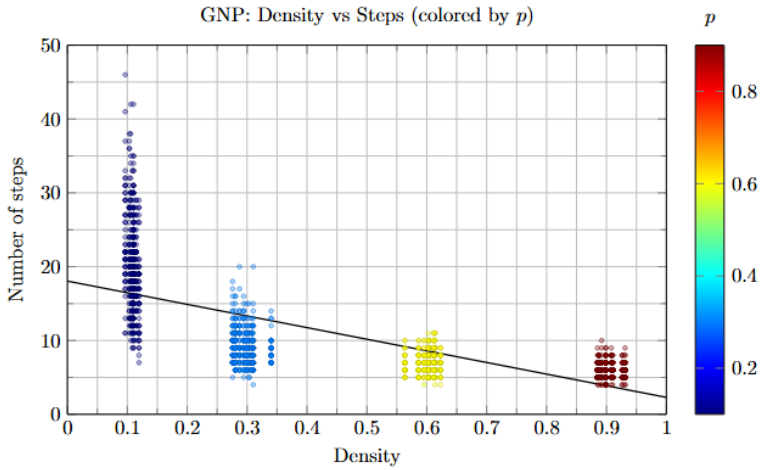

Fig. 1. $G(n, p)$ model of $wf(G)$ compared to Density for all tests.

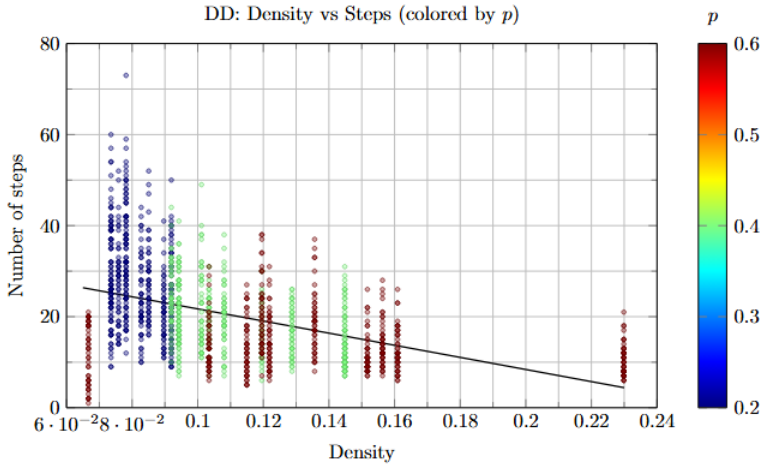

Fig. 2. Duplication Divergence model of $wf(G)$ compared to Density for all tests.

p	**Density**	**Mean** $wf(G)$	**Standard Deviation**
0.1	0.068966	59.04	16.94
0.4	0.068966	53.54	15.71
0.7	0.068966	48.16	15.18

Table 1. Mean and standard deviation of the number of burning steps for simulations on Watts-Strogatz graphs with varying rewiring probability p. The graph density remains constant across all values of p.

m	**Density**	**Mean**	**Standard Deviation**
2	0.128736	15.02	4.676
5	2.64371376	9.234	2.644
10	0.45977	7.69	2.185

Table 2. Mean and standard deviation of the number of burning steps for simulations on Barabási–Albert graphs. Increasing m, the number of edges each new node adds, increases the graph density and results in faster spread of burning.

4 Conclusion and Future Work

This model is incredibly relevant in the area of Graph Searching, as it extends the well-studied Graph Burning problem by including a linear-threshold allowing for broader application. From the theoretical side the natural next step is to improve the upper bound by utilizing the underlying graph structure for certain families.

In future models, each edge could model the conditions between individual forests, allowing for different values of ρ, thus simulating both a linear threshold model, in combination of a network flow problem. Network flow problems are a fundamental research area in combinatorial optimization [1][6][19].

Further, applied wildfire modeling would want to find real-world forest data to benchmark the accuracy of this model. These benchmarks could fine-tune the algorithm parameters (dryness and dryness factors ρ) to find values that closely match real-world fires. To better model real wildfires, the model should next incorporate the three other factors that heavily influence wildfire spread: ignition sources, continuous fuel, and weather. Applying critical node detection algorithms to this model once it has been calibrated using real world data, would tell forest fire-fighters which forests are the most critical to protect.

Acknowledgments. Supported by the Natural Sciences and Engineering Research Council of Canada NSERC (2025-05700)and by Queen's University Research Initiation Grant.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Ahuja, R.K., Magnantl, T.L., Orlin, J.B.: Network flows: theory, algorithms, and applications. Prentice-hall (1993)
2. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. Science **286**(5439), 509–512 (1999), http://www.jstor.org/stable/2899318
3. Bonato, A., Marbach, T.G., Marcoux, J., Mishura, T.: Between burning and cooling: liminal burning on graphs. arXiv preprint arXiv:2505.10727 (2025)
4. Chiarelli, N., Iršič, V., Jakovac, M., Kinnersley, W.B., Mikalački, M.: Burning game. arXiv preprint arXiv:2409.11328 (2024)
5. Chung, F.R., Lu, L.: Complex graphs and networks. No. 107, American Mathematical Soc. (2006)
6. Cook, W.J., Cunningham, W.H., Pulleyblank, W.R., Schrijver, A.: "Combinatorial Optimization". Wiley-Intersciece Series in Discrete Mathematics and Optimization (1997)
7. Devroye, L., Eide, A., Prałat, P.: Burning random trees. Electronic Communications in Probability **30**, 1–9 (2025)
8. Gilbert, E.N.: Random graphs. The Annals of Mathematical Statistics **30**(4), 1141–1144 (1959), http://www.jstor.org/stable/2237458
9. Ispolatov, I., Krapivsky, P.L., Yuryev, A.: Duplication-divergence model of protein interaction network. Phys. Rev. E **71**, 061911 (Jun 2005). https://doi.org/10.1103/PhysRevE.71.061911, https://link.aps.org/doi/10.1103/PhysRevE.71.061911
10. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 137–146 (2003)
11. Norin, S., Turcotte, J.: The burning number conjecture holds asymptotically. Journal of Combinatorial Theory, Series B **168**, 208–235 (2024)
12. Omar, M., Rohilla, V.: Burning graph classes. Graphs and Combinatorics **38**(4), 121 (2022)
13. Python Software Foundation: Python Language Reference, version 3.13.0
14. Radke, D., Hessler, A., Ellsworth, D.: Firecast: Leveraging deep learning to predict wildfire spread. In: IJCAI. pp. 4575–4581 (2019)
15. Roshanbin, E.: Burning a graph is hard. Discrete Applied Mathematics **232**, 73–87 (2017)
16. Shaddy, B., Ray, D., Farguell, A., Calaza, V., Mandel, J., Haley, J., Hilburn, K., Mallia, D.V., Kochanski, A., Oberai, A.: Generative algorithms for fusion of physics-based wildfire spread models with satellite data for initializing wildfire forecasts. Artificial Intelligence for the Earth Systems **3**(3), e230087 (2024)
17. Storey, M.A., Bedward, M., Price, O.F., Bradstock, R.A., Sharples, J.J.: Derivation of a bayesian fire spread model using large-scale wildfire observations. Environmental Modelling Software **144**, 105127 (2021). https://doi.org/https://doi.org/10.1016/j.envsoft.2021.105127, https://www.sciencedirect.com/science/article/pii/S1364815221001705
18. Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world'networks. nature **393**(6684), 440–442 (1998)
19. Williamson, D.P.: Network flow algorithms. Cambridge University Press (2019)

An Integrated ALNS–LSP Approach for the Pickup and Delivery Problem with Two Cross-Docks and 3D Loading Constraints *

Mehmet Asaf DÜZEN[1,2][0000−0002−5200−368X], Vildan ÖZKIR[2][0000−0002−2609−5234], and Umman Mahir YILDIRIM[3][0000−0003−3469−8112]

[1] Yalova University, Yalova, 77200, Türkiye
[2] Yildiz Technical University, İstanbul, 34330, Türkiye
[3] İstanbul Bilgi University, İstanbul, 34440, Türkiye

Abstract. This study investigates an extended Pickup and Delivery Problem (PDP) with three-dimensional (3D) loading constraints and cross-dock selection. Despite its practical relevance, the number of studies proposing solution methods to this extension remains very limited in the literature. This study proposes an Adaptive Large Neighborhood Search (ALNS) to solve PDP instances with two cross-dock facilities while considering 3D loading constraints, including orientation, non-overlapping, support, and stability. The contribution of this study relies on its predefined network structure, which permits transfers between these cross-docks. The primary objective of the proposed algorithm is to minimize the total travel distance over the network.
The solution framework is enhanced with a specialized Local Search-based Packing algorithm (LSP) that acts as a feasibility assessment procedure for the 3D loading subproblem. This LSP procedure ensures that the generated routes are physically loadable and satisfy all spatial constraints. Overall, the proposed approach provides an effective solution framework for generating operationally feasible and cost-efficient routing plans under realistic 3D loading constraints.

Keywords: PDP · Multiple Cross-Docking Facility · 3D Loading · ALNS

1 Introduction

The PDP is a well-known logistic problem in which each pickup request is associated with exactly one corresponding delivery request. From a practical perspective, incorporating 3D loading constraints [1] and cross-docking activities [2] significantly enhances the applicability of the PDP. The 3D Loading Pickup and Delivery Problem with multiple Cross-Docks (3L-PDPMCD) extends the classical PDP by jointly considering 3D loading constraints and cross-dock facilities, focusing on synchronizing inbound and outbound flows to minimize total cost, which is total travel distance in most cases. Several studies have addressed the

* Extended Abstract

PDP with a single cross-dock facility (PDPCD). Wen et al. [3] and Abad et al. [4] investigated routing and consolidation strategies within cross-docking systems, where vehicle capacity is modeled with solely weight capacity constraint, which can be considered as a one-dimensional packing problem. In a recent study, Ji et al. [5] incorporated spatial dimensions by integrating two-dimensional loading constraints into the vehicle routing problem with cross-docking. However, studies considering 3D loading constraints in a multi-cross-dock environment still remain very limited. In real-world applications, the physical dimensions and spatial arrangement of cargo significantly impact route feasibility. Ignoring 3D loading constraints, such as box orientation, non-overlapping, and stability, frequently results in solutions that are distance-wise ideal but practically infeasible to load [6].

This study introduces an integrated approach that incorporates 3D loading constraints into a two-cross-dock PDP framework. The primary contribution of this study is to propose an efficient solution methodology that ensures spatial feasibility while maintaining routing efficiency.

2 Solution Approach

Both the PDP and the 3D packing problem are well-known NP-hard problems. Their integration substantially increases problem complexity, often rendering exact solution approaches impractical for realistic, real-world instances.

Given the high computational complexity of the integrated 3L-PDPMCD, an improved ALNS algorithm is developed based on the principles introduced by Pisinger and Ropke [7]. For the PDP with multiple cross-docks, routes are constructed using the proposed ALNS algorithm, after which routing decisions are dynamically validated by a Local Search–based Packing (LSP) algorithm serving as a feasibility validation procedure. The proposed hybrid ALNS–LSP strategy guarantees both routing efficiency and spatial load compatibility, resulting in practical and operationally viable solutions. The structure of the proposed algorithm is presented in Algorithm 1.

The routing optimization is performed using an ALNS algorithm with Simulated Annealing acceptance criterion. In each iteration, the algorithm selectively disrupts the current solution using destroy operators and subsequently reconstructs it through repair operators to explore the solution space effectively. Based on the methodologies proposed in the literature [2, 6, 7], our implementation employs eight destroy and three repair operators. These procedures are designed to eliminate inefficient segments of the routes—such as high-cost nodes or spatial clusters—and re-insert requests to improve overall solution efficiency. The LSP validates each route by ensuring physical loadability, with items processed using a priority rule and packed via a Back-Left-Low strategy combined with Maximum Touching Area for stability [6]. The algorithm applies 'Item Exchange' and 'Sequence Swap' rules, which improves loading sequence and vehicle utilization. Any route confirmed as feasible must satisfy geometry, orientation, non-overlapping, and support constraints.

Algorithm 1 ALNS–LSP for 3L-PDPMCD

Input: Problem Instance I, Max Iterations N, Segment Size P
$S_0 \leftarrow$ Generate Initial Solution(I)
$S_{best} \leftarrow S_0, S_{curr} \leftarrow S_0$
$T \leftarrow T_0$ (Initial temperature), $\rho \leftarrow \{1, \ldots, 1\}$ (Operator weights)
$\pi \leftarrow \{0, \ldots, 0\}$ (Operator scores)
for $iter = 1$ **to** N **do**
    Select destroy operator $i \in H^-$ and repair operator $j \in H^+$ based on ρ
    $S_{new} \leftarrow j(i(S_{curr}))$
    $is\_feasible \leftarrow$ **Call LSP Validation** with S_{new} routes
    if $is\_feasible$ **then**
        if $Accept(S_{new}, S_{curr}, T)$ **then**
            $S_{curr} \leftarrow S_{new}$
            Update scores π based on solution quality
            if $Cost(S_{new}) < Cost(S_{best})$ **then**
                $S_{best} \leftarrow S_{new}$
            end if
        end if
    end if
    $T \leftarrow T \times \alpha$
    if $iter \pmod P == 0$ **then**
        Update weights ρ using scores π
        Reset scores $\pi \leftarrow 0$
    end if
end for
return S_{best}

3 Preliminary Results

To evaluate the computational performance of the proposed ALNS–LSP approach, five instances with varying numbers of items generated from the Li and Lim [8] benchmark dataset are considered. A preliminary experiment is conducted on these instances, ranging from 10 to 30 items (b10-b30) for 5 and 7 requests (r5-r7). For the small sized instances, we obtain the optimal solution with MILP model. As summarized in Table 1, for small-scale instances both the MILP model and the ALNS–LSP reached the same optimal solution. A notable

Table 1. Performance Comparison between MILP and ALNS–LSP

Instance	MILP Dist.	MILP Time (s)	ALNS–LSP Dist.	ALNS–LSP Time (s)	Gap (%)
r5-b10	75.3	0.91	75.3	1.40	0.00
r5-b20	75.3	3.11	75.3	1.47	0.00
r5-b30	118.7	3600.0*	75.3	1.60	-36.56
r7-b10	145.5	25.41	145.5	2.05	0.00
r7-b20	145.5	59.51	145.5	2.65	0.00

*Time limit of 3600 seconds reached without proving optimality.

observation concerns the performance of the ALNS–LSP on the more complex instances (r5-b30 to r7-b20). Whereas the MILP model was terminated at the 3600-second time limit with considerable optimality gaps, the ALNS-LSP heuristic obtained high-quality solutions in under 3 seconds. These results demonstrate

that the hybrid approach ensures 3D loading feasibility while generating high-quality routing plans within a short computational time.

4 Conclusion

This study addressed a PDP with two cross-docks and 3D loading constraints (3L-PDPMCD). We developed an ALNS algorithm with a specialized LSP subroutine to handle the spatial and routing complexities simultaneously. The results of the computational experiments indicate that the proposed approach constitutes a viable framework for generating operationally feasible and economically efficient routing plans under realistic 3D loading constraints. Future research will focus on further assessing the scalability of the ALNS–LSP on larger and more complex datasets.

Acknowledgments. This study is part of a PhD thesis. The first author (Mehmet Asaf DÜZEN) is supported by the TUBITAK 2211-A National PhD Scholarship Program. Additionally, this research is supported by the TUBITAK ARDEB 1001 Program (Project No: 123M504).

Disclosure of Interests. The authors have no competing interests to declare.

References

1. Goodarzi, A.H., Tavakkoli-Moghaddam, R., Amini, A.: A new bi-objective vehicle routing-scheduling problem with cross-docking: Mathematical model and algorithms. Comput. Ind. Eng.**149**, 106832 (2020). https://doi.org/10.1016/j.cie.2020.106832
2. Maknoon, Y., Laporte, G.: Vehicle routing with cross-dock selection. Comput. Oper. Res. **77**, 254–266 (2017). https://doi.org/10.1016/j.cor.2016.08.007
3. Wen, M., Larsen, J., Clausen, J., Cordeau, J.F., Laporte, G.: Vehicle routing with cross-docking. J. Oper. Res. Soc. **60**(12), 1708–1718 (2009).https://doi.org/10.1057/jors.2008.108
4. Abad, H.K.E., Vahdani, B., Sharifi, M., Etebari, F.: A bi-objective model for pickup and delivery pollution-routing problem with integration and consolidation shipments in cross-docking system. J. Clean. Prod. **193**, 784–801 (2018).https://doi.org/10.1016/j.jclepro.2018.05.046
5. Ji, B., Zhang, Z., Yu, S.S., Zhou, S., Wu, G.: Modelling and heuristically solving many-to-many heterogeneous vehicle routing problem with cross-docking and two-dimensional loading constraints. Eur. J. Oper. Res. **306**(3), 1219–1235 (2023).https://doi.org/10.1016/j.ejor.2022.08.001
6. Cen, X., Zhou, G., Ji, B., Yu, S.S., Zhang, Z., Fang, X.: Modelling and heuristically solving the three-dimensional loading constrained vehicle routing problem with cross-docking. Adv. Eng. Inform. **57**, 102029 (2023). https://doi.org/10.1016/j.aei.2023.102029
7. Pisinger, D., Ropke, S.: A general heuristic for vehicle routing problems. Comput. Oper. Res. **34**(8), 2403–2435 (2007). https://doi.org/10.1016/j.cor.2005.09.012
8. Li, H., Lim, A.: A metaheuristic for the pickup and delivery problem with time windows. In: Proceedings of the 13th International Conference on Tools with Artificial Intelligence (ICTAI 2001), pp. 160–167. IEEE Computer Society, Dallas (2001) https://doi.org/10.1109/ICTAI.2001.974461

Robust Portfolio Optimization with Return Prediction

Burak Kara[1], Emre Sefer[2], and İhsan Yanıkoğlu[1]

[1] Department of Industrial Engineering, Özyegin University, Istanbul, Türkiye
[2] Department of Artificial Intelligence and Data Engineering, Özyegin University, Istanbul, Türkiye burak.kara.18214@ozu.edu.tr

1 Introduction

Portfolio selection is a well-known problem in finance and operations research. The classical mean-variance model of Markowitz [6] balances expected return and risk by maximizing an estimated expected return while penalizing portfolio volatility. A key practical challenge is that future returns are unknown at the time of decision making, and expected returns are therefore replaced by estimates, often simple sample means. Replacing such estimates with more informative predictors is a natural direction for improving portfolio construction. In this study, we use machine learning (ML) to produce point forecasts of next period returns and embed these forecasts into the robust portfolio optimization model. We adopt robust optimization (RO) to explicitly control the impact of return estimation error, as RO enforces protection against all realizations in an uncertainty set [3]. In the OR literature, [2] and [1] survey robust portfolio selection models and solution approaches extensively. In particular, the robust mean-variance formulation of [7] provides a tractable way to protect against uncertainty in expected returns through robustness and risk aversion penalties. Our model is related in that it adopts this robustness and risk penalty structure, but it differs by integrating ML-based return forecasts, residual-based uncertainty calibration, and implementable trading components: a no-trade threshold and a minimum holding cardinality. With these components, the model becomes a mixed-integer (combinatorial) optimization problem and is NP-hard due to the discrete trading and cardinality decisions. We combine ML-based return predictions with residual-based uncertainty calibration and evaluate the resulting portfolio decisions through trading simulations detailed in Section 2.

1.1 Robust Optimization Model

The proposed model incorporates two implementable trading features: a no-trade threshold that avoids small reallocations and a minimum holding cardinality. The conic robustness and volatility penalty terms are modeled using the second-order cone representation in [7]. We consider a universe of n assets and make portfolio decisions at discrete rebalancing periods. Let $\boldsymbol{w} \in \mathbb{R}^n$ denote the portfolio weight vector and $\hat{\boldsymbol{\mu}} \in \mathbb{R}^n$ the predicted return vector for the next holding period. The

proposed model treats $\hat{\boldsymbol{\mu}}$ as a point estimate and protects the decision against deviations within an uncertainty set calibrated from a warm-up window of recent prediction residuals, yielding a forecast-error covariance estimate. Let w^0 be the current portfolio and define trades $q_i = w_i - w_i^0$. Binary variables x_i indicate whether asset i is held, y_i whether it is traded, and s_i selects the sign branch used in the no-trade constraints. Parameters $u \in (0, 1]$ and $\ell \in [0, u]$ denote upper and lower bounds for position sizes, $\tau_i \geq 0$ is the no-trade threshold, and $K_{\min}$ is the minimum number of assets selected. Finally, $t_\mu, t_\Sigma \geq 0$ are auxiliary variables for the robustness and volatility-penalty terms, Λ denotes the forecast-error covariance matrix estimated from prediction residuals, and $M > 0$ is a big-M constant.

$$\max_{\boldsymbol{w},\boldsymbol{q},\boldsymbol{x},\boldsymbol{y},\boldsymbol{s},t_\mu,t_\Sigma} \quad \hat{\boldsymbol{\mu}}^\top \boldsymbol{w} - \kappa\, t_\mu - \delta\, t_\Sigma \tag{1}$$

$$\text{s.t.} \quad \mathbf{1}^\top \boldsymbol{w} = 1 \tag{2}$$

$$\left\| \Lambda^{1/2} \boldsymbol{w} \right\|_2 \leq t_\mu \tag{3}$$

$$\left\| \hat{\Sigma}^{1/2} \boldsymbol{w} \right\|_2 \leq t_\Sigma \tag{4}$$

$$\ell\, x_i \leq w_i \leq u\, x_i \qquad \forall i \tag{5}$$

$$\sum_{i=1}^{n} x_i \geq K_{\min} \tag{6}$$

$$q_i = w_i - w_i^0 \qquad \forall i \tag{7}$$

$$-M\, y_i \leq q_i \leq M\, y_i \qquad \forall i \tag{8}$$

$$q_i \geq \tau_i\, y_i - M(1 - s_i) \qquad \forall i \tag{9}$$

$$q_i \leq -\tau_i\, y_i + M\, s_i \qquad \forall i \tag{10}$$

$$\boldsymbol{w} \geq \mathbf{0}, \;\; t_\mu, t_\Sigma \geq 0, \;\; x_i, y_i, s_i \in \{0, 1\} \qquad \forall i. \tag{11}$$

The term $\hat{\boldsymbol{\mu}}^\top \boldsymbol{w}$ is the predicted next-period portfolio return. The penalties $\kappa\, t_\mu$ and $\delta\, t_\Sigma$ trade off return against robustness (estimation error) and risk (volatility), respectively. t_μ measures sensitivity to return prediction error through Λ, while t_Σ measures volatility through $\hat{\Sigma}$, both enforced via the second-order cone constraints (3) and (4) as in [7]. Constraint (2) ensures full investment. Constraints (5) and (6) link selection and weights by enforcing $w_i \in [\ell, u]$ when $x_i = 1$, $w_i = 0$ when $x_i = 0$, and requiring at least $K_{\min}$ selected assets. Constraints (7)-(10) implement the no-trade threshold. If $y_i = 0$, then (8) enforces $q_i = 0$, while if $y_i = 1$ the big-M two-branch constraints (9) and (10) enforce $|q_i| \geq \tau_i$. Following [7], the parameters κ and δ control the trade-off between expected return, variance, and protection against estimation error. In particular, δ is induced by a user parameter $p \in [0, 1]$ as $\delta = \sqrt{\frac{p}{1-p}}$ for $p \in [0, 1)$ and $\delta = +\infty$ for $p = 1$, larger p implies stronger variance penalization. Larger κ corresponds to a larger uncertainty set and thus a more conservative treatment of prediction uncertainty. We select κ by calibrating it to a desired empirical coverage level

α. Using a warm-up window of recent residuals, we compute their ellipsoidal distances under the current Λ estimate and set κ^2 to the empirical α-quantile of these squared distances, so that approximately an α fraction of recent residuals satisfy $\varepsilon^\top \Lambda^{-1} \varepsilon \leq \kappa^2$ (higher α implies larger κ).

2 Numerical Results

2.1 Return Prediction using ML

Return forecasts enter the robust model through the point predictions $\hat{\boldsymbol{\mu}}$ and through the distribution of prediction residuals used to estimate Λ. Machine learning (ML) methods have been widely explored for return prediction in portfolio optimization settings [5]. We compare five prediction models from three families: rolling *sample mean* (SM), a *multilayer perceptron* (MLP), and *Transformer* models, all trained on log returns and evaluated in a walk-forward setting. These forecasts are then used as inputs to the optimization model to produce a robust and interpretable investment strategy.

Computational experiments are conducted on two 2025 datasets, each containing 125 two-day periods and 13 assets (yielding 1625 realized predictions per dataset), constructed from Dow 30 stocks, bond ETFs, commodities, and cryptocurrencies. Each model specification is identified by its lookback length (LB) and retraining frequency (RF), both measured in periods. Table 1 reports point-forecast errors (MAE, RMSE), directional accuracy (DirAcc), and cross-sectional ranking quality (RankIC) for the two markets. Overall, point forecast accuracy is similar across models, while Transformers tend to deliver slightly better ranking performance (RankIC), suggesting potential benefits when cross-asset structure is captured effectively. This indicates that Transformer models can be advantageous when trained with richer feature representations and tuned carefully. Further improvements in point forecast accuracy may be achieved by enhancing the feature inputs and refining the training setup.

Table 1. Prediction model performance comparison on two markets (LB: lookback window length, RF: retraining frequency in periods).

Pred. Model	LB	RF	crypto-stocks				crypto-commodities-bonds			
			MAE	RMSE	DirAcc	RankIC	MAE	RMSE	DirAcc	RankIC
SM	30	1	0.024	0.037	0.492	-0.018	0.016	0.031	0.527	-0.026
SM	60	1	0.024	0.037	0.475	-0.043	0.016	0.031	0.542	-0.011
MLP	20	1	0.026	0.039	0.506	0.001	0.018	0.035	0.496	-0.049
Transformer	50	2	0.025	0.038	0.490	0.011	0.017	0.031	0.536	0.122
Transformer	50	5	0.025	0.038	0.493	0.001	0.017	0.033	0.516	0.035

2.2 Trading Simulation

We evaluate our method, termed *robust portfolio optimization* (`RPO`), by simulating a sequence of trading decisions on the 2025 datasets introduced in Section 2.1.

Table 2. Trading simulation summary on the full-year dataset and the declining-market window. All rows report averages over their tested variants.

		125 periods - crypto-commodities-bonds			41 periods - crypto-stocks		
	α	Avg Capital	Avg Exp Cap	Avg # Inv	Avg Capital	Avg Exp Cap	Avg # Inv
RPO	0	1153.07	2484.55	3.19	857.49	1487.70	3.98
RPO	0.1	1056.94	1080.32	3.07	866.90	1122.85	5.29
RPO	0.3	1056.06	1075.79	3.06	868.94	1108.31	5.36
buy&hold	-	929.17	-	4	854.18	-	4
top-k-ewp	-	1217.96	3000.44	4	835.56	1513.66	4

Each dataset spans 125 two-day periods. In addition, to test robustness in a pessimistic market environment, we form a stress test instance by restricting the crypto-stocks dataset to a 41-period window characterized by a sustained market decline. At the beginning of each run, an initial capital of 1000 units is allocated and the portfolio is rebalanced every two-day period. In each period, a return forecast is produced in a walk-forward manner and `RPO` is solved using the point predictions $\hat{\boldsymbol{\mu}}$, the forecast-error covariance estimate Λ obtained from recent residuals, and the return covariance estimate $\hat{\Sigma}$ used for the volatility penalty. The resulting portfolio is held over the next period and wealth is updated using realized returns. All instances of `RPO` are solved using Gurobi Solver [4].

We conduct a grid search over the prediction model specification in Table 1, robustness coverage level α (with $\alpha = 0$ being non-robust), and risk-aversion parameter p. For each configuration, we run the full trading simulation and summarize performance by reporting the averages of realized final wealth (*Avg Capital*), the model-implied final wealth (*Avg Exp Cap*), and the number of invested assets (*Avg # Inv*) per period.

We compare `RPO` against two benchmark approaches. First, `top-k-ewp` constructs an equally weighted portfolio over a selected subset of assets. In each period, it selects the top-k assets by predicted return and allocates equal weights among them. Second, `buy&hold` selects the top-k assets once at the start using a sample-mean predictor and then holds this portfolio without rebalancing. For both `top-k-ewp` and `buy&hold`, we evaluate multiple variants by using a grid search over $k \in \{3, 4, 5\}$ and alternative prediction methods. Specifically, `buy&hold` uses SM with lookback window lengths 30 and 60, whereas `top-k-ewp` is paired with each prediction model specification in Table 1. Table 2 reports averages over these variants to keep the presentation brief. Overall, increasing α reduces the gap between expected and realized outcomes, indicating improved reliability. Robust variants are less prone to overconfidence than the non-robust case. This effect is particularly relevant in the declining market window, where robust models perform more favorably.

References

1. Ghahtarani, A., Saif, A., Ghasemi, A.: Robust portfolio selection problems: A comprehensive review. Operational Research **22** (2022). https://doi.org/10.1007/s12351-022-00690-5

2. Goldfarb, D., Iyengar, G.: Robust portfolio selection problems. Mathematics of Operations Research **28**(1), 1–38 (2003). https://doi.org/10.1287/moor.28.1.1.14260
3. Gorissen, B.L., Yanıkoğlu, İ., den Hertog, D.: A practical guide to robust optimization. Omega **53**, 124–137 (2015). https://doi.org/10.1016/j.omega.2014.12.006
4. Gurobi Optimization, LLC: Gurobi Optimizer Reference Manual (2024), https://docs.gurobi.com/projects/optimizer/en/current/index.html, last accessed 2026-02-10
5. Ma, Y., Han, R., Wang, W.: Portfolio optimization with return prediction using deep learning and machine learning. Expert Systems with Applications **165**, 113973 (2021). https://doi.org/10.1016/j.eswa.2020.113973
6. Markowitz, H.: Portfolio selection. The Journal of Finance **7**(1), 77–91 (1952). https://doi.org/10.1111/j.1540-6261.1952.tb01525.x
7. Zymler, S., Rustem, B., Kuhn, D.: Robust portfolio optimization with derivative insurance guarantees. European Journal of Operational Research **210**(2), 410–424 (2011). https://doi.org/10.1016/j.ejor.2010.09.027

The weak k-metric dimension of $K_n \times K_n$

Mohammad Farhan[1][0009−0000−2381−7919], Dorota Kuziak[1][0000−0001−9660−3284], and Ismael G. Yero[2][0000−0002−1619−1572]

[1] Departamento de Estadística e Investigación Operativa, Universidad de Cádiz, Algeciras Campus, Spain
[2] Departamento de Matemáticas, Universidad de Cádiz, Algeciras Campus, Spain
{mohammad.farhan, dorota.kuziak, ismael.gonzalez}@uca.es

Abstract. The weak k-metric dimension of a graph is roughly understood as the cardinality of a smallest set of vertices S of the graph with the property of uniquely recognizing all the vertices of the graph throughout summations of differences of distances to the vertices of S. The weak k-metric dimension of the direct product of two isomorphic complete graphs is considered in this work. Specifically, the value of such parameter is computed for almost all possibilities of these products and a bound is provided in the remaining case.

Keywords: weak k-metric dimension · weak k-resolving sets · direct product graphs.

1 Introduction

The metric dimension of graphs is a classical topic in graph theory whose origin is somehow coming from a related concept in the area of metric spaces (see [1]). The specific studies for graphs are understood to be initiated in the two independent works [2, 3]. However, the popularity of this topic was indeed motivated by the article [4], published at the beginning of this century. The investigations on this topic cover a very wide range of lines including classical combinatorial ones, together with other more algorithmic or applied ones.

Given a connected graph $G = (V(G), E(G))$, a set of vertices $S \subseteq V(G)$ is a *resolving set* for G if for each two vertices $x, y \in V(G)$ there is a vertex $w \in S$ such that $d_G(x, w)$ differs from $d_G(y, w)$, where $d_G(u, v)$ represents the length of a shortest path joining u and v. Such length is known as the *distance* between u and v. A resolving set of the smallest possible cardinality is called a *metric basis* of G, and its cardinality is known as the *metric dimension* of G, usually denoted by $\dim(G)$.

The weak k-metric dimension of graphs was presented in [5] as a variation of the classical metric dimension concept that involves a resolving related set of vertices in which the identification will be made by the whole set. Consider a set $S \subseteq V(G)$, and two vertices $x, y \in V(G)$. Let

$$\Delta_S(x, y) = \sum_{z \in S} |d_G(x, z) - d_G(y, z)|.$$

Given an integer $k \geq 1$, the set S is known as a *weak k-resolving set* for G if $\Delta_S(x,y) \geq k$ for each two vertices $x, y \in V(G)$. In the notation $\Delta_S(x,y)$, if $S = V(G)$, then we simply write $\Delta(x,y)$. In this sense, the weak k-resolving with the smallest possible cardinality is called a *weak k-metric basis* of G, and its cardinality is known as the *weak k-metric dimension* of G which is denoted by $\mathrm{wdim}_k(G)$. As was already noted in the seminal work [5], the value of k is bounded. We denote by $\kappa(G)$ the largest integer k such that G contains a weak k-resolving set. In [5], it was proved that indeed

$$\kappa(G) = \min\{\Delta(x,y) \ : \ x, y \in V(G)\}. \tag{1}$$

The weak k-metric dimension of graphs has been studied in the first work [5] for some classical graphs classes including trees and grid graphs (Cartesian products of paths). Further on, in [6], the parameter was also considered for the Cartesian product of two complete graphs. In this article, we aim to compute the weak k-metric dimension of the direct product of two isomorphic complete graphs. Given two graphs G and H, the *direct product graph* $G \times H$ is the graph with vertex set $V(G) \times V(H)$, where two vertices $(g,h), (g',h')$ are adjacent if $gg' \in E(G)$ and $hh' \in E(H)$. In order to simplify our notation, given an integer $t \geq 1$, we write $[t] = \{1, \ldots, t\}$. Also, we assume along the exposition that $V = V(K_n \times K_n) = [n] \times [n]$ to denote the vertex set of $K_n \times K_n$.

2 Results

To better understand the structure of $K_n \times K_n$, we define the two sets $L_i = \{(i,j) : j \in [n]\}$ and $L^j = \{(i,j) : i \in [n]\}$ as *vertical* and *horizontal layers* with respect to the vertex $(i,j) \in V$, respectively. Additionally, we define $L_i^j = L_i \cup L^j$. Two vertices $x, y \in V$ are said to *lie in the same layer* if they share a horizontal layer or a vertical layer; otherwise, they *lie in different layers*. Now, the distance between two vertices of $K_n \times K_n$ is as follows.

$$d_G(x,y) = \begin{cases} 1, & \text{if } x, y \text{ lie in different layers;} \\ 2, & \text{otherwise.} \end{cases}$$

From this distance formula, we can compute $\Delta_S(x,y)$ as follows.

Proposition 1. *Let $n \geq 4$, $\emptyset \neq S \subseteq V$, and take distinct vertices $x = (i,j), y = (i',j')$. Let $a_r = |L_r \cap S|$ and $b_s = |L^s \cap S|$ for $r \in \{i, i'\}$ and $s \in \{j, j'\}$.*

1. *If x, y lie in the same horizontal layer, then*

$$\Delta_S(x,y) = a_i + a_{i'} + |\{x,y\} \cap S|.$$

2. *If x, y lie in different layers and set $z_1 = (i',j)$ and $z_2 = (i,j')$, then*

$$\Delta_S(x,y) = a_i + a_{i'} + b_j + b_{j'} - |\{x,y\} \cap S| - 2|\{z_1, z_2\} \cap S|.$$

We first investigate the value of $\kappa(K_n \times K_n)$ in the following theorem.

Theorem 1. *For every integer $n \geq 3$, $\kappa(K_n \times K_n) = \begin{cases} 6, & n = 3; \\ 2n+2, & n \geq 4. \end{cases}$*

Proof. Let $x, y \in V$ be distinct. If x, y lie in the same layer, then Proposition 1 implies $\Delta(x, y) = 2n+2$. Otherwise, $\Delta(x, y) = 4n-6$. Hence, using (1), we have $\kappa(K_n \times K_n) = \min\{2n+2, 4n-6\}$ which yields the result.

Based on these values of κ, we are able to present the computation of weak k-metric dimension of $K_n \times K_n$ for the appropriate values of k. We begin by examining the large values for k and one particular case, where the techniques used are rather simple and require less effort.

Theorem 2. *For every integer $n \geq 4$,*

$$\mathrm{wdim}_k(K_n \times K_n) = \begin{cases} n^2, & k \in \{2n+1, 2n+2\}; \\ n^2-1, & k = 2n; \\ n^2-n, & k = 2n-1, \quad n \geq 5; \\ 13, & (n,k) = (4,7). \end{cases}$$

Proof. We analyze each case separately, depending on the values of k.

Case 1: $k \in \{2n+1, 2n+2\}$. Let S be a weak k-resolving set of $K_n \times K_n$ where $k \in \{2n+1, 2n+2\}$. Suppose there exists $x = (i, j) \in V \setminus S$. Then, for any $y \in L_i^j \setminus \{x\}$, $\Delta_S(x, y) = \Delta(x, y) - \Delta_x(x, y) = (2n+2) - 2 = 2n$, a contradiction. Thus, $S = V$ and $\mathrm{wdim}_k(K_n \times K_n) = n^2$.

Case 2: $k = 2n$. Let $S = V \setminus \{(1, 1)\}$. Since $\Delta_s(x, y) \leq 2$ for any triple $x, y, s \in V$, Theorem 1 gives $\Delta_S(x, y) = \Delta(x, y) - \Delta_{(1,1)}(x, y) \geq (2n+2) - 2 = 2n$. Hence, S is a weak $2n$-resolving set, and so $\mathrm{wdim}_{2n}(K_n \times K_n) \leq n^2 - 1$. Now, suppose that there is a weak $2n$-resolving set S of size at most $n^2 - 2$. Take any pair $(i, j), (i', j') \in V \setminus S$. If the two vertices lie in the same layer, then $\Delta_S((i, j), (i', j')) \leq 2n - 2 < 2n$. On the contrary, if the two vertices lie in different layers, then $\Delta_S((i, j), (i, j')) \leq 2n - 1 < 2n$. Both situations lead to a contradiction. Therefore, $\mathrm{wdim}_{2n}(K_n \times K_n) \geq n^2 - 1$.

Case 3: $k = 2n - 1$. The case $(n, k) = (4, 7)$ can be easily found by computer, where the set $S = V \setminus \{(1, 2), (2, 3), (3, 4)\}$ serves as its weak k-metric basis. Let $n \geq 5$ and $S = V \setminus D$ where $D = \{(i, i) : i \in [n]\}$. Hence, by construction, each layer has exactly $n - 1$ vertices in S. Let $x, y \in V$ be distinct. We first assume that x, y lie in the same layer. Our construction implies at least one of x, y is in S. Proposition 1 implies

$$\Delta_S(x, y) = a_i + a_{i'} + |\{x, y\} \cap S| \geq 2(n-1) + 1 = 2n - 1.$$

Now if x, y lie in different layers, then Proposition 1 implies

$$\Delta_S(x, y) = 4(n-1) - |\{x, y\} \cap S| - 2|\{z_1, z_2\} \cap S| \geq 4(n-1) - 2 - 2(2) \geq 2n - 1$$

since $n \geq 5$. Thus, S is a weak $(2n-1)$-resolving set, and so $\mathrm{wdim}_{2n-1}(K_n \times K_n) \leq n^2 - n$. If instead $|S| \leq n^2 - n - 1$, then some layer contains at least two vertices outside S, giving $\Delta_S(x, y) \leq 2n - 2$, a contradiction. Therefore, $\mathrm{wdim}_{2n-1}(K_n \times K_n) = n^2 - n$.

We continue by computing $\mathrm{wdim}_k(K_n \times K_n)$ for the small cases of k since they behave differently from the other larger values of k, and indeed require more technical arguments.

Theorem 3. *For every integer $n \geq 4$,*

$$\mathrm{wdim}_k(K_n \times K_n) = \begin{cases} n + \left\lceil \frac{n}{3} \right\rceil, & k = 2; \\ 2n, & k = 3. \end{cases}$$

For the case $k = 4$, we still only have an upper bound, which is

$$\mathrm{wdim}_4(K_n \times K_n) \leq 2n + 1 + \left\lfloor \frac{n}{4} \right\rfloor.$$

Finally, to complete our work, we present the general contribution for the remaining cases of k, which is addressed next.

Theorem 4. *For every integer $n \geq 4$ and $1 \leq t \leq n - 2$,*

$$\mathrm{wdim}_k(K_n \times K_n) = \begin{cases} n^2 - n - 1, & k = 2n - 2; \\ n^2 - tn, & k = 2n - 2t, \quad t \geq 2; \\ n^2 - tn - \left\lfloor \frac{n}{t+1} \right\rfloor, & k = 2n - 2t - 1. \end{cases}$$

Acknowledgments. The authors have been partially supported by "Ministerio de Ciencia, Innovación y Universidades" through the grant PID2023-146643NB-I00.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Blumenthal, L. M.: Theory and applications of distance geometry. Oxford University Press (1953)
2. Slater, P. J.: Leaves of trees. Cong. Numer. **14**, 549–559 (1975)
3. Harary, F., Melter, R. A.: On the metric dimension of a graph. Ars Combin. **2**, 191–195 (1976)
4. Chartrand, G., Eroh, L., Johnson, M. A., Oellermann, O. R.: Resolvability in graphs and the metric dimension of a graph. Discrete Appl. Math. **105**(1-3), 99–113 (2000)
5. Peterin, I., Sedlar, J., Škrekovski, R., Yero, I. G.: Resolving vertices of graphs with differences, Comput. Appl. Math. **43** article 275, (2024)
6. Fernandez, E., Klavžar, S., Kuziak, D., Muñoz-Márquez, M., Yero, I. G.: On the weak k-metric dimension of Hamming graphs. arXiv:2505.19642 [math.CO] (Mon, 26 May 2025)

Preference-Based Combinatorial Optimization

Malek Mouhoub[1] [0000−0001−7381−1064]

Department of Computer Science, University of Regina, Regina SK, Canada
mouhoubm@uregina.ca

Abstract. Combinatorial applications, such as timetabling, resource allocation, scheduling, and planning, consist of finding a good/best consistent scenario that satisfies a set of constraints while optimizing some objectives, including user preferences. In addition, constraints and objectives might not be explicitly defined and often come with uncertainty due to a lack of knowledge, missing information, or variability caused by external events. Finally, in some applications, these constraints and objectives can be temporal, spatial, or both. In the latter case, we are dealing with entities that occupy a given position in time and space.
To overcome the challenges we face when solving a combinatorial problem under the decision maker's preferences, we propose a methodology that explores both declarative and data-driven approaches. The proposed methodology is based on the Constraint Satisfaction Problem (CSP) paradigm and its variants. Solving techniques include both exact methods and metaheuristics. Exact methods include the backtracking algorithm and its variants. Constraint propagation and variable/value ordering heuristics are applied to improve the performance of backtracking in practice. Metaheuristics are good alternatives that trade time efficiency for the quality of the solutions returned. They include Stochastic Local Search (SLS) methods and nature-inspired techniques. We consider cases where constraint problems occur in dynamic environments and situations where some of the relevant information is incomplete/uncertain. We will also present extensions of CSPs to quantitative preferences (soft constraints) and conditional qualitative preferences. Finally, to deal with requirements and desires that are not explicitly defined, we will explore different constraint acquisition and preference learning algorithms.

Keywords: Constraint Satisfaction · Preference Reasoning · Combinatorial Optimization · Multi-Criteria Decision Making · Metaheuristics · Constraint Acquisition · Preference Learning.

1 Introduction

A wide variety of real-life applications require making decisions that meet a set of requirements while optimizing conflicting objectives and satisfying (societal) preferences as much as possible. This process, referred to as Multi-Criteria Decision Making (MCDM), is achieved through Preference-Centric combinatorial optimization with the goal of seeking a set of optimal trade-offs that balance cost-benefit objectives and preferences with fairness, representation, and social values.

Moreover, real-world scenarios do not operate under idealized assumptions of environmental stability. In fact, constraints, preferences, and objectives often come with uncertainty due to a lack of knowledge, missing or incorrect information, or variability caused by external and unpredictable events resulting from operating in highly dynamic environments [22]. Finally, in some applications, such as timetabling, urban planning, and robot motion planning, these constraints and objectives can be temporal, spatial, or both. In this latter case, we are dealing with entities occupying a given position in time and space. To achieve these goals, we propose a methodology that explores declarative and data-driven approaches for both the modeling and solving phases. Modeling a given combinatorial application is a tedious task that requires strong expertise in Constraint Programming. Therefore, we propose automating this process by learning constraints [4, 16, 10, 27], objectives [13], and (societal) preferences of the problem [20, 21, 2, 3, 14]. We consider both active learning based on a "human-in-the-loop" approach and passive learning from historical data following a data-driven strategy. The latter can be relevant for applications such as scheduling and timetabling, where past schedules and timetables are available and will be used as positive examples to learn from. The target models to learn are the constraint and preference networks. We use the Constraint Satisfaction Problem (CSP) paradigm [12] and its variants to represent both a two-level and a multi-level satisfiability, respectively, corresponding to hard and soft constraints. The latter are used to capture objectives and quantitative preferences. Examples of soft constraint networks include the C-Semiring-based CSP (SCSP) [5] and the Valued CSP (VCSP) based on a totally ordered commutative monoid. Graphical models representing preferences include Conditional Preference networks (CP-nets) [6] and Lexicographic Preference trees (LP-trees) [18] to capture qualitative and conditional preferences. In addition, to express the relative importance between variables, we rely on the Tradeoffs-enhanced Conditional Preference Network (TCP-net) [28, 7]. The above graphical models are extended to their respective probabilistic or possible variants to capture different forms of uncertainty through either the probability theory or the possibility theory [1, 9, 8, 15]. In case the relative importance of the problem objectives can be quantified, the related weights are acquired via active or passive learning. The combinatorial problem will then be converted into an optimization problem with one single objective defined as a weighted sum.

To solve a combinatorial problem using a declarative approach, we rely on exact and approximate methods (metaheuristics). Exact methods include a variant of the Branch and Bound algorithm augmented with constraint propagation and variable ordering heuristics to improve its efficiency in practice [23]. Metaheuristics [17, 26, 19] are good alternatives that trade time efficiency for the quality of the returned solution. Metaheuristics include evolutionary algorithms such as the Non-Dominated Sorting Genetic Algorithm (NSGA), the Pareto-dominance Evolutionary Algorithm, the Strength Pareto Evolutionary Algorithm (SPEA), and the Multi-Objective Evolutionary Algorithm Based on Decomposition (MOEA/D) [11, 24, 13, 25]. We will also explore other nature-inspired techniques adapted for multiple conflicting objectives such as the Multi-

Objective Particle Swarm Optimization (MOPSO) and the Multi-Objective Ant Colony Optimization (MOACO). For each technique adopted, we will consider a surrogate model to efficiently predict/discover trade-off solutions by approximating the objective functions instead of evaluating them directly.

Solving a combinatorial problem using a data-driven approach consists of approximating the optimal trade-offs by learning from empirical data corresponding to simulations or historical records. This end-to-end machine learning method builds a pipeline to learn to solve a combinatorial problem directly from data, without explicitly modeling constraints, objectives, and preferences.

The proposed methodology has the potential to improve efficiency, accuracy, scalability, fairness, and sustainability, and to play a vital role in balancing environmental, social, and economic concerns. In particular, real-world problems such as reactive scheduling and planning, timetabling, resource allocation, configuration, and transportation directly impact the environment and people's lives, making socially and environmentally responsible decisions imperative. Neglecting to embed social and environmental dimensions in these contexts can result in inequitable or even harmful outcomes.

References

1. Ahmed, S., Mouhoub, M.: Probabilistic TCP-net. In: Canadian Conference on Artificial Intelligence. pp. 293–304. Springer (2017)
2. Alanazi, E., Mouhoub, M., Zilles, S.: The complexity of exact learning of acyclic conditional preference networks from swap examples. Artificial Intelligence **278**, 103182 (2020)
3. Allen, T.E., Siler, C., Goldsmith, J.: Learning tree-structured CP-nets with local search. pp. 8–13. The Thirtieth International Flairs Conference (2017)
4. Belaid, M.B., Belmecheri, N., Gotlieb, A., Lazaar, N., Spieker, H.: Query-driven qualitative constraint acquisition. Journal of Artificial Intelligence Research **79**, 241–271 (2024)
5. Bistarelli, S., Montanari, U., Rossi, F., Schiex, T., Verfaillie, G., Fargier, H.: Semiring-based CSPs and valued CSPs: Frameworks, properties, and comparison. Constraints **4**(3), 199–240 (1999)
6. Boutilier, C., Brafman, R.I., Domshlak, C., Hoos, H.H., Poole, D.: CP-nets: A Tool for Representing and Reasoning with Conditional Ceteris Paribus Preference Statements. Journal of Artificial Intelligence Research **21**, 135–191 (2004)
7. Brafman, R.I., Domshlak, C., Shimony, S.E.: On graphical modeling of preference and importance. Journal of Artificial Intelligence Research pp. 389–424 (2006)
8. Cornelio, C.: Dynamic and Probabilistic CP-nets. Msc thesis, Univ. of Padova (2012)
9. De Filippo, A., Lombardi, M., Milano, M.: Methods for off-line/on-line optimization under uncertainty. Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (Jul 2018). https://doi.org/10.24963/ijcai.2018/177
10. De Raedt, L., Passerini, A., Teso, S.: Learning constraints from examples. In: Thirty-Second AAAI Conference On Artificial Intelligence. pp. 7965–7970. Assoc Advancement Artificial Intelligence (2018)

11. Deb, K., Sindhya, K., Hakanen, J.: Multi-objective optimization. In: Decision sciences, pp. 161–200. CRC Press (2016)
12. Dechter, R.: Constraint Processing. Morgan Kaufmann (2003)
13. Defresne, M., Mandi, J., Guns, T.: Preference elicitation for multi-objective combinatorial optimization with active learning and maximum likelihood estimation (2025), https://arxiv.org/abs/2503.11435
14. Drummond, J., Boutilier, C.: Elicitation and approximately stable matching with partial preferences. In: Twenty-Third International Joint Conference on Artificial Intelligence. pp. 97–105 (2013)
15. El Fidha, S., Mouhoub, M., Amor, N.B., Alanazi, E.: Representing and reasoning with constrained PCP-nets. In: 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC). pp. 1007–1012. IEEE (2017)
16. Gnecco, G., Gori, M., Melacci, S., Sanguineti, M.: Learning with mixed hard/soft pointwise constraints. IEEE Transactions on Neural Networks and Learning Systems **26**(9), 2019–2032 (2015)
17. Huang, C., Li, Y., Yao, X.: A survey of automatic parameter tuning methods for metaheuristics. IEEE Transactions on Evolutionary Computation **24**(2), 201–216 (2019)
18. Huelsman, M., Truszczynski, M.: A Lexicographic Strategy for Approximating Dominance in CP-nets. In: Barták, R., Bell, E. (eds.) Proceedings of the Thirty-Third International Florida Artificial Intelligence Research Society Conference. pp. 59–74. AAAI Press (2020), https://aaai.org/ocs/index.php/FLAIRS/FLAIRS20/paper/view/18409
19. Korani, W., Mouhoub, M.: Review on nature-inspired algorithms. In: Operations research forum. vol. 2, p. 36. Springer (2021)
20. Labernia, F., Zanuttini, B., Mayag, B., Yger, F., Atif, J.: Online learning of acyclic conditional preference networks from noisy data. In: 2017 IEEE International Conference on Data Mining (ICDM). pp. 247–256. IEEE (2017)
21. Liu, S., Liu, J.: CP-nets structure learning based on mRMCR principle. IEEE Access **7**(1), 121482–121492 (2019)
22. Mouhoub, M.: Systematic versus non systematic techniques for solving temporal constraints in a dynamic environment. AI Communications **17**(4), 201–211 (2004)
23. Mouhoub, M., Jafari, B.: Heuristic techniques for variable and value ordering in csps. In: Proceedings of the 13th annual conference on Genetic and evolutionary computation. pp. 457–464 (2011)
24. Muhuri, P.K., Biswas, S.K.: Bayesian optimization algorithm for multi-objective scheduling of time and precedence constrained tasks in heterogeneous multiprocessor systems. Applied Soft Computing pp. 1–27 (2020)
25. Purshouse, R.C., Deb, K., Mansor, M.M., Mostaghim, S., Wang, R.: A review of hybrid evolutionary multiple criteria decision making methods. In: 2014 IEEE congress on evolutionary computation (CEC). pp. 1147–1154. IEEE (2014)
26. Talbi, E.G.: Metaheuristics: from design to implementation, vol. 74. John Wiley & Sons (2009)
27. Tsouros, D.C., Stergiou, K., Bessiere, C.: Structure-driven multiple constraint acquisition. In: Schiex, T., de Givry, S. (eds.) Principles and Practice of Constraint Programming. pp. 709–725. Springer International Publishing (2019)
28. Zhang, S., Mouhoub, M., Sadaoui, S.: Integrating tcp-nets and csps: The constrained tcp-net (ctcp-net) model. In: International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems. pp. 201–211. Springer (2015)

Time-Dependent Quality Degradation in Olive Harvest-to-Mill Scheduling: A Mixed-Integer Programming Approach

Haluk Eliş [1][0000-0003-3748-841X], Ozan Pembe[1][0009-0005-4151-9997] and A. Özgür Toy[1][0000-0003-1603-6860]

[1] Yaşar University, İzmir 35100, Türkiye
{haluk.elis, ozan.pembe, ozgur.toy}@yasar.edu.tr

Abstract. Olive oil quality is highly sensitive to the time elapsed between harvest and processing. In practice, the commonly referenced 24-hour guideline is often treated as a rigid deadline, although quality deterioration occurs gradually and is strongly affected by congestion at the mill. We propose a mixed-integer programming model that jointly determines harvesting and processing schedules under harvesting and milling capacity constraints. Delay-related quality loss is represented as an economic penalty activated beyond a tolerance period, replacing hard feasibility deadlines with a soft and interpretable trade-off. Computational experiments show that capacity expansion does not affect delay linearly: once congestion is sufficiently relieved, additional capacity appears to improve coordination rather than utilization.

Keywords: harvest scheduling, mixed-integer programming, quality degradation, capacity constraints, perishable production.

1 Introduction

Olive oil quality is highly sensitive to the time elapsed between harvest and processing at the olive mill. Once olives are detached from the tree, biochemical and microbiological processes begin, leading to irreversible quality losses if processing is delayed. For this reason, both academic studies and industry practice emphasize rapid processing, often summarized by the commonly cited 24-hour rule, which recommends milling within 24 hours after harvest ([1], [2], [3]).

Despite its practical relevance, the 24-hour rule is often treated as a rigid operational threshold. In reality, quality degradation is gradual rather than discontinuous, and enforcing a hard deadline may generate infeasible or economically inefficient schedules, particularly during peak harvest periods when harvesting capacity, transportation, and mill throughput are constrained.

This paper develops a mixed-integer programming model for integrated harvest-to-mill scheduling. The model jointly assigns each batch to a harvest period and a processing start period, subject to harvesting capacity, transportation feasibility, and multi-period mill capacity constraints. Delay-related quality loss is represented as an

economic penalty beyond a tolerance period, replacing rigid processing deadlines with a soft and economically interpretable trade-off.

The contribution is twofold. First, harvest and processing decisions are formulated within a unified capacity-constrained scheduling model, allowing mill congestion to be internalized at the harvest planning stage. Second, the commonly used deadline-based representation of quality preservation is replaced by an endogenous delay penalty that captures gradual deterioration.

2 Literature Review

Operations Research methods have been widely used in agricultural production planning, including crop allocation, harvest scheduling, and agri-food supply chain coordination. Early work emphasized farm-level planning and resource allocation, while more recent studies have focused on integrated production, harvesting, storage, and distribution decisions in perishable supply chains [4].

Ahumada and Villalobos [5] review planning models in agri-food supply chains and emphasize the importance of integrating production and distribution decisions under perishability. Similarly, Taşkıner and Bilgen [6] classify optimization models for harvest and production planning and highlight perishability as a central modeling issue. However, perishability is often represented through fixed shelf-life limits, time windows, or simplified quality constraints.

In olive production, Herrera-Cáceres et al. [7] propose an optimization model for olive harvest planning that incorporates operational and climatic constraints. In broader fruit-harvest settings, Gómez-Lagos et al. [8] develop a tactical harvest planning model for multiple orchards with maturity-related penalties. These studies demonstrate the value of optimization in harvest planning, but quality effects are generally modeled indirectly or through stage-dependent penalties rather than as explicit delay-dependent economic degradation.

The present study differs by explicitly linking harvest timing, mill congestion, and post-harvest quality loss in a single scheduling formulation. Instead of enforcing a hard processing deadline, the proposed model represents quality degradation through a penalty activated beyond a tolerance period. This allows the model to capture the economic trade-off between operational feasibility and quality preservation.

3 Mathematical Model

Let I denote the set of olive batches, T the set of harvest periods, K the set of processing start periods, and S the set of periods used to track mill capacity. Parameter a_i is the harvesting capacity required by batch i, and H_t is the available harvesting capacity in period t. Transportation time from plot i to the mill is denoted by δ_i. Batch i requires p_i processing periods and consumes mi units of mill capacity per period. The mill capacity in period s is M_s, c_i is the penalty per unit of excess delay, and τ_0 is the tolerance time.

The binary variable z_{itk} equals 1 if batch i is harvested in period t and starts processing in period k. Variables D_i and u_i denote the harvest-to-processing delay and the excess delay beyond the tolerance time.

The model is:

minimize $\sum_{i \in I} c_i u_i$

subject to

$\sum_{t \in T} \sum_{k \in K} z_{itk} = 1, \ \forall \ i \in I$

$z_{itk} = 0, \ \forall \ i \in I$ and all (t, k) such that $k < t + \delta_i$

$\sum_{i \in I} \sum_{k \in K} a_i . z_{itk} \leq H_t, \ \forall \ t \in T$

$\sum_{i \in I} \sum_{t \in T} \sum_{k \in K:\, k \leq s \leq k + p_{i-1}} m_i . z_{itk} \leq M_s , \ \forall \ s \in S$

$D_i = \sum_{t \in T} \sum_{k \in K} (k - t) \, z_{itk}, \ \forall \ i \in I$

$u_i \geq D_i - \tau_0, \ \forall \ i \in I$

$z_{itk} \in \{0,1\}, \ \forall \ i \in I, t \in T, k \in K$

$D_i \geq 0, \ \forall \ i \in I$

$u_i \geq 0, \ \forall \ i \in I$

The first constraint assigns each batch exactly once. The second enforces transportation feasibility. The third and fourth impose harvesting and multi-period mill capacity limits. The remaining constraints define delay and excess delay. Thus, the model jointly captures operational feasibility, congestion, and the economic cost of delayed processing.

4 Computational Setting and Summary Results

We conduct computational experiments on synthetic instances representing olive batches served by a single mill over a finite planning horizon. Time is discretized into hourly periods. The experiments vary the mill capacity level and the tolerance threshold. Mill capacity is scaled by $C \in \{1.0,1.3,1.6,2.0\}$, while the tolerance threshold τ_0 takes values in $\{14,16,18,20,24\}$. All instances are solved to optimality using Gurobi. The resulting performance indicators, averaged over tolerance levels for each capacity level, are summarized in Table 1.

Table 1. Summary of computational results by capacity level.

Capacity level	Avg. mean delay	Avg. late fraction	Mill utilization
1.0	41.03	0.60	0.750
1.3	41.03	0.60	0.577
1.6	20.21	0.37	0.469
2.0	12.21	0.15	0.375

The results indicate a nonlinear capacity effect. When capacity increases from C=1.0 to C=1.3, mean delay remains essentially unchanged, suggesting that the system remains in a congested regime. At C=1.6, delays decrease sharply, the fraction of late batches is reduced, and harvest and processing decisions become better synchronized. Further increasing capacity to C=2.0 reduces delay further, but with diminishing marginal improvement.

A notable observation is that mill utilization decreases as capacity increases. This does not indicate inefficient use of resources. Rather, the model uses additional capacity as flexibility: instead of filling the mill continuously, it spreads processing decisions to reduce waiting and quality-loss exposure. Thus, additional capacity improves coordination rather than utilization.

5 Conclusion

This paper presents an integrated mixed-integer programming model for olive harvest-to-mill scheduling under time-dependent quality degradation. The model replaces rigid processing deadlines with a delay-dependent economic penalty and internalizes mill congestion within harvest planning. Computational results show that capacity has a threshold effect: once congestion is sufficiently relieved, delays fall sharply and the system becomes better coordinated. Future work may extend the framework to mobile processing capacity and uncertain harvest yields.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Stefanoudaki-Katzouraki, E.: Factors affecting olive oil quality. PhD thesis, Cardiff University (2004)

2. Vossen, P.: Olive oil: History, production and characterization of the world's classic oils. HortScience 42, 1093–1100 (2007)
3. Rotondi, A., Morrone, L., Bertazza, G., Neri, L.: Effect of duration of olive storage on chemical and sensory quality of extra virgin olive oils. Foods 10, 2296 (2021)
4. Lowe, T.J., Preckel, P.V.: Decision technologies for agribusiness problems: A brief review of selected literature and a call for research. Manufacturing & Service Operations Management 6(3), 201–208 (2004)
5. Ahumada, O., Villalobos, J.R.: Application of planning models in the agri-food supply chain: A review. European Journal of Operational Research 195, 1–20 (2009)
6. Taşkıner, T., Bilgen, B.: Optimization models for harvest and production planning in agri-food supply chain: A systematic review. Logistics 5(3), 52 (2021)
7. Herrera-Cáceres, C., Pérez-Galarce, F., Álvarez-Miranda, E., Candia-Véjar, A.: Optimization of the harvest planning in the olive oil production: A case study in Chile. Computers and Electronics in Agriculture 141, 147–159 (2017)
8. Gómez-Lagos, J.E., González-Araya, M.C., Soto-Silva, W.E., Rivera-Moraga, M.M.: Optimizing tactical harvest planning for multiple fruit orchards using a metaheuristic modeling approach. European Journal of Operational Research 290(1), 297–312 (2021)

A DRONE-BASED MULTI-WAREHOUSE VEHICLE ROUTING MODEL FOR POST-DISASTER EMERGENCY SUPPLY DISTRIBUTION

Onat Düzgün[1], Selin Özpeynirci[1], Arya Sevgen Misiç[1], Ozan Baran Demirçivi[1]

[1] İzmir University of Economics, Department of Industrial Engineering,
Sakarya Caddesi No:156, 35330 Balçova, İzmir, Türkiye
onat.duzgun@ieu.edu.tr, selin.ozpeynirci@ieu.edu.tr, arya.sevgen@ieu.edu.tr,
ozan.demircivi@ieu.edu.tr

Abstract. This paper presents a drone-based optimization model for post-disaster emergency supply distribution in urban areas where ground transportation may become inoperable. The problem is formulated as a capacitated multi-warehouse vehicle routing problem with drones, incorporating multiple drones, capacity limits, setup times, and multiple tours. Real-life geographical data from İstanbul are used, and aerial distances are adopted due to road inaccessibility. The mixed-integer linear programming model is solved for selected instances, showing effectiveness for moderate-sized problems but rapid growth in complexity for larger cases.

Keywords: Vehicle routing problem, Multi-warehouse routing, Unmanned aerial vehicles, Post-disaster logistics, Mixed-integer linear programming.

1 INTRODUCTION

Unmanned Aerial Vehicles (UAVs) are widely used in applications such as military operations, agriculture, mapping, and meteorology [1]. However, UAV-based last-mile delivery optimization in disaster-prone environments under realistic constraints remains limited in the literature [2]. Although UAV routing has been studied in logistics contexts [3], its application to post-disaster humanitarian relief is still underexplored. Motivated by this gap, this study investigates a UAV-based emergency supply distribution problem for post-earthquake scenarios.

Following a large-scale earthquake, road networks in cities such as İstanbul may become inoperable, limiting ground-based logistics. UAV-based systems offer an effective alternative for last-mile relief distribution. The problem considers multiple warehouses operating homogeneous UAVs under capacity and setup constraints, with the objective of minimizing the maximum delivery completion time. It is formulated as a capacitated multi-warehouse, multi-trip UAV routing problem and modeled as an NP-hard mixed-integer linear program.

2 MATHEMATICAL MODEL

This section presents the mixed-integer linear programming (MILP) formulation of the drone-based delivery problem. The objective of the proposed model is to minimize the

maximum completion time of all delivery operations while satisfying routing, capacity, and temporal constraints.

Sets

$W = set\ of\ warehouses$

$D = set\ of\ demand\ points$

$N = set\ of\ all\ nodes\ W \cup D$

$K = set\ of\ drones\ in\ warehouses$

$TR = set\ of\ maximum\ tours\ allowed$

Parameters

$dist_{ij} = Distance\ from\ node\ i\ to\ node\ j$

$V = Drone\ speed$

$time_{ij} = \frac{dist_{ij}}{V}$

$M = Very\ large\ positive\ constant$

$C = Drone\ capacity$

$setup = Setup\ time\ at\ warehouses$

$dem_j = Demand\ of\ node\ j \epsilon D$

Decision Variables

$x_{wktij} = 1, if\ drone\ k\ at\ warehouse\ w\ travels\ from\ node\ i\ to\ j\ at\ tour\ t;$

$0, otherwise.$

$load_{wktn} = load\ carried\ by\ drone\ k\ from\ warehouse\ w\ during\ tour\ t\ upon\ arrival\ at\ node\ n$

$T_{wktn} = arrival\ time\ of\ drone\ k\ from\ warehouse\ w\ at\ node\ n\ during\ tour\ t$

$T_{max} = maximum\ arrival\ time$

$y_{wkt} = 1, if\ drone\ k\ at\ \ warehouse\ w$ uses $tour$ t; $0, otherwise.$

Objective Function

$$Min\ T_{max} \tag{1}$$

Constraints

$$x_{wktij} \leq y_{wkt} \forall w \in W, \forall k \in K, \forall t \in TR, \forall i \in N, \forall j \in N. \tag{2}$$

$$y_{wkt} \leq y_{wk(t-1)}\ \forall w \in W, \forall k \in K, \forall t \in TR : t > 1. \tag{3}$$

$$\sum_{i \in W} \sum_{j \in D} x_{wktij} = y_{wkt}\ \forall w \in W, \forall k \in K, \forall t \in TR \tag{4}$$

$$\sum_{i \in D} \sum_{j \in W} x_{wktij} = y_{wkt}\ \forall w \in W, \forall k\ K, \forall t \in TR \tag{5}$$

$$\sum_{w \in W} \sum_{k \in K} \sum_{t \in TR} \sum_{i \in N} x_{wktij} = 1\ \forall j \in D \tag{6}$$

$$\sum_{w \in W} \sum_{k \in K} \sum_{t \in TR} \sum_{j \in N} x_{wktij} = 1\ \forall i \in D \tag{7}$$

$$\sum_{j\in N} x_{wktij} = \sum_{j\in N} x_{wktji} \;\; \forall w \in W, \forall k \in K, \forall t \in TR, \forall i \in D \quad (8)$$

$$\sum_{j\in N} x_{wktwj} = \sum_{i\in N} x_{wktiw}\,, \forall w \in W, \forall k \in K, \forall t \in TR \quad (9)$$

$$\sum_{i\in D} x_{wktwi} \leq 1, \quad \forall w \in W,\ \forall k \in K,\ \forall t \in TR \quad (10)$$

$$\sum_{i\in D} x_{wktiw} \leq 1, \quad \forall w \in W,\ \forall k \in K,\ \forall t \in TR \quad (11)$$

$$\text{load}_{wktw} = 0, \quad \forall w \in W,\ \forall k \in K,\ \forall t \in TR \quad (12)$$

$$\text{load}_{wkti} \leq C, \quad \forall w \in W,\ \forall k \in K,\ \forall t \in TR,\ \forall i \in N \quad (13)$$

$$\text{load}_{wktj} \geq \text{load}_{wkti} + \text{demand}_j - C\left(1 - x_{wktij}\right), \forall w \in W, \forall k \in K, \forall t \in TR, \forall i \in N, \forall j \in D \quad (14)$$

$$T_{wktw} \geq T_{wkti} + \text{time}_{iw} - M_{time}(1 - x_{wktiw}), \forall w \in W, \forall k \in K, \forall t \in TR, \forall i \in D \quad (15)$$

$$T_{wk1j} \geq \text{time}_{wj} - M_{\text{time}}\left(1 - x_{wk1wj}\right), \forall w \in W,\ \forall k \in K,\ \forall j \in D \quad (16)$$

$$T_{wktj} \geq T_{wk(t-1)w} + setup + time_{wj} - M_{time}\left(1 - x_{wktj}\right), \forall w \in W, \forall k \in K, \forall t \in TR\colon t > 1 \quad (17)$$

$$T_{wktj} \leq T_{max}, \forall w \in W,\ \forall k \in K,\ \forall t \in TR,\ \forall j \in D \quad (18)$$

$$x_{wktij} = 0, \forall t \in TR,\ \forall w \in W,\ \forall k \in K,\ \forall i\, W,\ \forall j \in N\colon i \neq w \quad (19)$$

$$x_{wktij} = 0, \forall t \in TR,\ \forall w \in W,\ \forall k \in K,\ \forall i \in N,\ \forall j \in W\colon j \neq \text{w} \quad (20)$$

$$x_{wktnn} = 0, \forall w \in W, \forall k \in K, \forall t \in TR, \forall n \in N \quad (21)$$

The objective function (1) minimizes the maximum delivery completion time T_{max}, representing the makespan of the emergency delivery operation. Constraints (2)–(5) define tour activation and warehouse departure/return consistency. Constraints (6)–(9) enforce routing feasibility and flow conservation. Constraints (10) and (11) prevent subtours. Capacity feasibility is ensured by Constraints (12)–(14), which initialize drone load at the warehouse, impose payload limits, and propagate load along the route. Arrival time calculations and linking delivery completion times to T_{max} are governed by Constraints (15)–(18). Finally, Constraints (19)–(21) eliminate infeasible arcs and self-loops.

3 NUMERICAL EXPERIMENTS

The proposed mixed-integer linear programming model was tested on a set of computational instances derived from real-life geographical data of İstanbul [4]. In Table 1, the problem instances are summarized according to the number of warehouses and demand points, and the reported values represent the average T_{max} and average CPU time over five randomly generated instances for each problem size, reflecting different configurations of drone availability and tour limits.

Table 1. Results of Computational Experiments

Warehouses	Demand points	Av. T_{max}	Av. CPU time
2	9	0.282	2.29
2	12	0.173	58.03
5	18	0.167	1496.90
3	12	0.248	153.08

All numerical experiments were conducted on a computer running Windows 11 Home 64-bit operating system. The system was equipped with a 12th Gen Intel® Core™ i7-12650H processor and 16 GB of RAM. IBM ILOG CPLEX was used as the optimization solver, and CPU time was recorded in seconds for all problem instances.

4 CONCLUSION

This study proposes a drone-based optimization model for post-disaster emergency logistics in urban areas where ground transportation may become inoperable. A capacitated multi-warehouse VRP with drones is formulated to minimize the maximum delivery completion time using real-life data from İstanbul. Computational results show that the mixed-integer linear programming model efficiently solves small- and medium-sized instances, while complexity increases significantly for larger cases, confirming the NP-hard nature of the problem. To address large-scale MP-hard instances, a simulated annealing-based metaheuristic is currently under development, and future work may incorporate battery constraints, no-fly zones, and stochastic demand.

References

1. Estrada, M.A.R., Ndoma, A.: The uses of unmanned aerial vehicles (UAVs) in the logistics of disaster management: A review. Journal of Humanitarian Logistics and Supply Chain Management 9(3), 375–392 (2019)
2. Rejeb, A., Rejeb, K., Simske, S.J., Treiblmaier, H.: Drones for supply chain management and logistics: A review and research agenda. International Journal of Logistics Research and Applications 26(6), 708–731 (2023)
3. Dorling, K., Heinrichs, J., Messier, G.G., Magierowski, S.: Vehicle routing problems for drone delivery. IEEE Transactions on Systems, Man, and Cybernetics: Systems 47(1), 70–85 (2016)
4. Kara, M.: Use of unmanned aerial vehicles for post-disaster response. M.Sc. Thesis, Graduate School, İzmir University of Economics, İzmir, Türkiye (2025)

Optimization of Shelter Placement in Post-Disaster Management*

Mehmet Melikşah Çalışkan[1], Cemal Bolat[1] and Didem Gözüpek[1]

Gebze Technical University, Department of Computer Engineering, Kocaeli, Türkiye
{m.caliskan2021, c.bolat2021, didem.gozupek}@gtu.edu.tr

Abstract. Post-disaster container cities often prioritize logistical speed over resident well-being, resulting in layouts where essential services are inaccessible. We propose a graph theoretic model that optimizes residential density while enforcing human-centric accessibility. We formulate an integer linear programming model that maximizes residential capacity while simultaneously satisfying multi-facility accessibility constraints based on international humanitarian standards. By integrating service distance thresholds, we ensure that every resident is within a critical walking distance of all required facilities, effectively translating the equitable accessibility concept into practice in a temporary housing context.

Keywords: Container City Design · Urban Planning · Facility Layout Problem · Post-Disaster Management · Optimization · Integer Linear Programming.

1 Introduction

Natural disasters, particularly earthquakes, present profound threats to human safety and infrastructure, often resulting in large scale displacement that necessitates the rapid deployment of container-based recovery settlements. However, traditional planning often defaults to rigid grids that maximize occupancy; nevertheless, it neglects the spatial distribution of essential services. This creates socially fragmented environments where residents lack easy access to essential services, exacerbating post-disaster trauma and logistic inefficiency.

The modern urban planning literature suggests that the layout of a city is a fundamental determinant of public health. Moreno et al. [2] introduce the 15 Minute City framework, advocating that all essential services must be reachable within a short walk to ensure urban resilience. In addition, Arslan et al. [1] focus on transforming cities into 15 Minute Cities. In the context of temporary housing, Afkhamiaghda et al. [3] argue that shelters should be treated as living environments rather than simple storage units. They highlight that current

* This work has been supported by the European Commission's Horizon Europe Research and Innovation programme through the Marie Skłodowska-Curie Actions Staff Exchanges (MSCA-SE) under Grant Agreement no.101182819 (COVER: (C)ombinatorial (O)ptimization for (V)ersatile Applications to (E)merging u(R)ban Problems)

models frequently fail because they do not guarantee service coverage. Thus, UNCHR published a shelter planning standard [4] that describes the walkability distance for each container type as well as their required sizes and capacities.

In this work, we formulate an integer linear program that maximizes the number of residential containers while ensuring service coverage. By incorporating spatial requirements and capacity limitations as constraints, our optimization model ensures that each residential unit is serviced within the distance thresholds established by the UNHCR shelter guidelines [4].

The UNHCR shelter guidelines [4] state that each service type necessitates distinct spatial and capacity requirements based on its operational scale. To address these standards, our optimization model explicitly accounts for the fact that different facility types require varying numbers of containers and possess unique service capacities. Accordingly, the model represents potential services as sets of adjacent containers. A cluster is categorized as a feasible group only when it satisfies the predefined UNHCR criteria.

2 Mathematical Formulation

We represent the potential layout of the container city as a complete graph $G = (V, E)$, where each vertex $v \in V$ corresponds to a physical location capable of hosting one container and each edge $e \in E$ represents a path between two vertices. The set of all possible different sized container types is denoted by T, where $r \in T$ represents the residential type, and $T \setminus \{r\}$ represents the set of essential service facilities (e.g., healthcare, water tap, school unit).

Table 1 and Table 2 describe the input and decision variables, respectively.

Table 1. Input Variables

C_t	The maximum capacity of a service facility of type $t \in T \setminus \{r\}$
S_t	The required number of containers for a group of type $t \in T$.
D_t	The maximum allowable walking distance between a residential container and a service facility of type $t \in T$
Δ_t	The maximum internal distance between a pair of vertices in a feasible group of type $t \in T$
G_t	The set of all feasible groups of type $t \in T$. A group $g \subseteq V$ is feasible if $\lvert g\rvert = S_t$ and the distance between any two vertices in g is $\leq \Delta_t$
$S_{v,t}$	The set of candidate groups of type $t \in T$ that are within the service distance D_t from vertex $v \in V$.

Table 2. Decision Variables

$x_{v,t}$	$=\begin{cases} 1; \\ 0; \end{cases}$	if vertex v is assigned container type $t \in T$ otherwise
$u_{g,t}$	$=\begin{cases} 1; \\ 0; \end{cases}$	if a group $g \in G_t$ of facility type t is used as an active service otherwise
$y_{v,g,t}$	$=\begin{cases} 1; \\ 0; \end{cases}$	if residential vertex v is assigned to be served by the specific group g of facility type $t \in T \setminus \{r\}$ otherwise

2.1 Objective Function

The objective is to maximize the total number of residential units while satisfying all service accessibility requirements:

$$\max \sum_{v \in V} x_{v,r} \tag{1}$$

2.2 Constraints

Each vertex can host at most one container type:

$$\sum_{t \in T} x_{v,t} \leq 1, \quad \forall v \in V \tag{2}$$

If a vertex is designated as residential ($x_{v,r} = 1$), it must be assigned to exactly one service group of any type of facility t within its service radius:

$$\sum_{g \in S_{v,t}} y_{v,g,t} = x_{v,r}, \quad \forall v \in V, \forall t \in T \setminus \{r\} \tag{3}$$

Each activated facility group g of type t can serve a limited number of residential units, defined by C_t:

$$\sum_{v \in V} y_{v,g,t} \leq C_t \cdot u_{g,t}, \quad \forall g \in G_t, \forall t \in T \setminus \{r\} \tag{4}$$

Equations (3) and (4) ensure the service coverage requirements highlighted by Afkhamiaghda et al. [3]. The number of residential containers assigned to the active group cannot exceed the capacity of that service type.

To maintain the integrity of each facility, every vertex v within a group g must be assigned the corresponding type t if that group is activated. For instance, all vertices constituting a specific active group must be designated as the same type:

$$u_{g,t} \leq x_{v,t}, \quad \forall t \in T \setminus \{r\}, \forall g \in G_t, \forall v \in g \tag{5}$$

To prevent overlapping groups and ensure exclusive membership, a vertex v assigned to type t must belong to exactly one active group of that type:

$$x_{v,t} = \sum_{g \in G_t : v \in g} u_{g,t}, \quad \forall v \in V, \forall t \in T \setminus \{r\} \tag{6}$$

Equations (5) and (6) maintain structural consistency by enforcing that every activated group comprises vertices of the same designated type. Furthermore, these constraints prevent any single vertex from participating in multiple groups.

3 Implementation and Results

We have evaluated the proposed model on a large-scale 35×35 grid comprising 1,225 container locations and approximately 400,000 potential edges with the types mentioned in Table 3. The proposed model successfully optimized a settlement layout containing 896 residential units. Given a designated capacity of five individuals per container [4], the resulting shelter layout accommodates a total population of 6,125 individuals. You can see our development source code. [6]

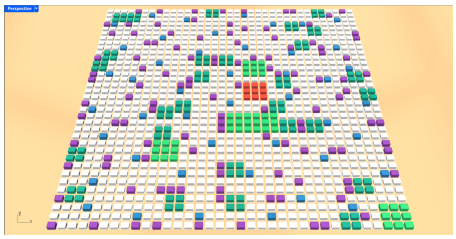

Fig. 1. The resulting 35×35 layout (896 residential units)

Table 3. Building Types and Visual Representation

Residence	: □	The primary residential unit.
Communal Latrine	: ■	Essential sanitation facility.
Shower	: ■	Personal hygiene unit.
Water Tap	: ■	Portable water access point.
Health Centre	: ■	Primary medical facility.
School	: ■	Educational and community space.

References

1. O. Arslan, G. Laporte: The 15-minute city concept: An operations research perspective and a research agenda (2025)
2. C. Moreno, Z. Allam, D. Chabaud, C. Gall and F. Pratlong: Introducing the 15 Minute City: Sustainability, Resilience and Place Identity. Smart Cities, 4(1), 93-111 (2021).
3. M. Afkhamiaghda, E. Elwakil, K. Afsari and R. Rapp: Factors affecting post-disaster temporary housing construction. Journal of Emergency Management, 19(4), 311-325 (2021).
4. UNHCR 2015c Camp Planning Standards (Planned Settlements) (2015)
5. J. Gehl: Cities for People. Island Press, Washington DC (2010).
6. C. Bolat, M. M. Çalışkan. (2026). `https://doi.org/10.5281/zenodo.17333020`

Subproblem Solving in Benders Decomposition for Affine Potential-Based Flow Problems with Topology Switching and Robustness Scenarios

Tim Donkiewicz[1 [ORCID]] and Oliver Gaul[1 [ORCID]]

Chair of Operations Research, RWTH Aachen University, Germany
{donkiewicz,gaul}@or.rwth-aachen.de

Potential-based flow networks [3,10,12] are used to model problems where node potentials $\pi_u \in \mathbb{R}$ and arc flows $x_{(u,v)} \in \mathbb{R}$ satisfy the potential equation $\pi_u - \pi_v = \mu_{(u,v)}^{-1}\psi_{(u,v)}(x_{(u,v)})$, with arc conductance $\mu_{(u,v)} \in \mathbb{R}$ and potential function $\psi_{(u,v)} : \mathbb{R} \to \mathbb{R}$. Examples are electrical circuits [4], gas pipelines [13], traffic assignment [8], and other applications [12]. Changes in topology can be modeled by switch arcs, making the problem NP-hard in general [9]. Switching has been combined with robustness against the outage of one or more edges [6,7]. Such problems are decomposable via Benders decomposition [2], where the master problem (MP) decides topology and then subproblems (SPs) evaluate the scenarios. Based on SP dual solutions, optimality or feasibility cuts are generated and added to the MP, guiding towards a globally optimal solution. Standard implementations solve SPs as linear programs, e.g., using simplex [11].

Contribution. We reduce problems with affine potential functions $\psi_{(u,v)}(x_{(u,v)}) = r_{(u,v)}x_{(u,v)} + s_{(u,v)}$, $r_{(u,v)} > 0$, $s_{(u,v)} \in \mathbb{R}$, used primarily in lossless DC power flow problems, to equivalent problems with $\psi_{(u,v)}(x_{(u,v)}) = x_{(u,v)}$. We present a fast method to obtain Benders optimality and feasibility cuts: SP primal solutions can be uniquely determined by a system of linear equations (SLE), solved using LU-factorization. Using the same factorization, we find an optimal dual solution. We exploit the similarity between scenarios by reusing factorizations via low-rank updates. The Benders iteration complexity for single-edge outages is cubic in the number of nodes. We demonstrate the method on security-constrained optimal transmission switching (SCOTS) instances, where runtime reductions of up to 98.80 % are observed compared to a standard LP-based approach.

Problem Statement. We consider graphs $(V, A_L \cup A_S)$ with nodes V, line arcs $A_L \subseteq V \times V$, disjoint from switch arcs $A_S \subseteq V \times V$. Each node $u \in V$ has potential $\pi_u \in \mathbb{R}$ and demand $b_u \in \mathbb{R}$ with $\sum_{u \in V} b_u = 0$. Its in- and outgoing arcs are denoted by $\delta^-(u), \delta^+(u) \subseteq A_L \cup A_S$. Each arc $(u, v) \in A_L$ has flow $x_{(u,v)} \in \mathbb{R}$, conductance $\mu_{(u,v)} > 0$, and satisfies the potential equation $\pi_u - \pi_v = \mu_{(u,v)}^{-1}\psi_{(u,v)}(x_{(u,v)})$, where $\psi_{(u,v)} = r_{(u,v)} \cdot x + s_{(u,v)}$, $r_{(u,v)} > 0$, $s_{(u,v)} \in \mathbb{R}$, is its potential function. Flows are bounded by $X_{(u,v)}^{\min}, X_{(u,v)}^{\max} \in \mathbb{R}$. They incur costs $c^{>} \in \mathbb{R}$ per unit surpassing the soft bounds $X_{(u,v)}^{>,\min}, X_{(u,v)}^{>,\max} \in \mathbb{R}$. Switch arcs $(u, v) \in A_S$ have binary state $z_{(u,v)} \in \{0, 1\}$. When on $(z_{(u,v)} = 1)$, potentials are equal $(\pi_u = \pi_v)$. Conservation for flows and demands holds at every node.

Reduction to Identity Function. For each line arc $(u,v) \in A_L$, we replace the potential function with the identity function $\mathrm{id}(x) = x$. We modify conductances, bounds and demands (denoted by superscript "id") to obtain an equivalent problem. It holds that

$$\begin{aligned} x_{(u,v)} &= \psi_{(u,v)}^{-1}\left((\pi_u - \pi_v)\cdot \mu_{(u,v)}\right) = \left(\frac{(\pi_u - \pi_v)\cdot \mu_{(u,v)} - s_{(u,v)}}{r_{(u,v)}}\right) \\ &= \mathrm{id}^{-1}\left((\pi_u - \pi_v)\,\frac{\mu_{(u,v)}}{r_{(u,v)}}\right) - \frac{s_{(u,v)}}{r_{(u,v)}}. \end{aligned}$$

We set the modified conductance $\mu^{\mathrm{id}}_{(u,v)} := \mu_{(u,v)}/r_{(u,v)}$. The remaining constant part $-s_{(u,v)}/r_{(u,v)}$ is added to the flow bounds, e.g., $X^{\mathrm{id,min}}_{(u,v)} = X^{\mathrm{min}}_{(u,v)} - s_{(u,v)}/r_{(u,v)}$. Additionally, we adjust the demands by

$$b^{\mathrm{id}}_u = \sum_{(u',v')\in\delta^-(u)} \frac{s_{(u',v')}}{r_{(u',v')}} - \sum_{(u',v')\in\delta^+(u)} \frac{s_{(u',v')}}{r_{(u',v')}} + b_u \quad \forall u \in V,$$

Benders Loop. We solve one MP and then multiple SPs. We omit the original and MP formulations here for brevity. The MP fixes switch arcs $\overline{z}_{(u,v)} \in \{0,1\}$, $(u,v) \in A_S$, and demands $\overline{b}_u \in \mathbb{R}$, $u \in V$. For each outage scenario ("contingency") $c \in \mathcal{C}$, the SP is:

$$\begin{aligned}
\min \quad & \sum_{(u,v)\in A^c_L} c^{>} x^{>,c}_{(u,v)} \\
\text{s.t.} \quad & \sum_{(v',u)\in\delta^-(u)\setminus c} x^c_{(v',u)} - \sum_{(u,v')\in\delta^+(u)\setminus c} x^c_{(u,v')} = \overline{b}_u \;[\varphi^c]\; \forall u \in V && (1)\\
& (\pi^c_u - \pi^c_v)\mu_{(u,v)} = x^c_{(u,v)} && [\omega^c]\;\forall (u,v)\in A^c_L && (2)\\
& X^{>,\min}_{(u,v)} - x^{>,c}_{(u,v)} \le x^c_{(u,v)} \le X^{>,\max}_{(u,v)} + x^{>,c}_{(u,v)} && [\sigma^c]\;\forall (u,v)\in A^c_L && (3)\\
& X^{\min}_{(u,v)} \le x^c_{(u,v)} \le X^{\max}_{(u,v)} && [\tau^c]\;\forall (u,v)\in A^c_L && (4)\\
& (1-\overline{z}_{(u,v)})(\pi^{\min}_u - \pi^{\max}_v) \le \pi^c_u - \pi^c_v \\
& \qquad \le (1-\overline{z}_{(u,v)})(\pi^{\max}_u - \pi^{\min}_v) && [\alpha^c]\;\forall (u,v)\in A_S && (5)\\
& \pi^c_u \in \mathbb{R} && \forall u \in V \\
& x^c_{(u,v)} \ge 0 && \forall (u,v)\in A_S \\
& x^c_{(u,v)} \ge 0 && \forall (u,v)\in A^c_L \\
& x^{>,c}_{(u,v)} \ge 0 && \forall (u,v)\in A^c_L
\end{aligned}$$

It determines potentials π^c_u, $u \in V$, flows $x^c_{(u,v)}$, $(u,v) \in A^c_L$, and overload $x^{>,c}_{(u,v)}$, $(u,v) \in A^c_L$. We assume that the active graph in each subproblem is connected, this is ensured by a separate separation algorithm.

Primal Solution. We contract switched-on switch arcs by merging incident nodes. This reduces the primal subproblem to determining node potentials satisfying

flow conservation:

$$\sum_{(v',u)\in\delta^-(u)\setminus c} \mu_{(v',u)}(\pi^c_{v'} - \pi^c_u) - \sum_{(u,v')\in\delta^+(u)\setminus c} \mu_{(u,v')}(\pi^c_u - \pi^c_{v'}) = \bar{b}_u \quad \forall u \in V,$$
$$\pi^c_{|V|} = \pi^{\text{ref}},$$

where $\pi^{\text{ref}} \in \mathbb{R}$ is an arbitrarily chosen reference potential to ensure solution uniqueness. We denote this SLE by $D_p\pi^c = b_p$, where $\pi^c \in \mathbb{R}^{|V|}$ is the vector of potentials. It has a unique solution [10]: let $D'_p \in \mathbb{R}^{(|V|-1)\times(|V|-1)}$ be the reduced matrix after fixing the reference potential $\pi^c_{|V|}$ to π^{ref}. Its entries are

$$D'_{p,(u,v)} = \begin{cases} \sum_{(u',v')\in\delta(u)\setminus c} \mu_{(u',v')}, & u = v, \\ -\mu_{(u,v)}, & u \neq v \text{ and } (u,v) \in A^c_L, \\ -\mu_{(v,u)}, & u \neq v \text{ and } (v,u) \in A^c_L, \\ 0, & \text{otherwise.} \end{cases}$$

This matrix is (i) weakly diagonally dominant with positive diagonal entries, (ii) strictly dominant for at least one row (graph connected), and (iii) irreducible. This implies full rank and, by Rouché-Capelli, that a unique solution exists.

Optimality Cuts. To construct Benders optimality cuts, we require dual variables of an optimal dual solution. Again, we at first contract switch arcs. Given an optimal primal solution $(\pi^{c*}, x^{c*}, x^{>c*})$, the dual variables $\sigma^{c*}_{(u,v)}$ for Constraints (3) of an optimal dual solution satisfy by complementary slackness

$$\sigma^{c*}_{(u,v)} = \sigma^{\min,c*}_{(u,v)} - \sigma^{\max,c*}_{(u,v)} = \begin{cases} 0, & X^{>,\min}_{(u,v)} \le x^{c*}_{(u,v)} \le X^{>,\max}_{(u,v)}, \\ -c^>, & x^{c*}_{(u,v)} < X^{>,\min}_{(u,v)}, \\ c^>, & x^{c*}_{(u,v)} > X^{>,\max}_{(u,v)}, \end{cases} \quad \forall (u,v) \in A^c_L.$$

The dual variables φ^{c*}_u are then determined with the SLE $D_p\varphi^c = D_d\varphi^c = b_d$

$$\sum_{(v',u)\in\delta^-(u)\setminus c} \mu_{(v',u)}(\varphi^c_u - \varphi^c_{v'}) - \sum_{(u,v')\in\delta^+(u)\setminus c} \mu_{(u,v')}(\varphi^c_{v'} - \varphi^c_u)$$
$$= \sum_{(u,v')\in\delta^+(u)\setminus c} \sigma^{c*}_{(u,v')} - \sum_{(v',u)\in\delta^-(u)\setminus c} \sigma^{c*}_{(v',u)} \quad \forall u \in V,$$
$$\varphi^c_{|V|} = \pi^{\text{ref}}.$$

Following that, the potential equation dual variables are computed as

$$\omega^c_{(u,v)} = \varphi^{c*}_u - \varphi^{c*}_v - \sigma^{c*}_{(u,v)} \quad \forall (u,v) \in A^c_L.$$

We observe that $D_p = D_d$, so the same arguments for uniqueness hold. In particular, the reference value $\pi^{\text{ref}} \in \mathbb{R}$ can be chosen freely: the dual objective contains the term $\sum_{u\in V} \varphi^c_u \bar{b}_u$, but $\sum_{u\in V} \bar{b}_u = 0$, so $\sum_{u\in V}(\varphi^c_u + \varepsilon)\bar{b}_u = \sum_{u\in V} \varphi^c_u \bar{b}_u$ for any $\varepsilon \in \mathbb{R}$. Switch arc duals τ^{c*} are computed via spanning tree propagation for each component of the graph restricted to active switch arcs.

Feasiblity Cuts. A feasibility cut is added if the hard bounds of an arc $(u, v) \in A_L$ are violated. We modify the SLE for optimality cuts. The coefficient matrix has an additional column with entries 1 and -1 for u and v. This accounts for the arc's dual variable for Constraint (5). An additional row sets the dual objective equal to a positive value on the right-hand side. The right-hand side is otherwise equal to 0. A solution of this SLE yields a dual direction of unboundedness.

Factorization Updates. Optimality and feasibility cuts share similar matrix structures. Between subproblems, only outage arcs change, each handled by two rank-1 updates to LU-factorizations. For the complete set of singleton outages, complexity reduces from $\mathcal{O}(|V|^4)$ to $\mathcal{O}(|V|^3)$ per Benders iteration when reusing factorizations. Updates can also be used between Benders iterations.

Computational Results. We use instances from pglib-opf [1] (up to 1888 nodes, 2531 scenarios) and Gurobi as solver [5]. Preliminary results show that using SLEs over LP solvers reduces mean total (SP) solving time by ca. 78.20 % (96.50 %), 84.30 % (98.80 %) with factorization updates.

References

1. Babaeinejadsarookolaee, S., Birchfield, A., Christie, R.D., Coffrin, C., DeMarco, C., Diao, R., Ferris, M., Fliscounakis, S., Greene, S., Huang, R., et al.: The power grid library for benchmarking AC optimal power flow algorithms. arXiv preprint arXiv:1908.02788 (2019)
2. Benders, J.F.: Partitioning procedures for solving mixed-variables programming problems. Numerische Mathematik **4**(1), 238–252 (1962)
3. Birkhoff, G., Diaz, J.B.: Non-linear network problems. Quarterly of Applied Mathematics **13**(4), 431–443 (1956)
4. Fisher, E.B., O'Neill, R.P., Ferris, M.C.: Optimal transmission switching. IEEE Transactions on Power Systems **23**(3), 1346–1355 (2008)
5. Gurobi Optimization, LLC: Gurobi Optimizer Reference Manual (2024)
6. Hedman, K.W., O'Neill, R.P., Fisher, E.B., Oren, S.S.: Optimal transmission switching with contingency analysis. IEEE Transactions on Power Systems **24**(3), 1577–1586 (Aug 2009)
7. Khodaei, A., Shahidehpour, M.: Transmission switching in security-constrained unit commitment. IEEE Transactions on Power Systems **25**(4), 1937–1945 (2010)
8. LeBlanc, L.J., Morlok, E.K., Pierskalla, W.P.: An efficient approach to solving the road network equilibrium traffic assignment problem. Transportation Research **9**(5), 309–318 (1975)
9. Lehmann, K., Grastien, A., Hentenryck, P.V.: The complexity of DC-switching problems. CoRR **abs/1411.4369** (2014)
10. Pfetsch, M.E., Schmidt, M., Skutella, M., Thürauf, J.: Potential-based flows – an overview (Jan 2026)
11. Rahmaniani, R., Crainic, T.G., Gendreau, M., Rei, W.: The Benders decomposition algorithm: A literature review. European Journal of Operational Research **259**(3), 801–817 (2017)
12. Rockafellar, R.T.: Network Flows and Monotropic Optimization. SIAM (1998)
13. Schmidt, M., Thürauf, J.: An exact method for nonlinear network flow interdiction problems. Optimization Online (2022)

A mixed-integer linear programming model for data collection with a drone avoiding obstacles

Y.HU[1][0009−0008−5145−9432], F. BENDALI[1][0000−0002−6147−5101], J.MAILFERT[1][0009−0000−1137−935X], C. CARIOU[2][0000−0002−7343−7452], and L. MOIROUX-ARVIS[2][0000−0002−3146−5041]

[1] Laboratoire, LIMOS, UMR CNRS 6158, Université Clermont Auvergne
Yuankang.HU@doctorant.uca.fr
[2] Laboratoire, TSCF, UR INRAE
Christophe.Cariou@inrae.fr

Keywords: Automated Drone Control · Obstacles · Close Enough Traveling · Salesman Problem · Integer Linear Programming.

1 Introduction

Nowadays, communication sensors are widely used in agricultural fields to measure the physical properties of the topsoil. For protection purposes, these devices are usually buried in the crops. As a result, their radio signals suffer attenuation, leading to a reduced communication range. These communication ranges which are generally modeled as truncated hemispheres of various sizes define the neighborhoods of the sensors. In our study, an Unmanned Aerial Vehicle (UAV) has to collect sensor data by flying through the communication regions of sensors (in red on Figure 1) at different altitudes taking obstacles into account.

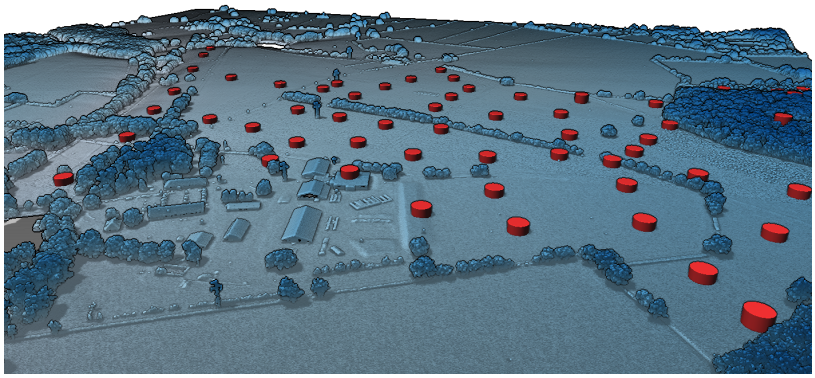

Fig. 1. Experimental farm of Montoldre (INRAE-France)

2 A Mixed-Integer Linear Formulation

The main purpose of the UAV is to reach every sensor communication ranges to collect sensor information. However, in order to be energy efficient during its trip, it needs to build a shortest tour. Our resolution approach consists in using the framework of the Close Enough Travelling Salesman Problem (CETSP) [1, **?**]. Actually, in the CETSP, we have to find a visiting order between n target domains, the communication areas, and select a hitting point in each of them.

We assume that the domain of a sensor $i, i \in \{1, \ldots, n\}$, is a parallelepiped $\mathcal{C}_i$ totally included in the sensor communication range and without any obstacle inside. The drone only needs to hit a point s_i of this zone $\mathcal{C}_i$ for considering the place i as visited. The take-off (resp. landing) area of the drone is numbered by 0 (resp. $n+1$). Let us denote by $\mathcal{V}$ the set $\{0, 1, \ldots, n, n+1\}$ and by $\mathcal{E}$ the set of arcs $\{(i,j) : 0 \leq i \leq n, 1 \leq j \leq n+1; i \neq j\}$.

As we aim to obtain a mixed linear model, the distance between two hitting points s_i and s_j is measured by the Manhattan distance denoted by $d_1(s_i, s_j)$.

Then we get the following formulation.

$$\min \quad \sum_{(i,j)\in\mathcal{E}} \alpha_{ij} \tag{1}$$

$$\text{s.t.} \quad \alpha_{ij} \geq d_1(s_i, s_j) - D_M(1 - x_{ij}) \qquad \forall (i,j) \in \mathcal{E} \tag{2}$$

$$x = (x_{ij}) \in \mathcal{X}$$

$$s_i \in \mathcal{C}_i \qquad \forall i \in \mathcal{V}$$

$$\alpha_{ij} \geq 0 \qquad \forall (i,j) \in \mathcal{E}$$

The binary vector $x = (x_{ij})$ gives the visiting order of the sensor domains since x_{ij} is equal to one if the drone directly travels from domain i to another domain j and 0 otherwise. Vectors x correspond to incidence vectors of Hamiltonian paths from node 0 to node $n+1$. The set of these vectors is denoted by $\mathcal{X}$.

Constraints (2) imply that α_{ij} will be the Manhattan distance between a point $s_i \in \mathcal{C}_i$ and a point $s_j \in \mathcal{C}_j$ if $x_{ij} = 1$, and 0 otherwise. D_M is a large constant used in the big-M technique. The objective (1) is to minimize the total trajectory length of the UAV, while ensuring that every domain is visited once and obstacles are avoided.

Our test zone comes from the INRAE experimental farm in Montoldre (Allier) as illustrated in Figure 1 and is described by Airborne LiDAR data [2][3]. A few preprocessing steps need to be performed first. We aim to check the presence of obstacles in a cubic shaped area between every pair of sensor domains (i,j) as shown in Figure 2. This leads to the definition of obstacle sets $\Omega_{i,j}$, for any pair (i,j). Subsequently, we determine the highest altitude $Z_{max}(i,j)$ within the obstacle set $\Omega_{i,j}$ and incorporated it as a data into the MILP model. For the path optimization, each non empty set $\Omega_{i,j}$ generates specific linear constraints for the calculus of the distance between s_i and s_j.

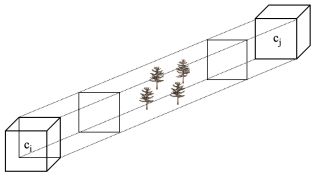

Fig. 2. Detection of obstacles between two domains $\mathcal{C}_i$ and $\mathcal{C}_j$

3 Solving our problem

The mixed linear model was tested on a set of instances of size 15 using the CPLEX solver. As reported in Table 1, optimal solutions were obtained for all instances. The computational times range from 27 to 175 seconds. We further extended the model to a three-dimensional environment including obstacles and tested it on 15 capture instances. The corresponding results are reported in Table 2. Optimal solutions were also obtained for all these instances, showing that the proposed MILP formulation remains effective when extended to more realistic 3D scenarios with obstacles. These results indicate that the proposed MILP formulation is capable of solving small-sized INRAE instances within reasonable times. Unfortunately bigger instances bring huge difficulties in the solving process. So we propose a genetic algorithm that gives the optimal solutions with better running times for the same instances of size 15, see Table 1.

Table 1. Optimal solution for n=15

Instance	**Size**	**Tree Size**	**Solution(m)**	**CplexTime(s)**	**HeuristicTime(s)**
ns15nb0	15	662691	2657.84	53.41	20.16
ns15nb1	15	1970904	2884.00	174.14	20.21
ns15nb2	15	934355	2923.24	67.58	18.92
ns15nb3	15	404685	5002.66	35.45	19.58
ns15nb4	15	591538	4213.26	40.37	19.04
ns15nb5	15	370745	4460.60	27.64	19.65
ns15nb6	15	1316026	3130.64	98.84	18.91
ns15nb7	15	349384	3932.22	26.89	18.91
ns15nb8	15	1240108	3975.62	85.79	20.22
ns15nb9	15	511792	2912.32	36.88	20.18

The genetic heuristic also creates tours of good quality for larger sizes as we can see on Table 3 where CS is the optimal solution or a lower bound obtained by Cplex, UB is the upper bound and the solution of the genetic heuristic is denoted by HS.

Table 2. Optimal solution for 3D instances with obstacles (n=15)

Instance	Size	Tree Size	Solution(m)	CplexTime(s)	Gap(%)
ns15nb0	15	40415	4004.11	15.73	0.000
ns15nb1	15	158292	2897.97	37.40	0.000
ns15nb2	15	61787	4840.40	21.82	0.000
ns15nb3	15	39482	7391.32	15.84	0.000
ns15nb4	15	43649	6175.89	13.80	0.000
ns15nb5	15	37279	5710.58	17.99	0.000
ns15nb6	15	125441	4329.16	29.22	0.000
ns15nb7	15	45762	5055.49	19.19	0.000
ns15nb8	15	159395	5134.09	42.52	0.000
ns15nb9	15	34628	4263.32	13.49	0.000

Table 3. Comparison of results between Cplex and the genetic heuristic

Instance	Size	UB(m)	CS(m)	CplexTime(s)	HS(m)	heuristicTime(s)	HS/CS
ns20nb0	20	3043.34	2505.31	4h	3068.34	24.38	1.2248
ns20nb1	20	3822.02	3368.09	4h	3822.02	23.42	1.1349
ns20nb2	20		4166.90	301.66	4166.90	23.63	1.0000
ns20nb3	20		5018.78	3299.55	5092.16	24.74	1.45688
ns30nb0	30	4578.62	2532.35	4h	4823.54	31.55	1.9052
ns30nb1	30	5136.54	3967.49	4h	5200.84	32.36	1.3110

4 Conclusion

In this paper, we propose a fully linear model that integrates obstacle avoidance with path planning, and solve the CETSP in cluttered environments using CPLEX as a standard MILP solver. In addition, we develop a genetic algorithm able to compute optimized paths efficiently. The proposed approach is tested on real-world agricultural use cases.

This work was supported by the International Research Center "Innovation Transportation and Production Systems" of the I-SITE CAP 20-25"

References

1. Behdani, B., Smith, J.C.: An integer-programming-based approach to the close-enough traveling salesman problem. INFORMS Journal on Computing 26(3), 415–432 (2014)
2. Cariou, C., Moiroux-Arvis, L., Bendali-Mailfert, F., Hu, Y., Mailfert, J.: Unmanned aerial vehicle optimal route planning for data collection of underground communicating sensor nodes in agriculture. Journal on Autonomous Transportation Systems 3(1), 1–17 (2025)
3. Cariou, C., Moiroux-Arvis, L., Bendali, F., Hu, Y., Mailfert, J.: Path planning of unmanned aerial vehicle flying at low height for Internet of Underground Things: application to data collection of buried communicating sensor nodes in agriculture. In: 10th International Conference on Smart and Sustainable Technologies (SpliTech 2025), pp. 1–6. IEEE (2025)

A MILP Approach to Regular MaxSAT ⋆

Jordi Coll[1], Marcelo Finger[2], Felip Manyà[3], Sandro Preto[4], and Elifnaz Yangin[3]

[1] Universitat de Girona, Spain
[2] University of Sao Paulo, Brazil
[3] Artificial Intelligence Research Institute (IIIA, CSIC), Spain
[4] Federal University of ABC, Brazil

1 Introduction

Regular propositional logic is a multiple-valued logical formalism for knowledge representation that lies in the intersection of the areas of constraint programming, many-valued logics and annotated logic programming [1].

Regular formulas are multiple-valued propositional formulas annotated with a regular sign. Given a set of truth values N equipped with a total order $\leq$, a regular sign is a subset of N either of the form $\{j \in N \mid j \geq i\}$, denoted by $\geq i$, or $\{j \in N \mid j \leq i\}$, denoted by $\leq i$, for some $i \in N$. For simplicity, we assume that N is the interval $[0,1]$ or is a finite set of equidistant rational numbers between 0 and 1. Given a multiple-valued propositional formula ϕ, a regular formula is an expression either of the form $\geq i:\phi$ or $\leq i:\phi$. In regular logic, a truth assignment v maps every propositional variable to a value of N. An assignment v is extended to regular formulas by interpreting conjunction as the minimum function, disjunction as the maximum function, and negation as the Łukasiewicz negation. Then, an assignment v satisfies the regular formula $\geq i:\phi$ iff $v(\phi) \geq i$, and v satisfies the regular formula $\leq i:\phi$ iff $v(\phi) \leq i$. Given a multiset of regular formulas Φ, the Regular Maximum Satisfiability problem (Regular MaxSAT) is to find an assignment that minimizes the number of unsatisfied formulas in Φ.

Boolean MaxSAT, along with its multiple-valued extensions Regular MaxSAT and Signed MaxSAT, offers a suitable formalism for representing combinatorial optimization problems. When equipped with competitive solvers (e.g. [12, 11]), MaxSAT and its variants also provide a scalable and efficient framework capable of solving a broad spectrum of combinatorial optimization problems via their reduction to MaxSAT.

In this paper, we introduce a novel reduction from Regular MaxSAT to Mixed Integer Linear Programming (MILP), enabling the use of state-of-the-art MILP solvers to tackle computationally hard Regular MaxSAT instances. To the best of our knowledge, this yields the first competitive solvers for Regular MaxSAT. Prior research on Regular

⋆ This work was carried out at the Artificial Intelligence Research Institute (IIIA-CSIC), with support by grants PID2022-139835NB-C21 and PID2024-157625OB-I00 funded by MCIN/AEI/10.13039/501100011033 and by ERDF, EU; and at the Center for Artificial Intelligence (C4AI-USP), with support by the São Paulo Research Foundation (FAPESP) [grant #2019/07665-4] and by the IBM Corporation. This study was financed, in part, by the São Paulo Research Foundation (FAPESP), Brasil. Process Number #2024/19144-7. E. Yangin was supported by grant PTA2022-022408-I funded by MICIU/AEI/10.13039/501100011033 and FSE+.

MaxSAT has primarily focused on the development of sound and complete resolution and semantic tableaux calculi [4, 3, 5, 6]. However, despite their theoretical significance, no solver implementations based on these calculi have been reported.

This work was inspired by the constraint tableaux approach proposed by Hähnle [9] and Finger and Preto [8] for addressing the satisfiability in the infinitely-valued logic of Łukasiewicz, as well as by the contribution of Haniková et al. [10] on solving the MaxSAT problem within the same logic. In both cases, the original logical problem is reduced to a MILP formulation, and the experimental results reported in [2, 7] confirm the competitiveness of this approach.

2 A MILP transformation

Given a multiset of regular formulas $\Phi = \{S_1 : A_1, \ldots, S_m : A_m\}$, where S_i denotes a regular sign, our aim is to derive a MILP formulation to solve the Regular MaxSAT problem for Φ using a MILP solver.

As a preprocessing step, we eliminate the negation connective from each formula in Φ whenever it is the dominant operator. This is achieved by applying the following transformation rules, each of which replaces a given premise with its corresponding conclusion:

$$\frac{\geq i : \neg A}{\leq 1 - i : A} \qquad \frac{\leq i : \neg A}{\geq 1 - i : A}$$

From now on, no formula in Φ is dominated by the negation connective. It is dominated either by the conjunction or disjunction connective, or it is an atomic regular formula (i.e.; a formula of the form $\geq i : x$ or $\leq i : x$, where x is a propositional variable).

As in every linear optimization problem, we must define the objective function and the constraints that define the feasible region.

The objective function is $\min \sum_{i=1}^{m} p_i$, where each p_i is a binary decision variable. Minimizing $\sum_{i=1}^{m} p_i$ enforces the satisfiability of $S_i : A_i$ in an optimal solution when p_i evaluates to 0. In this way, the number of decision variables set to 0 in an optimal solution is the maximum number of formulas that can be satisfied in Φ, and the optimal value returned by the objective function is the minimum number of formulas that can be unsatisfied.

The constraints that define the feasible region consist of a set of linear expressions designed to ensure that there exists an interpretation that satisfies formula $S_i : A_i$ when p_i is 0. When p_i cannot be set to 0 in an optimal solution, p_i enforces the satisfaction of the constraints, but $S_i : A_i$ cannot be satisfied.

We consider two types of constraints: Given a regular formula $\geq i : A$ ($\leq i : A$), the first type computes the truth value that an interpretation assigns to A, while the second type checks whether this value is greater (smaller) than or equal to i. We refer to the first type as decomposition constraints and to the second type as regular constraints.

The MILP formulation considers three types of variables to capture the essential elements of Regular MaxSAT:

– Definition variables: For each subformula B of A, we introduce a variable $z_B \in N$ that denotes the truth value assigned to B. The formula A has its corresponding variable z_A.
– Decision variables: For each regular formula $S_i : A_i \in \Phi$, we introduce a binary variable $p_i \in \{0,1\}$. In an optimal assignment, if p_i is set to 0, then $S_i : A_i$ is satisfied; otherwise, it is unsatisffied.
– Auxiliary variables: Decomposition constraints need to use fresh binary variables $c, c', c'', \ldots$ to encode the interpretation of connectives as linear equations.

2.1 Regular Constraints

The regular constraint for a regular formula $\geq i : A_i$ ($\leq i : A_i$) checks whether the value assigned to A_i, represented by the assignment variable z_{A_i}, is greater (smaller) than or equal to i. Given a multiset of regular formulas $\Phi = \{S_1 : A_1, \ldots, S_m : A_m\}$, where S_i denotes a regular sign, we add the following regular constraints:

For each regular formula $\geq i_k : A_k$ whose sign has positive polarity, we add the linear inequality $z_{A_k} + p_k \geq i_k$.

For each regular formula $\leq i_k : A_k$ whose sign has negative polarity, we add the linear inequality $z_{A_k} - p_k \leq i_k$.

Observe that, when $p_k = 0$, the first inequality becomes $z_{A_k} \geq i_k$, and the second inequality becomes $z_{A_k} \leq i_k$. Since z_{A_k} denotes the value assigned to A_k, it follows that $S_k : A_k$ is satisfied. In contrast, when $p_k = 1$, both inequalities are trivially satisfied for any values of z_{A_k} and i_k given that $0 \leq z_{A_k}, i_k \leq 1$. This implies that the regular constraints are fulfilled by any feasible solution, but $S_k : A_k$ is not necessarily satisfied. Additionally, by maximizing the number of decision variables assigned the value 0, one ensures that $S_k : A_k$ is unsatisffied in an optimal solution whenever $p_k = 1$.

These transformations can be represented by the following inference rules, which replace the premise by its conclusions:

$$\frac{\geq i : A}{\begin{array}{c} z_A + p \geq i \\ z_A = A \end{array}} \qquad \frac{\leq i : A}{\begin{array}{c} z_A - p \leq i \\ z_A = A \end{array}}$$

$$\geq\text{-rule} \qquad \leq\text{-rule}$$

where p is a fresh binary variable.

2.2 Decomposition Constraints

In the MILP encoding, logical connectives are interpreted by translating their definitions into systems of linear constraints. For each subformula of a given formula, we recursively introduce a definition variable to capture its truth value and enforce the inference rule associated with its dominant connective. These inference rules, which replace the premise by its conclusions, are applied recursively until no further rules can be applied. In the end, we get a set of linear constraints.

The rules for conjunction, disjunction, and negation are defined as follows:

$$\frac{z_{A\wedge B} = A \wedge B}{\begin{array}{c} z_{A\wedge B} \leq z_A \\ z_{A\wedge B} \leq z_B \\ z_{A\wedge B} \geq z_A - c \\ z_{A\wedge B} \geq z_B - (1-c) \\ z_A = A \\ z_B = B \end{array}} \wedge\text{-rule} \qquad \frac{z_{A\vee B} = A \vee B}{\begin{array}{c} z_{A\vee B} \geq z_A \\ z_{A\vee B} \geq z_B \\ z_{A\vee B} \leq z_A + c \\ z_{A\vee B} \leq z_B + (1-c) \\ z_A = A \\ z_B = B \end{array}} \vee\text{-rule} \qquad \frac{z_{\neg A} = \neg A}{\begin{array}{c} z_{\neg A} = 1 - z_A \\ z_A = A \end{array}} \neg\text{-rule}$$

where c is a fresh auxiliary variable.

The $\wedge$-rule encodes $z_{A\wedge B} = \min(z_A, z_B)$, the $\vee$-rule encodes $z_{A\vee B} = \max(z_A, z_B)$, the $\neg$-rule encodes $z_{\neg A} = 1 - z_A$. Notice that, at the end of the rules, we add $z_A = A$ and $z_B = B$ so that we must apply recursively the rules if A or B are not propositional variables.

Example 1. Let $\Phi = \{A_1, A_2, A_3\}$ be a multiset of regular formulas, where $A_1 = \geq 0.2 : x_1 \wedge (x_2 \vee \neg x_3)$, $A_2 = \geq 0.9 : \neg x_1$, and $A_3 = \geq 0.5 : \neg x_2 \vee x_3$. We describe how to reduce the Regular MaxSAT problem for Φ to a MILP problem using our tableau-based approach. First of all, we eliminate negation from A_2, so that the multiset Φ becomes $\{\geq 0.2 : x_1 \wedge (x_2 \vee \neg x_3), \leq 0.1 : x_1, \geq 0.5 : \neg x_2 \vee x_3\}$.

The objective function is $\min \sum_{i=1}^{3} p_i$ and is subject to the following constraints:

$$z_{x_1 \wedge (x_2 \vee \neg x_3)} + p_1 \geq 0.2 \quad z_{x_1} - p_2 \leq 0.1 \quad z_{\neg x_2 \vee x_3} + p_3 \geq 0.5 \tag{1}$$

$$z_{x_1 \wedge (x_2 \vee \neg x_3)} = x_1 \wedge (x_2 \vee \neg x_3)$$

$$z_{x_1} = x_1 \tag{2}$$

$$z_{\neg x_2 \vee x_3} = \neg x_2 \vee x_3$$

In order to derive a MILP instance, we must translate the last three equations into linear constraints. Notice that the second equation, $z_{x_1} = x_1$, is already linear and therefore no transformation is required. For the rest of equations, we must apply the tableau-based rules defined in Section 2.2.

We start by transforming $z_{x_1 \wedge (x_2 \vee \neg x_3)} = x_1 \wedge (x_2 \vee \neg x_3)$. First, we apply the $\wedge$-rule:

$$\begin{array}{ll} z_{x_1 \wedge (x_2 \vee \neg x_3)} \leq z_{x_1} & z_{x_1 \wedge (x_2 \vee \neg x_3)} \leq z_{x_2 \vee \neg x_3} \\ z_{x_1 \wedge (x_2 \vee \neg x_3)} \geq z_{x_1} - c & z_{x_1 \wedge (x_2 \vee \neg x_3)} \geq z_{x_2 \vee \neg x_3} - (1-c) \end{array} \tag{3}$$

$$z_{x_1} = x_1 \tag{4}$$

$$z_{x_2 \vee \neg x_3} = x_2 \vee \neg x_3$$

Since $z_{x_2 \vee \neg x_3} = x_2 \vee \neg x_3$ is not a linear constraint, we apply the $\vee$-rule to it:

$$z_{x_2 \vee \neg x_3} \geq z_{x_2} \quad z_{x_2 \vee \neg x_3} \geq z_{\neg x_3} \quad z_{x_2 \vee \neg x_3} \leq z_{x_2} + c' \tag{5}$$

$$z_{x_2 \vee \neg x_3} \leq z_{\neg x_3} + (1 - c') \quad z_{x_2} = x_2 \tag{6}$$

$$z_{\neg x_3} = \neg x_3$$

Since $z_{\neg x_3} = \neg x_3$ is not a linear constraint, we apply the $\neg$-rule to it:

$$z_{\neg x_3} = 1 - z_{x_3} \qquad z_{x_3} = x_3 \tag{7}$$

We now transform $z_{\neg x_2 \vee x_3} = \neg x_2 \vee x_3$. We first apply the $\vee$-rule to it:

$$\begin{array}{ll} z_{\neg x_2 \vee x_3} \geq z_{\neg x_2} & z_{\neg x_2 \vee x_3} \geq z_{x_3} \\ z_{\neg x_2 \vee x_3} \leq z_{\neg x_2} + c'' & z_{\neg x_2 \vee x_3} \leq z_{x_3} + (1 - c'') \end{array} \tag{8}$$

$$z_{\neg x_2} = \neg x_2$$

$$z_{x_3} = x_3 \tag{9}$$

Since $z_{\neg x_2} = \neg x_2$ is not a linear constraint, we apply the $\neg$-rule to it:

$$z_{\neg x_2} = 1 - z_{x_2} \qquad z_{x_2} = x_2 \tag{10}$$

The MILP encoding of Φ consists of the objective function along with the linear constraints 1–10. By submitting this encoding to a modern MILP solver, we obtain an optimal solution to the Regular MaxSAT problem for the input multiset Φ.

References

1. Bernhard Beckert, Reiner Hähnle, and Felip Manyà. The SAT problem of signed CNF formulas. In David Basin, Marcello D'Agostino, Dov Gabbay, Seán Matthews, and Luca Viganò, editors, *Labelled Deduction*, volume 17 of *Applied Logic Series*, pages 61–82. Kluwer, Dordrecht, 2000.
2. Miquel Bofill, Felip Manyà, Mateu Villaret, and Amanda Vidal. New complexity results for Łukasiewicz logic. *Soft Computing*, 23:2187—2197, 2019.
3. Jordi Coll, Chu Min Li, Felip Manyà, and Elifnaz Yangin. A complete tableau calculus for the Regular MaxSAT problem. In *Proceedings of CCIA*, pages 359–368. IOS Press, 2021.
4. Jordi Coll, Chu Min Li, Felip Manyà, and Elifnaz Yangin. MaxSAT resolution for regular propositional logic. *Int. J. Approx. Reason.*, 162:109010, 2023.
5. Jordi Coll, Chu Min Li, Felip Manyà, and Elifnaz Yangin. A complete tableau calculus for non-clausal regular MinSAT. In *Proceedings of CCIA*, pages 278–287, 2024.
6. Jordi Coll, Chu Min Li, Felip Manyà, and Elifnaz Yangin. Complete tableau calculi for Regular MaxSAT and Regular MinSAT. *Cogn. Syst. Res.*, 90:101319, 2025.
7. Marcelo Finger and Sandro Preto. Probably half true: Probabilistic satisfiability over Łukasiewicz infinitely-valued logic. In *Proceedings of IJCAR, Oxford, UK, Proceedings*, volume 10900 of *Lecture Notes in Computer Science*, pages 194–210. Springer, 2018.
8. Marcelo Finger and Sandro Preto. Polyhedral semantics and the tractable approximation of Łukasiewicz infinitely-valued logic. *J. Log. Comput.*, 35(5), 2025.
9. Reiner Hähnle. Short conjunctive normal forms in finitely-valued logics. *Journal of Logic and Computation*, 4(6):905–927, 1994.
10. Zuzana Haniková, Felip Manyà, and Amanda Vidal. The MaxSAT problem in the real-valued MV-algebra. In *Proceedings of TABLEAUX, Prague, Czech Republic*, pages 386–404. Springer LNCS 14278, 2023.
11. Chu-Min Li, Zhenxing Xu, Jordi Coll, Felip Manyà, Djamal Habet, and Kun He. Boosting branch-and-bound MaxSAT solvers with clause learning. *AI Communications*, 2021.
12. Ruben Martins, Vasco M. Manquinho, and Inês Lynce. Open-WBO: A modular MaxSAT solver. In *Proceedings of SAT 2014*, pages 438–445, 2014.

Large Language Models in Job Shop Scheduling: A Structured Analysis of Current Approaches

Hasan Demir[1][0009−0002−0167−5870], Deniz Demirkan[2][0009−0007−2194−0367], Kürşat Mustafa Karaoğlan[1][0000−0001−9830−7622], and Hakan Kutucu[2][0000−0001−7144−7246]

[1] Department of Computer Engineering, Karabuk University, Türkiye
[2] Department of Software Engineering, Karabuk University, Türkiye
{hasandemir,kkaraoglan,hakankutucu}@karabuk.edu.tr

Abstract. The Job Shop Scheduling Problem (JSSP) is an NP-hard optimization problem that aims to minimize makespan under priority and resource constraints. Traditional methods and deep reinforcement learning approaches have been widely used but face challenges such as lengthy training and dependence on domain expertise. These limitations have motivated the exploration of Large Language Models (LLMs) as an alternative scheduling tool. This study reviews recent LLM-based JSSP studies and classifies them into three categories, namely end-to-end schedule generation, multi-agent systems, and integration with existing optimization methods. The results show that integration is the most effective approach, as algorithmic frameworks constrain LLM outputs and reduce the risk of hallucination. End-to-end approaches provide accessibility but lack scalability, while multi-agent systems offer flexibility at a higher computational cost. Five key challenges are identified, including hallucination, scalability, computational cost, limited generalization, and insufficient benchmark standardization. This study contributes a structured classification framework to the literature and outlines future directions, including domain-specific fine-tuning, small language models, and standardized benchmarking.

Keywords: Job Shop Scheduling Problem · Large Language Models · Combinatorial optimization · Flexible Job Shop Scheduling · LLM-assisted optimization

1 Introduction

The Job Shop Scheduling Problem (JSSP) is an NP-hard optimization problem in which the goal is to minimize the makespan by assigning jobs to machines subject to priority and resource constraints [10]. Exact methods such as branch-and-bound are insufficient for large-scale instances, and traditional approaches, including priority dispatching rules, meta-heuristics, and constraint programming, have been widely used [4]. In recent years, deep reinforcement learning (DRL) methods have shown promise by formulating JSSP as a Markov Decision Process [1, 3, 11]. However, DRL-based methods face limitations, including

lengthy training, dependence on domain expertise for reward design, and difficulty in capturing dynamic scheduling conditions [11].

Large Language Models (LLMs) have emerged as a new tool in combinatorial optimization, enabling scheduling problems to be expressed in natural language [1,10]. LLM applications to JSSP can be grouped into three categories. The first is end-to-end schedule generation, where fine-tuned models directly produce schedules but struggle with scalability and constraint violations [10]. The second involves multi-agent systems, where separate LLM agents handle modeling, coding, and optimization tasks using techniques such as Retrieval-Augmented Generation (RAG) [13]. The third integrates LLMs with existing optimization methods for tasks such as heuristic design, reward function generation, and evolutionary operator evolution. A common finding is that LLM hallucinations must be controlled with verification mechanisms [5].

Despite these developments, a comprehensive study that systematically classifies and evaluates LLM-based JSSP approaches is needed. This study analyzes 13 recent papers within a classification framework, evaluates their strengths and weaknesses, and presents future research directions. Section 2 covers the problem definition and methodology; Section 3 presents the comparative analysis; and Section 4 discusses the conclusions and future work.

2 Background and Methodology

This section presents the literature review process, problem definitions, and the classification framework.

2.1 Literature Review Process

A systematic search covering 2025–2026 was conducted in IEEE Xplore, ScienceDirect, Springer, Web of Science, and Google Scholar. Studies were included if they directly used at least one LLM in the scheduling process, addressed JSSP or its variants, and provided experimental results or contributed datasets. 13 articles met these criteria and were classified into three categories based on the LLM's role: end-to-end generation, multi-agent systems, and integration with existing methods.

In the 13 studies summarized in Table 1, LLMs have been incorporated into the scheduling process using three different approaches: direct schedule generation, multi-agent collaboration, and integration with existing optimization methods. In the integration category, which comprises 8 of the 13 studies, LLMs were used for tasks such as operator design, reward function generation, heuristic evolution, and explanation generation. Different models, such as GPT-3.5, GPT-4, LLaMA, and Qwen, were preferred in the studies, and the problems addressed ranged from classical JSSP to Flexible Job Shop Scheduling Problem (FJSP) and dynamic fuzzy JSSP.

2.2 Problem Definitions

JSSP is an NP-hard combinatorial optimization problem that involves processing n jobs on m machines in a predefined order. Each machine processes only one operation at a time; operations cannot be interrupted, and the job-

internal processing order must be preserved. The primary objective is to minimize makespan [1, 10]. FJSP extends JSSP by allowing each operation to be assigned to one of several suitable machines, requiring simultaneous machine assignment and sequencing decisions [6, 9]. Additional variants addressed in the literature include dynamic fuzzy JSSP [4] and lot-streaming hybrid JSSP [5].

2.3 Analytical Framework and Classification

In the first category, end-to-end schedule generation, the LLM directly produces a schedule from a natural language problem description without intermediate solvers. Two studies utilize LoRA-based fine-tuning for this purpose [1,10]. Figure 1 illustrates the end-to-end scheduling approach using a 3×3 JSSP example taken from the Starjob dataset. The problem is expressed in natural language in Alpaca format, and the LLM, fine-tuned with LoRA, processes this input to produce the makespan value [1].

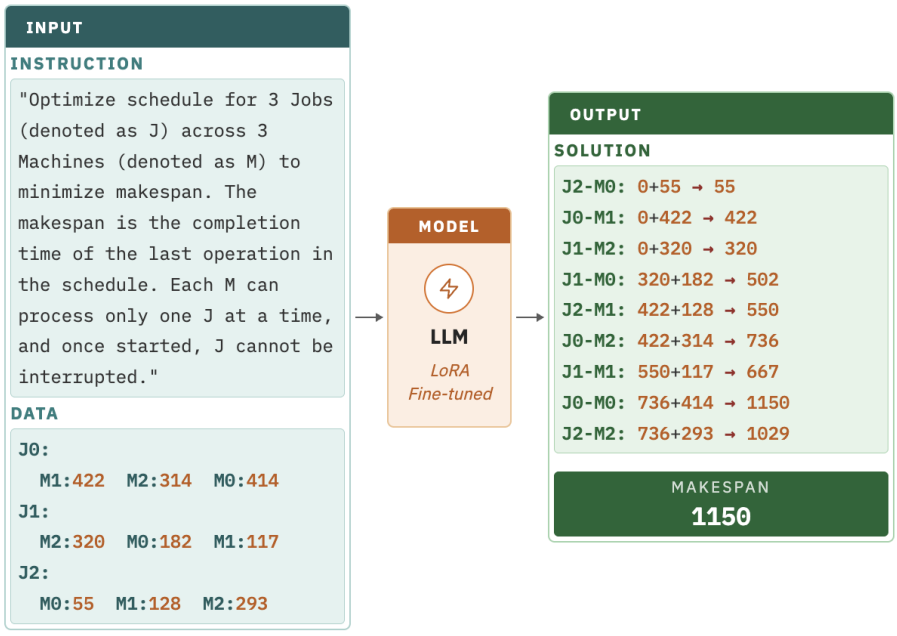

Fig. 1. Alpaca-format input-output transformation for end-to-end LLM-based JSSP.

The second category, multi-agent LLM systems, involves multiple agents collaborating on tasks such as modeling, coding, and optimization, with domain knowledge transferred through RAG [13] and agent chaining [8,9]. The third and broadest category integrates LLMs with existing optimization methods, where LLMs serve as heuristic designers [2, 4–6], reward function generators [3, 11], neighborhood search operators [12], and explanation generators [7]. Figure 2 il-

lustrates the overall classification framework, showing the three categories based on the LLM's role in the scheduling process, along with their respective strengths, limitations, and the mandatory feasibility verification step common to all approaches.

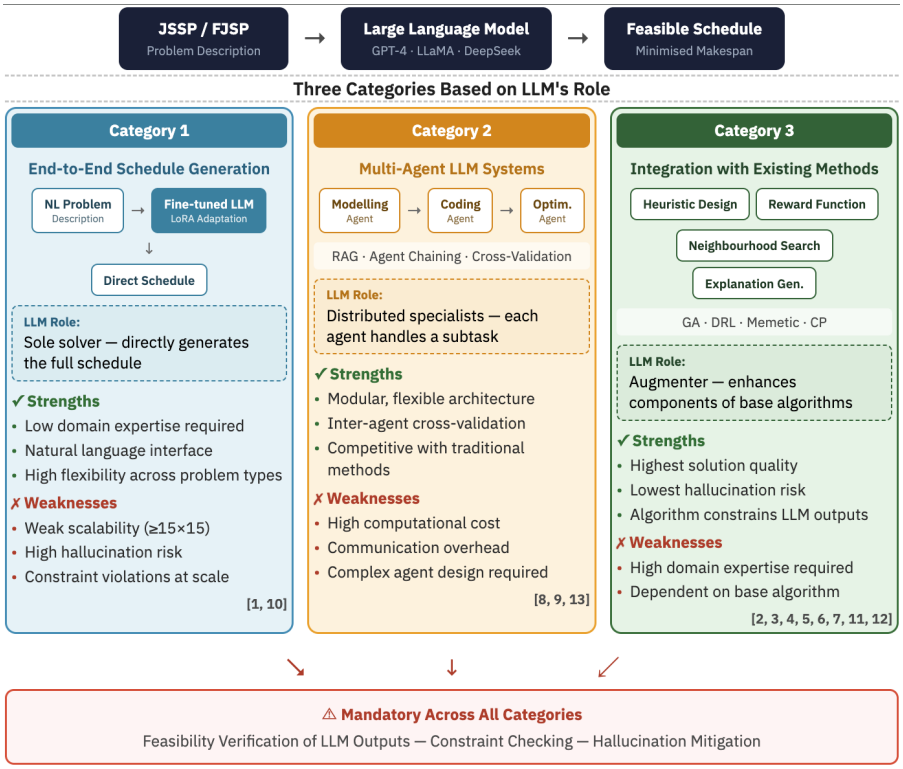

Fig. 2. Classification of LLM-based approaches to JSSP by the role of the LLM in the scheduling process.

3 Comparative Analysis and Discussion

The 13 studies examined in this section are evaluated from a cross-category comparative perspective. First, the strengths and weaknesses of each category are analyzed. Next, the benchmarking datasets and performance metrics used in the studies are compared. Finally, common challenges that limit LLMs' contributions to the scheduling process are discussed.

3.1 Cross-Category Comparison

End-to-end generation requires minimal domain expertise but is the weakest in scalability and hallucination risk. Multi-agent systems offer flexibility through a modular structure but carry a high computational cost. Integration leverages

existing algorithms to provide the highest solution quality and lowest hallucination risk, though it requires algorithmic knowledge.

In end-to-end approaches, untuned models fail to produce viable solutions. LoRA-based fine-tuning significantly improves performance, but constraint violations persist at larger scales (15×15 and above) [1,10]. The Starjob dataset evaluation confirmed that scalability remains a fundamental bottleneck [1]. Multi-agent systems manage complexity by distributing tasks. The MASC framework assigns modeling, coding, optimization, and evaluation to different agents, achieving competitive results in both static and dynamic environments [9]. Multi-agent evolutionary optimization also enables broader exploration of the solution space than single-agent systems [8]. However, inter-agent communication overhead increases computation time.

Integration with existing methods is the most successful category. Population self-evolution outperformed traditional rules in dynamic fuzzy environments [4], cooperative evolutionary frameworks demonstrated superior performance across 85% of tested instances [5], and clustering-based reflection improved multi-objective heuristic design [2]. In DRL integration, LLM-designed reward functions matched or exceeded manually designed ones [11], and the LLM4A3C framework significantly outperformed the baseline A3C algorithm [3]. Table 1 summarizes the reviewed studies, listing the problem type addressed, the role of the LLM, and the corresponding category for each work.

Table 1. Summary of reviewed studies

Study	Problem	LLM Role	Category
Zhang et al. (2025) [13]	JSSP	Multi-agent scheduling	Multi-agent
Zeng et al. (2025) [11]	JSSP	Reward design	Integration
Yang et al. (2025) [10]	JSSP	Schedule generation	End-to-end
Z. Wang et al. (2025) [9]	FJSP	Multi-agent chain	Multi-agent
Y. Wang et al. (2025) [8]	Scheduling	Evolutionary optim.	Multi-agent
J. Zhang et al. (2025) [12]	JSSP/FJSP	Neighborhood search	Integration
Abgaryan et al. (2025) [1]	JSSP	Dataset generation	End-to-end
Powell & Riccardi (2025) [7]	JSSP	Explanation gen.	Integration
Li et al. (2025) [5]	LHJSV	Heuristic design	Integration
Forniés-Tabuenca et al. (2025) [2]	FJSP	Heuristic evolution	Integration
Hong & Li (2025) [3]	FJSP	State/reward def.	Integration
Liao et al. (2025) [6]	FJSP	Operator design	Integration
Huang et al. (2026) [4]	Dyn. fuzzy JSSP	Auto programming	Integration

3.2 Benchmark Datasets and Performance Evaluation

Standard benchmarks such as Taillard and Lawrence have been widely used in classical JSSP studies [11,13], while Brandimarte, Dauzere-Peres, and Barnes datasets have been preferred for FJSP [2,6,9]. The Starjob dataset, specifically designed for LLM-driven scheduling, comprises approximately 130,000 instances ranging from 2×2 to 20×20, with additional larger instances up to 50×20.

Each instance includes a natural-language problem description along with step-by-step solutions that include operation sequences and makespan calculations [1]. Makespan is the dominant performance metric, though some studies also consider total delay time [3], machine load balance [2], and explainability criteria such as accuracy and consistency [7].

3.3 Common Challenges and Limitations

Five fundamental challenges emerge in the application of LLMs to JSSP. The first is hallucination and constraint violations. LLM outputs lack a feasibility guarantee, particularly in end-to-end approaches where priority and capacity violations are common [1, 10]. Even in integration approaches, syntax errors in generated code have been reported [4, 5]. The second challenge is scalability. LLM performance declines as problem size increases, most notably in end-to-end approaches [1, 10]. Integration approaches manage this better through the base algorithm's capacity, but increasing LLM call costs remain a limitation.

The third is computational cost. LLM inference significantly increases computation time, especially in multi-agent systems [8, 9] and in population-based approaches that require LLM calls each generation [4, 6]. The fourth relates to generalization. Fine-tuned models show limited generalization beyond their training distribution [10], and transferability across problem types remains unexplored. The fifth is the lack of standardization in datasets and metrics. The Starjob dataset [1] provides a partial solution with 130,000 natural-language instances, but a comprehensive benchmarking framework covering diverse problem variants and evaluation metrics remains needed.

Overall, integration approaches manage these challenges most effectively by constraining LLM outputs within existing algorithmic frameworks. End-to-end approaches offer accessibility but require significant reliability improvements.

4 Conclusion and Future Work

This study systematically analyzed 13 recent papers on LLM applications to JSSP and its derivatives, categorized into three groups: end-to-end generation, multi-agent systems, and integration with existing methods. Integration emerged as the most common and successful category, where algorithmic frameworks naturally constrain LLM outputs and reduce hallucination risk. End-to-end approaches offer accessibility but face scalability limitations, while multi-agent systems provide consistency through task distribution.

Key findings indicate that LLMs can serve diverse roles, including operator design, reward function generation, and heuristic evolution, but feasibility verification remains mandatory across all approaches. Performance declines with increasing problem size, confirming scalability as an ongoing challenge.

This study has limitations: rapidly evolving model versions may become outdated, different experimental setups hinder direct quantitative comparison, and only JSSP variants were covered. Most methods were tested on benchmarks rather than real industrial settings.

Future research could focus on domain-specific fine-tuning to reduce hallucination, open-source small language models to reduce costs and improve privacy,

multi-objective optimization scenarios, standardized benchmarking frameworks, and real-time rescheduling mechanisms for dynamic production environments.

References

1. Abgaryan, H., Cazenave, T., Harutyunyan, A.: Starjob: Dataset for llm-driven job shop scheduling. arXiv preprint arXiv:2503.01877 (2025)
2. Forniés-Tabuenca, D., Uribe, A., Otamendi, U., Artetxe, A., Rivera, J.C., de Lacalle, O.L.: Remoh: A reflective evolution of multi-objective heuristics approach via large language models. arXiv preprint arXiv:2506.07759 (2025)
3. Hong, T.Y., Li, K.H.: Large language model driven adaptive deep reinforcement learning with dynamic definition refinement for flexible job shop scheduling problem. Applied Soft Computing p. 114253 (2025)
4. Huang, J., Liu, Q., Li, X., Gao, L., Teng, Y.: Automatic programming via large language models with population self-evolution for dynamic fuzzy job shop scheduling problem. IEEE Transactions on Fuzzy Systems (2026)
5. Li, R., Wang, L., Sang, H., Yao, L., Pan, L.: Llm-assisted automatic memetic algorithm for lot-streaming hybrid job shop scheduling with variable sublots. IEEE Transactions on Evolutionary Computation (2025)
6. Liao, R., Qiu, J., Chen, X., Li, X.: Online Operator Design in Evolutionary Optimization for Flexible Job Shop Scheduling via Large Language Models (2025). https://doi.org/10.48550/ARXIV.2511.16485, https://arxiv.org/abs/2511.16485, version Number: 3
7. Powell, C., Riccardi, A.: Generating textual explanations for scheduling systems leveraging the reasoning capabilities of large language models. Journal of Intelligent Information Systems **63**(4), 1287–1337 (2025)
8. Wang, Y., Wang, J., Chu, Z.: Multi-agent large language models as evolutionary optimizers for scheduling optimization. Computers & Industrial Engineering **206**, 111197 (2025)
9. Wang, Z., Wan, C., Liu, J., Zhang, X., Wang, H., Hu, Y., Hu, Z.: Masc: Large language model-based multi-agent scheduling chain for flexible job shop scheduling problem. Advanced Engineering Informatics **67**, 103527 (2025)
10. Yang, X., Huang, J., Liu, Q., Li, X.: Leveraging large language models for end-to-end job shop scheduling scheme generation. In: 2025 6th International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE). pp. 274–278. IEEE (2025)
11. Zeng, Y., Lou, P., Hu, J., Fan, C., Liu, Q., Hu, J.: Large language model-assisted deep reinforcement learning from human feedback for job shop scheduling. Machines **13**(5), 361 (2025)
12. Zhang, J., Luo, C., Su, Z., Zhang, Q., Lü, Z., Ding, J., Jin, Y.: Ns4s: neighborhood search for scheduling problems via large language models. In: Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence. pp. 8687–8695 (2025)
13. Zhang, Z., Ji, Z., Wang, H., Wang, J.: Applying large language models as hybrid-algorithm experts for job shop scheduling problems. In: 2025 IEEE International Conference on Systems, Man, and Cybernetics (SMC). pp. 7263–7268. IEEE (2025)

Polyhedral Analysis of an Envy-Free Assignment Problem

Pierre Fouilhoux[1], Eric Gourdin[2], Roland Grappe[3],
Jules Nicolas-Thouvenin[2], Emilie Sirvent-Hien[2]

[1] LIPN, Université Sorbonne Paris-Nord
[2] Orange Innovation, Chatillon
[3] LAMSADE, Université Paris Dauphine - PSL

Abstract. In this work we introduce inequalities to strengthen the formulation of an envy-free assignment problem. Envy-freeness is a fairness criterion originating in the field of computational social choice that ensures agents prefer their own bundle of resources over every other agent's bundle. We present this fair assignment problem and the polyhedral analysis we derive from it.

Keywords: Polyhedral analysis · Assignment · Envyfreeness.

1 Introduction

We consider a fair assignment problem in which a set of indivisible goods I must be assigned to a set of agents J. Each agent j derives utility s_{ij} from receiving good i. The objective is to maximize the sum of the agents utilities while satisfying two requirements: each good is assigned to exactly one agent, and the allocation is envy-free, i.e., no agent prefers any other agent's bundle over their own bundle. Formally, we define a bundle as the set of goods. An agent j prefers a bundle B_1 to a bundle B_2 if and only if $\sum_{i\in B_1} s_{ij} \geq \sum_{i\in B_2} s_{ij}$. The problem, denoted as the *Envy-Free Assignment Problem* (EFAP), can be formulated as:

$$\max \sum_{i\in I, j\in J} s_{ij}x_{ij} \tag{1}$$

$$\text{s.t.} \sum_{j\in J} x_{ij} = 1, \quad \forall i \in I \tag{2}$$

$$\sum_{i\in I} s_{ij}(x_{ij} - x_{ij'}) \geq 0, \quad \forall j, j' \in J^2 \tag{3}$$

$$x_{ij} \in \{0,1\}, \quad \forall i \in I, \forall j \in J. \tag{4}$$

In the model above, the envy-freeness constraint (3) ensures the difference between the utility for the agent j ($\sum_{i\in I} s_{ij}x_{ij}$) is greater or equal to the utility the agent j would have generated with the bundle of agent j' ($\sum_{i\in I} s_{ij}x_{ij'}$).

This problem comes from a practical assignment problem where additional operational constraints arise in the delivery of promotional messages to clients.

The EFAP corresponds to the standard setting for envy-free division of indivisible goods. It admits no solutions when $|I| < |J|$, is polynomial when $|I| = |J|$ as it is equivalent to an envy-free perfect matching [1] and NP-hard otherwise [2].

Our goal is to study the polyhedron defined by the convex hull of integer solutions, both in this general setting and a particular case where there is one more good than the number of agents.

2 Related Work

This fair assignment problem originates from and has been widely studied in the field of computational social choice. It falls within a setting of fair division of indivisible goods, commonly associated with criteria such as envy-freeness, proportionality or their relaxations (e.g., envy-freeness up to one good (EF1), proportionality up to any good (PROPX)).

The objective function we consider is consistent with the computational social choice literature as it corresponds to the utilitarian welfare. Other common objective functions are the Nash product and Pareto optimality. However, much of this literature focuses on decision problems. Typical questions include whether a fair allocation exists for a given instance, whether the problem can be solved in polynomial time, which algorithms can construct such allocations, and whether particular fairness criteria imply additional desirable properties.

On the other hand, studies on fair assignment problems in operations research feature other fairness criteria [3], such as Jain's index, maxmin fairness, $\alpha-$fairness to name a few. These approaches in particular formulate fairness within an optimization framework rather than as feasibility conditions.

A recent survey [2] presents key complexity results for envy-free assignment problems and discusses several solving methods, including mixed-integer programming (MIP) solvers.

To the best of our knowledge, no research has yet been done on the polyhedral characterizations of envy-free assignment problems.

3 Results

We derive several inequalities from a polyhedral analysis of the feasible region. These inequalities represent initial steps toward translating the numeric nature of envy-freeness into combinatorial aspects. We also propose experimental analysis of their impact on MIP solvers.

3.1 Cover Inequalities

The first family generalizes knapsack cover inequalities: when two agents j and j' have such preferences over the distinct bundles B_j and $B_{j'}$ that assigning

simultaneously all items of B_j to j and all items of $B_{j'}$ to j' creates envy from j towards j', this leads to a constraint of the form:

$$\sum_{i \notin B_j} x_{ij} \;\geq\; \sum_{i \in B_{j'}} x_{ij'} - |B_{j'}| \tag{5}$$

Note that the family of these cover inequalities forms a combinatorial alternative to the numerical envy-freeness inequalities (3).

Unfortunately, these covers are quite weak. We then show how to strengthen the cover inequality (5) when the bundle $B_{j'}$ is envied by more than one agent and propose a heuristic separation method.

3.2 Maximum-Value Based Inequalities

The second inequality relies on the notion of *maximum-value items* for an agent. Let $m_j = \max_{i \in I} s_{ij}$ denote the maximum item value for agent j, let

$$M_j = \{i \in I : s_{ij} = m_j\}$$

be the set of items attaining this value for j, and let $\bar{M}_j$ denote its complement. Since every good must be assigned, each agent receives at least one item. Moreover, agent j must receive a bundle of value at least m_j.

Suppose that j does not receive any item in M_j. Then j must receive multiple items to compensate for the loss of a maximum-value item. Let k_j be a lower bound on the size of such a *compensation bundle* that can be easily computed as

$$k_j = \min_{S \subseteq \bar{M}_j} \left\{ |S| : \sum_{i \in S} s_{ij} \geq m_j \right\}.$$

The following inequality is therefore valid:

$$\sum_{i \in \bar{M}_j} x_{ij} \geq k_j \left(1 - \sum_{i \in M_j} x_{ij} \right), \quad \forall j \in J. \tag{6}$$

In addition, the same compensation principle can be generalized in a rank-based manner. For instance, if agent j receives neither an item of value m_j nor an item of value m_j^2 (its second-highest value), we can compute a corresponding bound k_j^2 and derive another valid inequality.

3.3 The One More Good Case

We show how the inequalities above improve our understanding of the polyhedron in the case where there is one more good than the number of agents, i.e. $|I| = |J| + 1$.

References

1. Arbib, C., Karaşan, O., Pınar, M.: On envy-free perfect matching. Discrete Applied Mathematics **261**, 22–27 (2019), gO X Meeting, Rigi Kaltbad (CH), July 10–14, 2016
2. Nguyen, T.T., Rothe, J.: Complexity Results and Exact Algorithms for Fair Division of Indivisible Items: A Survey. In: Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence. pp. 6732–6740. International Joint Conferences on Artificial Intelligence Organization, Macau, SAR China (Aug 2023)
3. Xinying Chen, V., Hooker, J.N.: A guide to formulating fairness in an optimization model. Annals of Operations Research **326**(1), 581–619 (Jul 2023)

FalCom: An MCMC Sampling Framework for Districting and Hierarchical Capacitated Facility Location Problems

Hemanshu Kaul and Alaittin Kırtışoğlu

Department of Applied Mathematics, Illinois Institute of Technology,
Chicago, IL 60616, USA
kaul@illinoistech.edu, akirtisoglu@hawk.illinoistech.edu

Abstract. We introduce *FalCom* (Facility Location Combination), the first Markov chain Monte Carlo framework for sampling hierarchical and capacitated facility location solutions with contiguous districts. It is a graph partitioning method that extensively utilizes spanning tree cuts to design hierarchical districts with capacity-demand constraints. The chain is irreducible and aperiodic, guaranteeing convergence to a unique stationary distribution. FalCom handles instances with up to 50,000 basic units, at least one order of magnitude beyond existing methods.

Keywords: Facility location · Districting · MCMC · Graph partitioning

1 Introduction

Districting is the problem of grouping small geographic areas, called *basic units*, into larger geographic clusters, called *districts*, such that the resulting partition satisfies balance, contiguity, and compactness criteria [2]. In the capacitated facility location setting, each district is anchored by a *facility*, such as a clinic, depot, precinct station, or sales office, whose capacity determines how much demand can be served within the district. Many real-world systems require districts at multiple administrative levels. We call these higher-level aggregations *superdistricts*. A hierarchical districting problem simultaneously partitions basic units into districts and districts into superdistricts, subject to balance and contiguity constraints at every level.

Enforcing contiguity requires each district to induce a connected subgraph, which introduces an exponential number of constraints in integer programming formulations [5]. Adding strict capacity bounds further shrinks the feasible region, and multi-level hierarchy compounds the combinatorial explosion. As a result, exact mixed-integer programming methods are generally limited to instances with a few hundred to one thousand basic units [4]. Metaheuristic approaches extend this range to roughly 1,000–5,000 units [3]. A comprehensive survey [2] concludes that "research should first and foremost concentrate on a common and generic framework for districting problems" rather than continue developing application-specific heuristics with yet another compactness measure

or balance constraint. This paper contributes toward that goal. We introduce *FalCom* (Facility Location Combination), the first MCMC sampler in the literature for the hierarchical capacitated facility location problem with contiguous service areas. FalCom generalizes the recombination method [1] from political redistricting literature to handle graphs with up to 50,000 basic units, at least one order of magnitude beyond most published location-allocation methods.

2 Problem Formulation and State Space

We consider a hierarchical facility location formulation with the set of hierarchy levels $L = \{1, \ldots, h\}$ over a geographical region. Facility locations are decision variables selected from a set of candidate sites, and each demand unit is allocated to exactly one facility at every level, forming contiguous service districts. The number of service teams assigned to each facility is determined. Demand and capacity must be balanced at every level, and the total operational cost is bounded by a budget.

The base graph $G^1 = (V^1, E^1)$ is a planar graph in which each node $i \in V^1$ corresponds to a basic geographic unit and each edge $ij \in E^1$ indicates that units i and j share a boundary. Every node $i \in V^1$ carries a demand value $d_i \geq 0$, and the total demand for the region is $d = \sum_{i \in V^1} d_i$. A *partition* $\mathcal{P}^1 = \{D_1^1, \ldots, D_{k_1}^1\}$ splits V^1 into pairwise disjoint districts, each inducing a connected subgraph. The *supergraph* $G^2 = (V^2, E^2)$ has one node per district in $\mathcal{P}^1$ and one edge per pair of adjacent districts sharing a boundary. A partition $\mathcal{P}^2$ of G^2 groups districts into superdistricts. We define a valid *h-hierarchical districting* as

$$\mathcal{P} = \{\mathcal{P}^1, \mathcal{P}^2, \ldots, \mathcal{P}^h\} \quad \text{s.t.} \quad \bigcup_{i \leq k_\ell} D_i^\ell = \mathcal{P}^{\ell-1}, \quad \forall\, \ell \geq 2, \tag{1}$$

ensuring every level-ℓ district is a union of level-$(\ell-1)$ districts. We note that the number of districts k_ℓ is not fixed a priori but emerges from the capacity and demand. Let $F^\ell \subseteq V^1$ denote the set of candidate nodes at which level-ℓ facilities may be located; the candidate sets are pairwise disjoint, $F^\ell \cap F^{\ell'} = \emptyset$ for $\ell \neq \ell'$. At each level ℓ, a *facility assignment* $f^\ell \colon \mathcal{P}^\ell \to F^\ell$ maps each district to a candidate site within it, and a *capacity assignment* $c^\ell \colon \mathcal{P}^\ell \to \{1, \ldots, c_{\max}^\ell\}$ specifies the number of service teams at level ℓ, where $c_{\max}^\ell$ denotes the maximum allowable capacity. We define Ω as the set of solutions $s = (\mathcal{P}^\ell, f^\ell, c^\ell \mid \ell \in L)$.

Contiguity requires that every $G^\ell[D_i^\ell]$ is connected. Consider w as the workload each service team can handle, and $\varepsilon^\ell \in (0, 1)$ the demand-balance tolerance at level ℓ. A facility at node j with capacity c_j^ℓ therefore supports a total workload of wc_j^ℓ. When demand is concentrated, the model responds by increasing the capacity of assigned facilities rather than locating multiple facilities in the same unit, an assumption consistent with realistic budget and operational constraints. Capacity-demand balance requires

$$(1-\varepsilon^\ell) wc^\ell(D_i^\ell) \leq d(D_i^\ell) \leq (1+\varepsilon^\ell) wc^\ell(D_i^\ell) \tag{2}$$

for every district $D_i^\ell \in \mathcal{P}^\ell$ and level $\ell \in L$. Let $\sum_{j \in F^\ell} b_j \leq b$ be the total budget constraint for which b_j is the operational cost of assigning a candidate facility j. Coverage bounds the aggregate capacity

$$\sum_{j \in F^\ell} c_j^\ell \leq d/w, \quad \forall \ell \in L, \tag{3}$$

which imposes an upper bound on the total number of service teams deployed at level ℓ. Since the total demand is d and each team serves a workload of w, the aggregate capacity across all open facilities cannot exceed d/w. We define a rule set $\mathcal{C} = \bigcup_{\ell \in L} \mathcal{C}^\ell$ for these constraints, where

$$\mathcal{C}^\ell = \big\{\text{contiguity}(\mathcal{P}^\ell),\ \text{balance}(\mathcal{P}^\ell, c^\ell),\ \text{coverage}(\mathcal{P}^\ell, c^\ell),\ \text{budget}(f^\ell)\big\}. \tag{4}$$

Each of these rules are satisfied either by construction or checked via an explicit rejection step in FalCom. Finally, a solution $s \in \Omega$ is *feasible* if it satisfies every rule in $\mathcal{C}$. The *feasible state space* is

$$\Omega_{\mathcal{C}} = \big\{s \in \Omega : s \text{ satisfies all constraints in } \mathcal{C}\big\}, \tag{5}$$

and each $s \in \Omega_{\mathcal{C}}$ is a *state* of FalCom.

3 The Outline of FalCom

FalCom consists of four phases and six algorithms presented for two hierarchical levels. A Markov chain iterates over the states

$$s^{(i)} = \big(\mathcal{P}^\ell, f^\ell, c^\ell \mid \ell \in L\big)^{(i)} \longmapsto s^{(i+1)} = \big(\mathcal{P}^\ell, f^\ell, c^\ell \mid \ell \in L\big)^{(i+1)}$$

via the following bottom-up strategy:

1. **Upper level.** Build the supergraph G^2 of $(\mathcal{P}^1)^{(i)}$ and partition it into new superdistricts $(\mathcal{P}^2)^{(i+1)}$ with capacity assignments $(c^2)^{(i+1)}$.
2. **Lower level.** Choose a superdistrict $D_j^2 \in (\mathcal{P}^2)^{(i+1)}$ uniformly at random. Re-partition the induced subgraph $G^1[D_j^2]$ into new districts $D_1^1, \ldots, D_m^1$ with capacity assignment $(c^1)^{(i+1)}$, while keeping all districts outside D_j^2 unchanged.
3. **Facility assignment.** Assign facilities by defining $(f^1)^{(i+1)}$ and $(f^2)^{(i+1)}$.
4. **Accept/reject.** If candidate-awareness score $\psi > 0$ for all cuts used in the proposal, accept $s^{(i+1)} \leftarrow s'$; otherwise reject and set $s^{(i+1)} \leftarrow s^{(i)}$.

Steps 1 and 2 each call a recursive partitioning phase to partition a graph recursively with feasible capacity assignments. Each recursion extracts a district from the graph with a capacity assignment, and returns a residual for the following recursions. To extract a district from a subgraph of the lower or upper level graph, we generate a random spanning tree of the merged districts in D_j^2 via Wilson's random walk algorithm [6], and cut an edge in the capacitated cut

phase with respect to a probability distribution over edges. As a result, recursive partitioning phase shrinks the residual graph using these capacitated tree cuts, with a dynamic demand-debt mechanism ensuring that every district satisfies the capacity-demand condition. Figure 1 illustrates one iteration of FalCom without capacity-demand balance and facility assignment details.

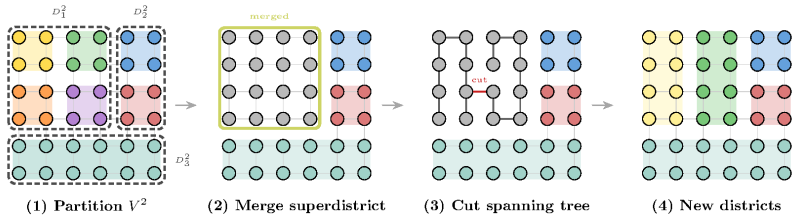

Fig. 1. One iteration of FalCom without capacity-demand balance and facility assignment.

Removing an edge from a spanning tree always yields two connected components, therefore contiguity is guaranteed by construction. A candidate-awareness score biases cut selection toward districts with centrally located facilities. Let $e = uv$ be an edge in a rooted tree T. Let T_u denote the subtree beneath u in T. The *candidate-awareness score* of T_u is

$$\psi(T_u) = \phi(u) \cdot e^{-\gamma\, r(T_u)},$$

where $\phi(u)$ is the feasibility function that receives 1 if T_u has capacity-demand balance for a positive capacity level $c \le c^{\ell}_{\max}$ and contains a candidate facility, zero otherwise. $r(u)$ is the *facility-demand radius* of u in G, defined as the eccentricity of the best-located candidate in T_u, and $\gamma \ge 0$ controls preference for centrally located facilities. Cuts are selected with probability proportional to ψ, biasing centrally located candidate facilities within districts. When $\gamma = 0$, cut selection is uniform over admissible edges. After we complete recursive partitioning phase, the most central candidate facility within each district is assigned to that district with the capacity c.

Since every spanning tree of every connected subgraph has positive probability and every admissible cut has $\psi > 0$, for any two feasible states $s, s' \in \Omega_{\mathcal{C}}$ there exists $n \ge 1$ such that $P^n(s, s') > 0$, thus the chain is irreducible. The same spanning tree and same cut that produced the current state can be sampled again with positive probability, giving $P(s, s) > 0$ and aperiodicity. By the fundamental convergence theorem for finite-state Markov chains, FalCom converges to a unique stationary distribution π^*. The distribution π^* is determined implicitly by the spanning tree distribution and the candidate-awareness score ψ, states whose districts have centrally located facilities are easier to reach and to reproduce, so π^* naturally concentrates on such configurations. A future direc-

tion is to sample from an augmented state space that carries the spanning forest forward, yielding a computable proposal ratio and an explicit target distribution.

4 Computational Experiments

We evaluate FalCom on synthetic grid graphs and two real-world case studies: emergency service zone design and commercial territory planning. Synthetic square grids with $|V|$ ranging from 100 to 50,000 are used to measure wall-clock time per step. Because each iteration operates on a local merged subgraph $H = G^1[D_j^2]$, per-step cost scales with $|H|$ rather than $|V|$. Varying the candidate-awareness parameter γ shows that $\gamma > 0$ produces partitions with smaller demand radii compared to uniform cut selection at $\gamma = 0$, validating that ψ effectively steers the chain toward districts with centrally located facilities. For both real-world instances, FalCom generates ensembles of thousands of feasible solutions, enabling boundary frequency maps that distinguish robust from fragile boundaries.

5 Conclusion

FalCom provides the first principled MCMC framework for hierarchical capacitated facility location with contiguous districts. FalCom does not compete with existing optimization models; rather, it complements them. Flow-based formulations, stochastic and robust models, MIP solvers, and metaheuristics can all be applied within FalCom's districts and superdistricts to optimize facility placement, route vehicles, or allocate staff inside each zone. As a result, FalCom can serve as a framework and ensemble generation tool by sampling thousands of feasible, demand-capacity balanced partitions.

Disclosure of Interests. The authors have no competing interests to declare.

References

1. DeFord, D., Duchin, M., Solomon, J.: Recombination: A family of markov chains for redistricting. Harvard Data Science Review **3**(1), 3 (2021)
2. Kalcsics, J., Ríos-Mercado, R.Z.: Districting problems. In: Location science, pp. 705–743. Springer (2020)
3. Ríos-Mercado, R.Z., Fernández, E.: A reactive GRASP for a commercial territory design problem with multiple balancing requirements. Computers & Operations Research **36**(3), 755–776 (2009)
4. Salazar-Aguilar, M.A., Ríos-Mercado, R.Z., Cabrera-Ríos, M.: New models for commercial territory design. Networks and Spatial Economics **11**(3), 487–507 (2011)
5. Validi, H., Buchanan, A., Lykhovyd, E.: Imposing contiguity constraints in political districting models. Operations Research **70**(2), 867–892 (2022)
6. Wilson, D.B.: Generating random spanning trees more quickly than the cover time. Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing pp. 296–303 (1996)

Author Index

118 contributing authors